Practical Language Processing Project
Bard - Poem Generation System

Tang Yi A0261851E, Lin Zhengxi A0261674Y, Wang Zhiyuan A0261827Y, Guo Rui A0176188N, Zou Yufan A0261786R

1 Abstract

Recently, people not only aspire to a high quality of material life but also put forward higher requirements for the richness of their spiritual life. In their leisure time, people engage in various recreational activities to enrich their spiritual life, such as reading and writing. Among these activities, Chinese classical poem writing is a meaningful one. It not only exercises literary creativity and cultivates imagination, but also helps to promote overall development. As a result, more and more people are engaging in Chinese poem writing, hoping to express their emotions and thoughts through poetry. However, many people need help understanding the rules of poem writing and have poor writing abilities and imagination, resulting in unsatisfactory poetry creation. There are a few aids available such as JIUGE, but some of them have unsatisfactory performance and very single function. To solve this business problem, our team proposes a poetry-generating system, named Bard, which provides meaningful references to users and assists them to compose beautiful and creative poems that conform to the rules of composition. Our system is open to anyone who likes to create poetry. More importantly, this system is a multi-tasking collection that enables diversified poetry creation.

In the paper, our primary proposition is the utilization of various computational models, including Mixpoet and GPT2. These models, through their unique attributes and capabilities, will be harnessed to generate poetry that satisfies diverse sets of requirements. The poem-matching and recommendation model is also proposed. Several compelling elements pertain to this field of study, including the choice of sentiment, voice input methodologies, and the dynamics of chatbot interaction, among others. Each of these aspects constitutes an integral part of the broader research landscape and contributes to its complexity and richness. We have conducted experiments on the poem generation task with a dataset of half a million poems for GPT2 and a dataset of over 10 thousand labeled poems for Mixpoet. The final evaluation has shown that our poetry-generating system achieves a satisfactory performance.

2 Business Problem

Recent social trends reflect not only a desire for higher material standards of living but also a dramatic demand for a richer and more varied spiritual life. In pursuit of this goal, individuals seek to enrich their inner world and spiritual experience by engaging in a variety of recreational activities, such as reading and writing, during their leisure time.

Within this range of activities, poetry writing has become a particularly meaningful way to do so. Not only is this pursuit an exercise in literary creativity and a cultivation of the imagination, but it also contributes to the overall development of the individual. As a result of these benefits, poetry writing has seen a surge in participation, with more and more people looking to express their emotions and thoughts through this creative medium.

However, a significant number of these individuals experience challenges in understanding the complex rules of poetry writing, often coupled with underdeveloped writing skills and a lack of imagination. These factors often end up in poems that do not meet the expectations of their creators. In response to these challenges, some supplementary resources or tools have been developed, such as JIUGE[1], which is an

AI poem-writing system developed by Tsinghua University. More early, approaches are based on rules or templates. The genetic algorithm is also employed. More recently, research on deep learning has become a hot topic, which has given rise to its application in the field of poetry composition. Yi et al [2] have proposed an RNN encoder-decoder method to generate Chinese classical poems. Yan et al.[3] employ recurrent neural networks with the iterative polishing schema to improve context coherence. However, these tools often have performance issues and some offer only a single function, thus limiting their usefulness in aiding the poetry writing process.

Therefore, there remains a critical need for a comprehensive approach to support people who want to create poetry. Such an approach can provide robust performance and multifaceted functionality to enrich and enhance their creative abilities.

To solve this common problem in the field of poetry creation, our team took the initiative to develop a Chinese classical poem generation system that we call Bard. Bard can assist users in creating creative, well-structured, and logical poems. The system provides users with the necessary guidelines, examples, and assistance to ensure that the poems they create conform to accepted poetic standards and structures, thus ensuring that the result is consistent and satisfactory with the creator's intentions. Bard is integrated and designed to be a multi-tasking, multi-functional system. This feature allows users to delve into the creation of various poetry forms.

The Bard system mainly consists of 5 sections: GPT-2 quatrain generation model, GPT-2 acrostic poem generation model, Mixpoet generation model, Chinese classical poem recommendation model, and chatbot and front-end page interaction design. Quatrain(JUEJU) generation model is a tool where users provide a summary and keywords of the poem they want to write, use those keywords to communicate their ideas to an AI model, and then generate multiple versions to find the one that resonates most with them. The Acrostic poem generation model is planned to create acrostic poems, which is a type of Chinese poetry in which the first character of each line combines to form a specific phrase or message. The Mixpoet generation model enables the combination of keywords and sentiment selection to generate Chinese classical poems. The recommendation model applies similarity calculation to recommend poems based on keywords input. The chatbot interaction design achieves usersâ speech input for the desired poem information. In addition. Our team conducts experiments based on online labeled datasets and self-labeled datasets.

3 Literature Review

The creation of poetry has gone through several stages: creation based on rule, semantic, and grammar templates; creation using deep learning models. For rule-based approaches, Tosa et al [4] proposed an interactive supporting system to generate haiku poems based on â1000 Books and 1000 Nightsâ. Wu et al[5] apply cultural characteristics such as Japanese-style color patterns to build an interactive Renku poem generation system. Netzer et al [6] consider Word Association Norms to perform word association to analyze the computational generation ability of linguistics. Oliveira [7] proposed a versatile platform, named PoeTryMe to generate poems, which includes high-level customization. All the above approaches are traditional rule base models. They achieve the poem generation function with automatic methods. However, the disadvantages of the rule-based method are also distinct. Lack of flexibility is the main issue. Rules are usually static and do not adapt to new or unknown situations. If the input or the environment changes, the rules may no longer apply and would require manual tweaking. Rules have limited scalability, which means creating and managing all possible rules can become impractical in large-scale or complex systems. Greene et al. [8] created a generation and translation rhythmic poetry system based on statistical methods. Yan et al.[3] applied a summarization framework with constraints to consider the poem generation issue as an optimization problem. Zhou et al.[å“@inpenc@undefined@latin1¨æ“@inpenc@undefined@latin1“@inpenc@undefined@latin1ä[1]\@*inpenc@unde*
considered genetic algorithms to generate poems.

Recently, deep learning methods attract much attention for poem generation. Lapata et al. [9] consider RNN architecture for generating quatrain, which generates the first line of a poem from keyword input with RNN, followed by subsequent lines generated from accumulating status of existing lines. Wang et al.[2] proposed an end-to-end neural network translation model for Chinese song iambics. Marjan Ghazvininejad et al.[10] applied an encoder-decoder model to generate rhyme words by keywords. Compared with the above-mentioned approaches, our system is an integrated platform combing several models to achieve more powerful functions.

4 The Datasets Used

4.1 GPT-2 acrostic poem generation model dataset

The dataset we use originates from GitHub and contains over 550,000 five-character and seven-character quatrains and regulated verses. The dataset is structured with fields including "form", "title", "dynasty", "author", "first character of each line", and "content".

4.2 GPT2 quatrain(JUEJU) poem generation model dataset

The data set from THUNLP-AIPoet and another resource in GitHub. They are a variety of classic Chinese poems with keywords and styles marked. Figure 1 shows the datasets for JueJu generation model.

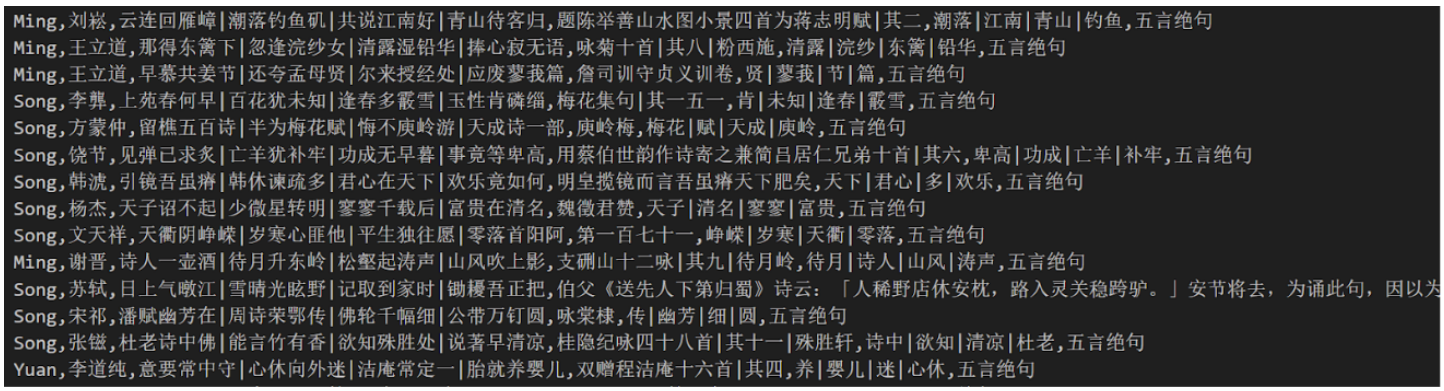

Figure 1: JueJu datasets

4.3 Mixpoet model dataset

The Mixpoet model mainly considers two sentiment factors: living experience and historical background. to meet the requirements of sentiment analysis and diversity. The first one consists of three classes: military career, which is generally negative sentiment, countryside life, which is a positive sentiment, and others. The historical background is divided into two classes: prosperous times (positive sentiment) and troubled times (negative sentiment). Based on these factors, we have created six combinations. The datasets are labeled according to the six combination styles. The labeled corpus is called the Chinese Quatrain Corpus with Factors (CQCF). There are around 11000 poems in CQCF, some of which are obtained from GitHub, and some of which are labeled by ourselves. Besides, an unlabeled dataset is provided, which comprises around 120000 poems. The unlabeled dataset is used to perform self-supervised learning. TextRank[11] method is applied to form labels for each poem. The figure 2 is an example of a MixPoet data.

```
{"label": "-1 1", "content": "横林渺渺夜生烟|野水茫茫远拍天|菱唱一声惊梦断|始知身在钓鱼船", "keywords": "茫茫 渺渺 钓鱼 梦断"}
{"label": "-1 1", "content": "此君何坦坦|回首杏园游|魂魄湘潭去|声名彭泽休", "keywords": "杏园 回首 彭泽 声名"}
{"label": "-1 1", "content": "门前天镜倒千峰|舍后菰蒲与海通|乘兴出游无远近|烟波何处觅孤篷", "keywords": "菰蒲 烟波 乘兴 远近"}
{"label": "-1 1", "content": "有兴或饮酒|无事多掩关|寂静夜深坐|安稳日高眠", "keywords": "安稳 掩关 日高 饮酒"}
{"label": "-1 1", "content": "季子黄金尽|安仁白发新|无情五更雨|便送一年春", "keywords": "安仁 黄金 白发 季子"}
{"label": "-1 1", "content": "家随兵尽屋空存|税额宁容减一分|衣食旋营犹可过|赋输长急不堪闻", "keywords": "减 空存 屋 衣食"}
{"label": "-1 1", "content": "偃蹇松逃世|翩仙鹤驾风|旧交零落尽|一笑与谁同", "keywords": "偃蹇 松 零落 鹤驾"}
{"label": "-1 1", "content": "老境虽深身更健|清秋欲半日犹长|闭门莫道都无事|又了移花一段忙", "keywords": "清秋 老境 闭门 移花"}
```


Figure 2: Mixpoet dataset

5 Solution Approach

Figure 3 shows the whole architecture of our system design. Two GPT2 models, MixPoet and recommendation system will all interact with the front end page, while there is a chatbot for both GPT2-Quatrain Poem and GPT2- Acrostic Poem.

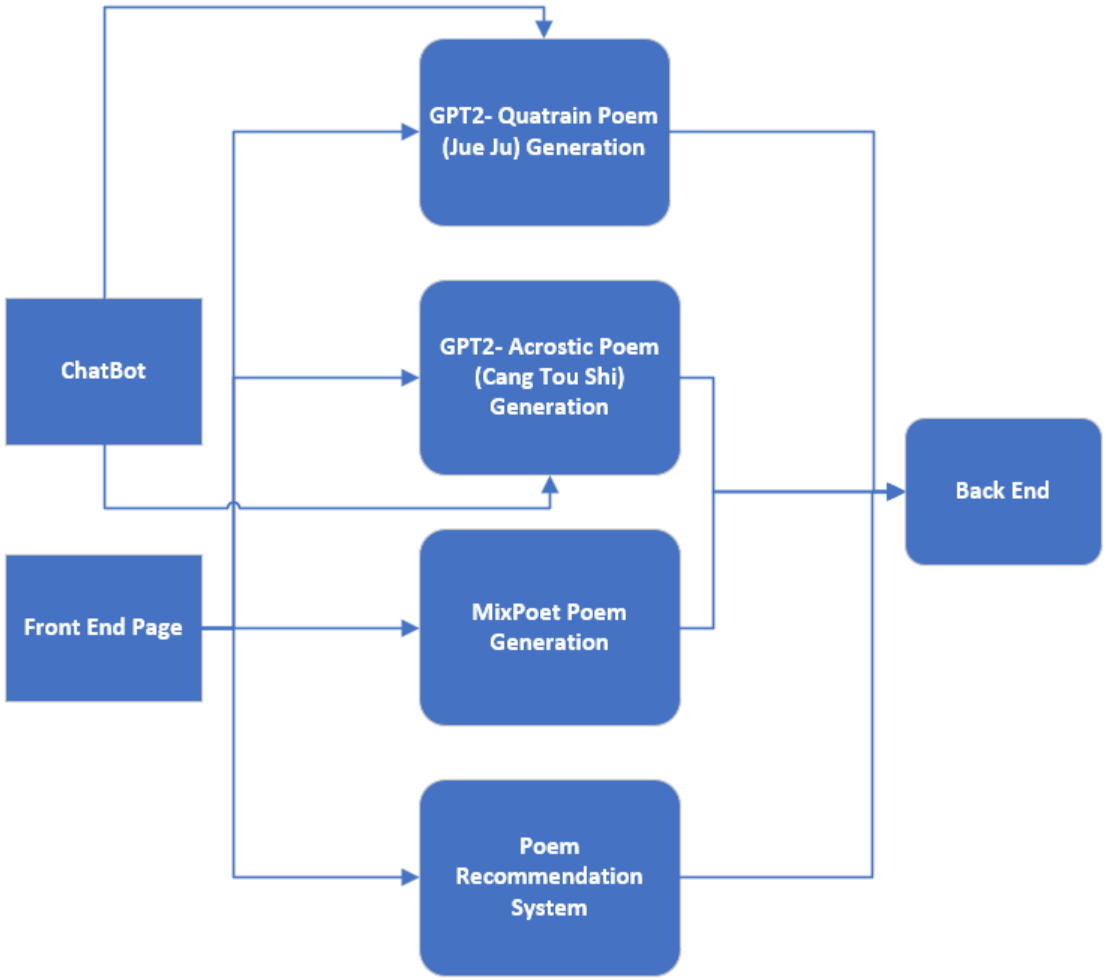

Figure 3: system architecture

5.1 Normal GPT2

GPT-2 has a generative pre-trained transformer architecture that implements a deep neural network, specifically a transformer model, which uses attention in place of previous recurrence and convolution-based architectures.

We choose to use GPT-2 as our base model. Because it is an open-source model, which allows us to

fine-tune it with our dataset of ancient poems to achieve our specific objectives. Besides, it is a powerful model that is easy to train, making it a practical choice for our poetry generation system.

5.2 GPT-2 acrostic poem generation model

Acrostic poetry is a type of Chinese poetry in which the first character of each line combines to form a specific phrase or message. We have over 550k Chinese poems as our input data. which contains different formats of poetry including five-character and seven-character quatrains and regulated verses.

We preprocess our ancient poetry dataset into the format of "form + [SEP] + title + [SEP] + content" and tokenize it using the BERT tokenizer. The "[SEP]" symbol is used as a separator to delineate different components of each poem, ensuring the model recognizes the structure and can process each part accurately.

The final model takes two inputs: The "head" of an acrostic poem, which typically refers to the initial characters that make up the vertical constraint in an acrostic. The format of the poem provides the structural guide for the poetry generation process.

5.3 GPT-2 quatrain poem generation model

Training using the gpt2 model with poem types combined with tagged style data. Data preparation: A dataset of ancient poems is collected, including different types and styles of ancient poems. Either use publicly available ancient poetry datasets or collect data from various sources. Make sure the dataset contains markers for the type and style of poems. Most of the poetry data needs to be tagged with its own style. Use the format to combine the type of poem + the style keywords + poem content and use it for fine-tuning the GPT2.

User Input: Users are prompted to provide a brief summary or idea of the poem they wish to write, along with several keywords that encapsulate the theme or mood of the desired poem. These keywords help in defining the context and directing the generation process. The summary and keywords are then passed to an AI model, specifically trained in understanding and generating poetic content. The AI model uses these inputs to grasp the user's idea and emotion behind the poem.

GPT2LMHeadModel is used to create the GPT-2 model and tokenization bert is used to split the words. The AI model generates multiple versions of the poem based on the provided inputs. Each version might approach the theme or mood slightly differently, offering the user a variety of choices.

Users review the generated poems and select the one that resonates most with their original intent. The model learns the laws and patterns of the input text and tries to generate output text that matches these laws and patterns. This includes the correspondence between length, content, and style when generating poems.

By providing enough samples in the training data and the corresponding labels (e.g., style, length, etc. of the poem), the model can be helped to learn these correspondences. During the training process, the model learns and adjusts to the input text and labels in order to make the generated text more consistent with the desired patterns and regularities.

5.4 Mixpoet model

Mixpoet, proposed by Yi [12], is a Chinese classical poem generation model, which is based on a semi-supervised variational autoencoder. Mixpoet can combine different topics, shown as keywords and different factors (sentiments), explained in the dataset section, to generate poems of different styles. Mixpoet

successfully solves the issue of diversity. For the usage of the variational autoencoder, the independence of the latent variable and influence factors is not assumed as poetic style and semantics are closely linked. Therefore, the latent space is divided into some subspaces and each subspace is corresponding to a factor by adversarial training. For the training section, Mixpoet can predict feasible factors of unlabelled poems and then apply a semi-supervised manner to train the model. In the testing section, we can select the specific values for each factor and then create various combinations of factor properties to generate poems with distinct styles. Poems generated by Mixpoet indicate two features: inter-topic diversity and intra-topic diversity. Mixpoet can automatically generate distinguishable poems with different keyword inputs, which is inter-topic diversity. By manually selecting the mixture of factors and using the same keyword, Mixpoet can create distinct poems which show the features of defined factors, performing intra-topic diversity.

In summary, adversarial training is employed to divide the latent space into subspaces to involve corresponding styles and create various poems with the usage of both desired topics and factors (sentiment). In addition, Mixpoet is semi-supervised and can use a small set of labeled data and a large amount of unlabeled data to train.

5.4.1 Model Preprocess

Before starting the usage of Mixpoet, the task should be formalized to analyze a poem. The poem is represented as x with n lines $x1, x2, ...xn$. The number of words in each line is l, and w is used as the keyword of a poem to represent the topic. There are m factors, $y1, y2, ..., ym$, and each factor y_i is divided into k_i classes. Based on these definitions, the labeled data and unlabelled data are defined as $p_l(x, w, y1, y2, ..., ym)$ and $p_u(x, w)$ respectively. We aim to create poems based on w (topic) and the mixture of factors of styles.

5.4.2 Basic Generator

A basic generator is used as one of the model baselines. s_i, j is defined as the corresponding GRU decoder hidden state. Then the probability distribution of each x_i, j for generation is used as

$$s_{i,j} = GRU(s_{i,j-1}, [e(x_{i,j-1}); g_{i-1}]) \tag{1}$$

$$s_{i,0} = f(e(w), o_i) \tag{2}$$

$$p(x_{i,j}|x_{i,<j}, x_{<i}, w) = softmax(f(s_{i,j})) \tag{3}$$

where [;] means concatenation; $e()$ means the embedding; $x_{<}i$ is the abbreviation of $x1, \hat{a}, x_{i-1}$; o_i is a special length embedding to determine the length of each line; f is a non-linear layer; g_{i-1} is applied to record so far created content in a poem as a context vector and preserve global information for the generator, which is employed to save context coherence.

5.4.3 Semi-supervised Conditional VAE

A latent variable z is used to form $p(x, y|w) = R\, p(x, y, z|w)dz$. Then, according to the subspace idea, $p(x, y, z|w)$ is computed as $p(x, y, z|w) = p(y|w)p(z|w, y)p(x|z, w, y)$. This formula indicates the process of poem generation: if users donât provide labels, the factor class can be predicted based on the keyword, and the model produces a value of z according to topic w and auto-generated factor class. Finally, a poem is generated. KL divergence is used to reconstruct the poem x.

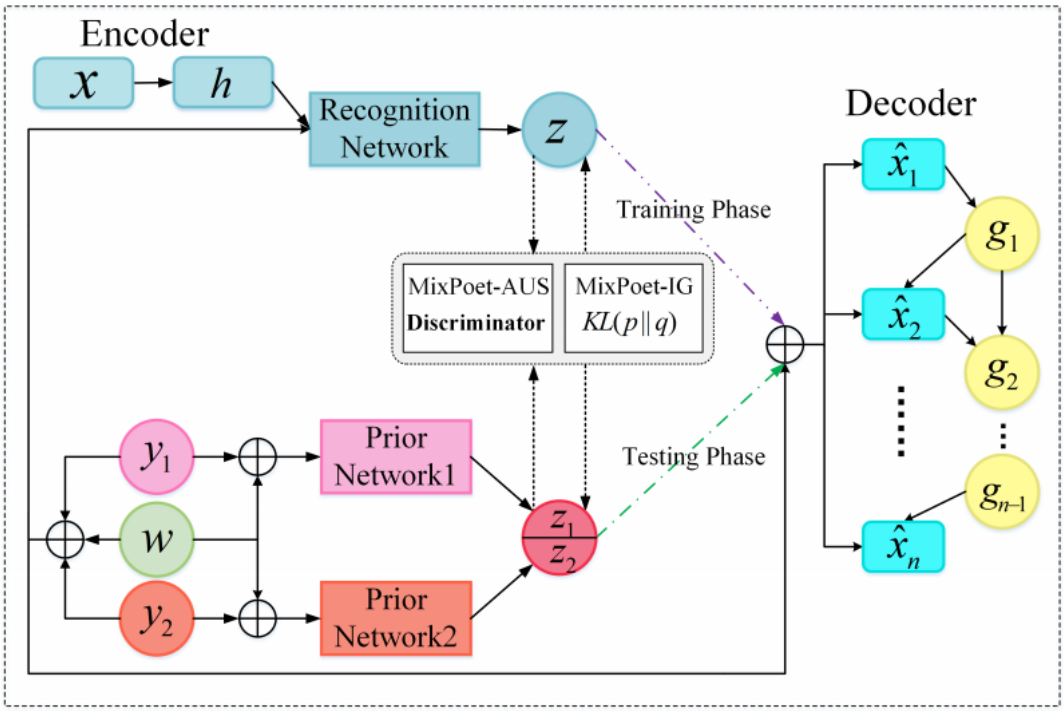

Figure 4: MixPoet Architecture

Figure 4 indicates the whole process of the MixPoet model. The poem x is used as a long sequence and a bidirectional GRU is applied to process x. Then the last forward and backward hidden states are concatenated to form h, which is the vector representation of x. Multiple Layer Perceptron (MLP) is considered to be the classifiers: $pw(y|w) = softmax(MLP(e(w)))$ and $qw(y|x, w) = softmax(MLP(e(w), h))$. $p\psi(x|z, w, y)$ is used as the decoder and the initial decoder state is set as $s_{i,0} = f(e(w), oi, z, e(y))$.

5.4.4 Latent Space Mixture

To provide the ability for the model to analyze multiple factors, the latent space is divided into 2 subspaces. Then the whole latent variable is formed from the subspaces, which combines the features of various factors. Two methods are applied to learn the integrated latent space: Mixture for Isotropic Gaussian Space, and Adversarial Mixture for Universal Space. For the first one, BOW loss is used to make z capture more global information. For the second one, a projection discriminator is used and spectral normalization is used to make the training process more stable.

For the practice with MixPoet, we have changed its hyper-parameters and some parts of layers to achieve poem generation with higher quality.

5.5 Chatbot

5.5.1 Intuition

The inclusion of a chatbot in our project aims to enhance the user experience by providing a more natural interaction, allowing users to input their queries through typing or speaking. Extensive research was conducted to develop and deploy a chatbot, including the implementation of speech recognition. However, it was realized that leveraging an existing platform would offer a more efficient solution. In this case, Google Dialogflow emerged as a suitable choice.

5.5.2 Design of Intents

Currently, we have successfully implemented the generation of Cang Tou Shi (acrostic poetry) and Jue Ju (quatrain). To enable the chatbot to fulfill users' needs effectively, it is necessary to define appropriate intents for capturing their inputs.

For Cang Tou Shi, the input requirement is a simple 4-character word. Consequently, we extract a single parameter from the given sentence, which corresponds to the word of 4 characters. Google Dialogflow simplifies this process by allowing us to define an intent for capturing the input. We provide several example sentences as training phrases and label the words as parameters within these sentences. Figure 5 is an example where a user asks the bot to write a Cang Tou Shi.

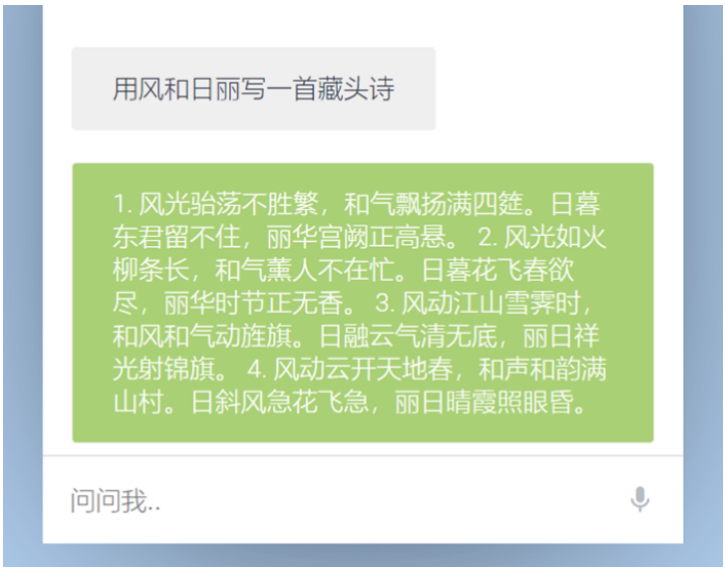

Figure 5: Cang Tou Shi exapmle in chatbot

Regarding Jue Ju, the input involves multiple parameters, including the type of line length (Wu Yan - 5 words per sentence, or Qi Yan - 7 words per sentence) and several keywords related to the desired theme of the generated poem. Figure 6 is an example where a user asks the bot to write a Jue Ju, given line length and keywords.

Figure 6: Jue Ju example in chatbot

While it is technically possible to extract multiple arguments from a sentence, it is not user-friendly to require a lengthy sentence that includes all parameters. To address this, the use of follow-up intents as a sequence of questions, each capturing one parameter, was considered. However, this approach lacks flexibility since users may not always provide parameters in the expected sequence. Consequently, a more straightforward and flexible approach was adopted. Instead of using follow-up intents, we provide various training phrases, some of which include all arguments, while others include only a subset. All occurrences of parameters are marked as "required," and preset prompts are provided for missing parameters. Figure 7 indicates some parameters of chatbot.

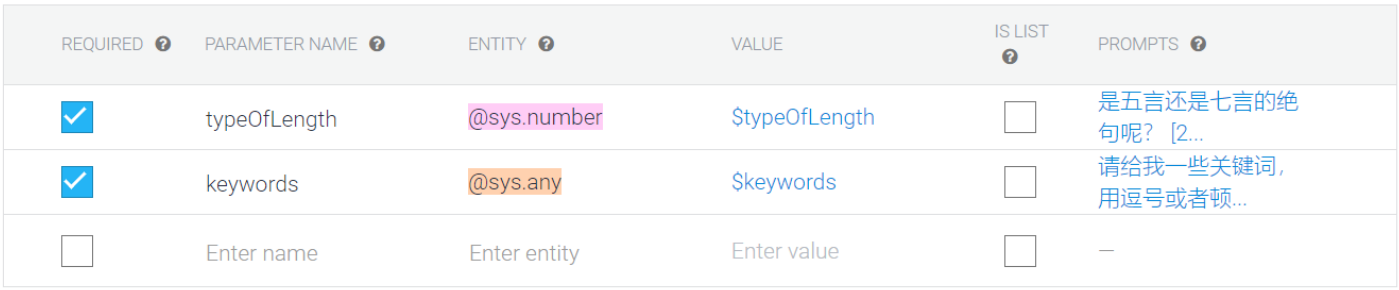

Figure 7: Some parameters in chatbot

5.5.3 Rectification of User Input

In some cases, the keywords extracted may require correction due to voice input misrecognition or a change in the user's intent. After the entity extraction step, it is essential to present users with the extracted results and allow them to verify their accuracy. Figure 8 indicates the rectification function of chatbot

Figure 8: Example of the rectification function of chatbot

To facilitate this process, follow-up intents are utilized to differentiate between "yes" and "no" responses. If the response is affirmative, these parameters are passed to the backend model for further processing. However, if the response is negative, an alternative intent is invoked to handle the rectified input.

In Dialogflow, contexts serve as a means to keep parameter values and enable their transfer between intents. The key to obtaining rectified input lies in the definition of a Context that establishes a link between the follow-up "no" intent and the intent handling the rectified input.

Within the follow-up "no" intent, the input context is set as the automatically generated "follow-up" context, which contains the type of line length and keywords related to the themes. Additionally, we define an output context called "rectify-keyword". Figure 9 shows the 'rectify keywords'.

Figure 9: Example of 'rectify keywords' in our chatbot

Subsequently, another intent is created to receive the rectified input. In this intent, the contexts are reversed. The input context becomes the "rectify keyword," while the output context becomes the "followup" context. This allows the capture and processing of new parameter values by existing intents. The image above shows the reverse of contexts with two intents.

By employing this methodology, users are provided with a mechanism to rectify their inputs when necessary, ensuring accurate and tailored responses from the chatbot.

5.6 Poem Recommendation System

In addition to generating ancient poems based on user input, we also offer an additional feature. This feature recommends similar poems from our database of ancient poetry, based on the user's input. Here, we face two technical challenges: 1. What method should we use to match the ancient poems in the database with the user's input? 2. Our ancient poetry database contains more than 550,000 poems. How can we calculate the poems to be recommended in a short period of time? In our solution, we utilize Elasticsearch, BERT, and cosine similarity calculations.

5.6.1 Data Processing

We employ Elasticsearch to support our big data processing tasks. Elasticsearch is a highly scalable open-source full-text search and analytics engine. It enables us to store, search, and analyze big volumes of data quickly and in near real time. One of its strengths is that it offers robust RESTful APIs and a query DSL, making it flexible to work with.

The BERT model we employ is BERT-CCPoem [13], an adaptation of the BERT model specifically trained for Chinese classical poems. This model was developed by the Research Center for Natural Language Processing, Computational Humanities, and Social Sciences at Tsinghua University. BERT-CCPoem is trained on an extensive collection of Chinese classical poems, specifically the CCPC-Full v1.0 dataset, which consists of 926,024 classical poems with 8,933,162 sentences. This pre-training on a large and relevant dataset makes BERT-CCPoem highly effective for our use case of recommending ancient Chinese poems.

In our application, we first create an index in Elasticsearch named "poems", and define several fields for this index including "title", "dynasty", "author", "poem content", and "vector". We utilize the BERT model for Chinese text, specifically 'bert-base-chinese', to tokenize the poems and generate vector representations for each word and the entire poem. These vector representations alongside the other fields of the poems are then stored in Elasticsearch. Figure 10 indicates the index count

Figure 10: Elasticsearch Poem index count

5.6.2 Recommendation

During the recommendation stage, we also use the BERT-CCPoem model to tokenize and vectorize the user's input. Subsequently, we leverage Elasticsearch's cosine similarity method to calculate the cosine similarity between the user's input vector and the vectors of the poems in the database. The top 5 poems with the highest similarity scores are selected and returned as the recommendation results.

6 Test results demonstrating the performance of your systems

6.1 The result of GPT-2 acrostic poem (Cang Tou Shi) generation model

Figure 11 indicates Cang Tou Shi with five-character quatrain, while figure 12 indicates ang Tou Shi with seven-character quatrain.

The model performs well, generating corresponding poetry based on the format and acrostic provided by the user.

6.2 The result of quatrain(Jue Ju) poem generation model

Figure 13 indicates Jue Ju with five-character quatrain, while figure 14 indicates Jueju with seven-character quatrain

The model combines this set of keywords and generates verses based on the learned rules and structure of the 5 or 7words a sentence JUEJU.

I believe the model performs well in JUEJU 7 words a sentence, the reason is the dataset of 7 words a sentence JUEJU is bigger. And the model fits better during training.

Figure 11: Cang Tou Shi, five-character quatrain

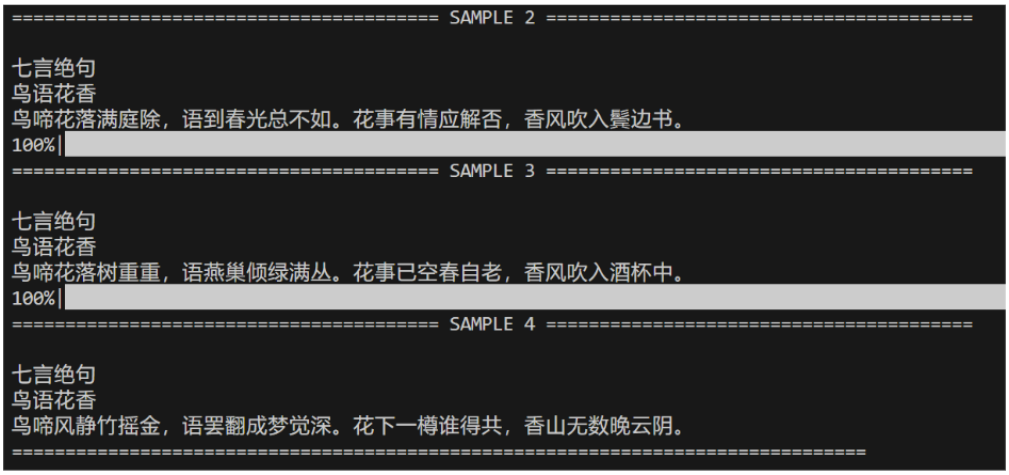

Figure 12: Cang Tou Shi, seven-character quatrain

6.3 The result of the MixPoet poem generation model

For MixPoet, we can input a keyword, specify the length of each line (the number of words in each line), and specify the living experience label and historical background label. A poem that satisfies our requirements will be generated. An example of poem generation is shown in the figure 15.

6.4 Poem Recommendation System

Figure 16 shows the results of poem recommendation system.

Based on the content input by the user, the model provides the user with the top 5 classical poems with the highest similarity. We can intuitively see that the content of the poems corresponds with the user's input. Additionally, due to Elasticsearch's powerful indexing capabilities, the computation speed is quite fast, and it can retrieve matching poems from 550,000 classical poems within three seconds.

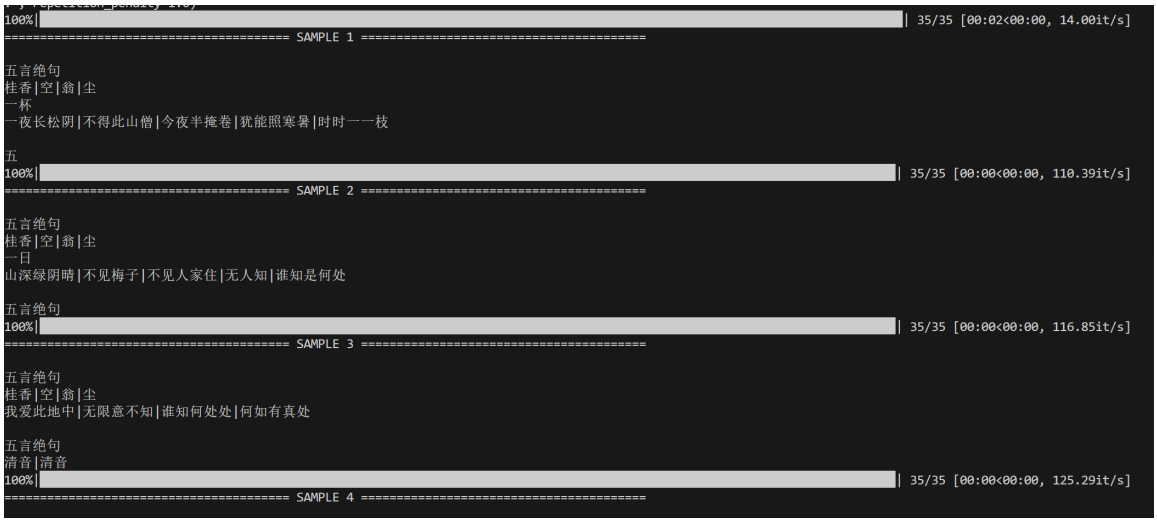

Figure 13: Jue Ju, five-character quatrain

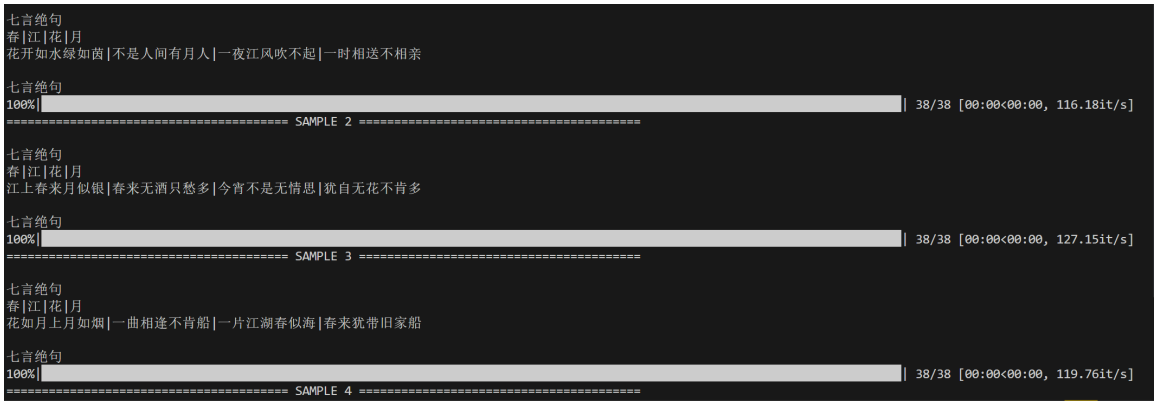

Figure 14: Jue Ju, five-character quatrain

6.5 Front End Page

Figure 17 is the front end page to operate our Bard system.

We use Streamlit to build an app, providing a web-based graphical interface for users. Streamlit is a flexible and easy-to-use framework that allows developers to create apps for machine learning and data science projects.

In our frontend UI, users can select the generation model(Acrostic, Keywords, Mixpoet and format to use (five-character, seven-character, quatrains, regulated verse), specify the number of classical poems to be generated, and designate inputs (acrostics, keywords). The generated results will then be displayed on the page, along with the recommended results obtained from our recommendation system based on user input.

6.6 Evaluation

The evaluation of a Chinese ancient poetry generation model plays a crucial role in assessing its performance and ensuring its effectiveness. It allows us to quantify the model's capabilities and measure its ability to generate high-quality and coherent ancient Chinese poems. By evaluating the model, we can gain insights into its strengths and weaknesses, identify areas for improvement, and refine its performance through iterative development. To comprehensively assess the model's performance, we utilize both automated and human evaluation methods.

```
input a keyword:>山河
specify the length, 5 or 7:>7
specify the living experience label
          0: military career, 1: countryside life, 2: other:, -1: not specified>1
specify the historical background label
          0: prosperous times, 1: troubled times, -1: not specified>0

generating step: 0

generating step: 1

generating step: 2

generating step: 3
石桥流水碧潺湲
万壑松风响佩环
借问仙源何处去
青山如在白云还
```

Figure 15: MixPoet result

```
Input: 战争 沙场
Title: 后凯歌词九首 其五 幝舍, Dynasty: 元, Author: 耶律铸, Content: 畔敌休矜战骑多，纷罗区落遍山坡。那知未鼓投戈地，待阅前徒竟倒戈。
Title: 嘉禾百咏 其二十七 四城, Dynasty: 宋, Author: 张尧同, Content: 吴越争雄日，区区在用兵。空馀争战地，无处不高城。
Title: 旧边诗九首 其九 甘肃, Dynasty: 清, Author: 方还, Content: 风急荒原落雁声，西河霜气逼严城。金笳几处秋乘障，铁马连群夜点兵。充国留屯沙际没，嫖姚遗垒月中明。古来无限安边策，哈密徒劳苦战争。
Title: 游龙门山（三首） 其二, Dynasty: 近现代, Author: 洪存恕, Content: 东都历历几兴亡，形胜山川总战场。巨阙枉多貔虎集，今馀沙草过牛羊。
Title: 老将, Dynasty: 明, Author: 林廷玉, Content: 摧锋皱敌手挥戈，矍铄浑如马伏波。牙帐连云屯朔漠，金笳吹月度关河。谋谟久练兵机熟，危脸曾经战阵多。净却狼烟人乐业，沙场闻唱太平歌。
```

```
Input: 月明星稀，乌鹊南飞
Title: 书寿昌驿, Dynasty: 宋, Author: 程俱, Content: 岁暮白日速，风高黄叶稀。归心与寒雁，一夜向南飞。
Title: 寄丁元珍, Dynasty: 宋, Author: 刘敞, Content: 南海飞鸿北海鸟，青冥相望羽翰孤。虚弓祇是弦声急，不向云霄堕九乌。
Title: 从军行五首 其五, Dynasty: 明, Author: 徐祯卿, Content: 青天碛路挂金微，明月洮河树影稀。胡雁哀鸣飞不度，黄云戍卒几时归。
Title: 燕京四时歌 其三 秋, Dynasty: 明, Author: 徐祯卿, Content: 蓟门桑叶落濛沱，代北浮云鸿雁多。莫向云中传尺素，空将明月对嫫蛾。
Title: 春日寄许浑先辈, Dynasty: 唐, Author: 杜牧, Content: 蓟北雁初去，湘南春又归。水流沧海急，人到白头稀。塞路尽何处，我愁当落晖。终须接鸳鹭，霄汉共高飞。
```

Figure 16: Results of poem recommendation system

Automated evaluation techniques involve the use of quantitative metrics, such as rhyme accuracy, and vocabulary repetition rate statistics to measure the model's adherence to the rules and structure of ancient Chinese poetry. These metrics provide objective and reproducible assessments of the model's performance.

In addition to automated evaluation, human evaluation plays a crucial role in capturing the nuanced aspects of poetic quality that may be challenging to quantify with automated metrics alone. Human evaluators, possessing an understanding of ancient Chinese poetry, assess the generated poems based on criteria such as Poetic coherence, artistic conception, and intention. Their subjective judgments provide valuable insights into the aesthetic appeal of the generated poems.

6.6.1 Automated evaluation

The automated evaluation of the Chinese ancient poetry generation model comprises several metrics, including rhyme accuracy, vocabulary repetition rate, keyword coverage rate, and keyword distribution rate.

Rhyme accuracy measures the proportion of correctly rhymed sentences within the entire poem. A higher rhyme accuracy indicates that the model performs better in learning the rhyming patterns of Chinese poetry.

Vocabulary repetition rate refers to the ratio of repeated words to the total number of words in the generated poem. A lower vocabulary repetition rate suggests that the model exhibits greater lexical richness and creativity.

The keyword coverage rate assesses whether the generated poem includes the user-specified input criteria. A higher keyword coverage rate indicates that the model can accurately understand the user's input and create poetry based on the given criteria.

Keyword distribution rate measures how evenly the user-specified keywords are distributed throughout

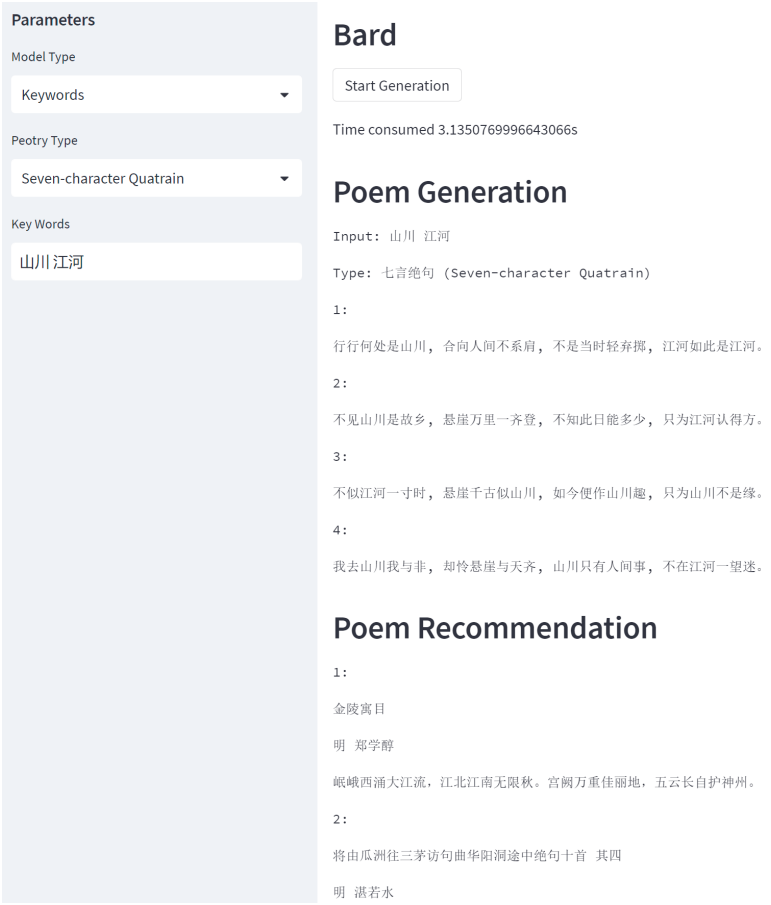

Figure 17: Frontend

the poem. It evaluates whether the keywords are appropriately integrated into different parts of the poem. A higher keyword distribution rate signifies that the model can evenly allocate the specified keywords across the poem.

In this evaluation, we assessed the performance of three models: Mixpoet, GPT2 Quatrain, and GPT-2 Acrostic by the above metrics. Each model generated a set of 200 poems. Additionally, we conducted an automated evaluation using the same metrics on a dataset consisting of 50,000 existing poems. The evaluation results of this dataset were considered as the ground truth against which we compared our models. The table 1 presents our evaluation results.

Model	Rhyme Accuracy	Vocabulary Repetition Rate	Keyword Coverage	Keyword Distribution Rate
Mixpoet	0.30	0.05	1	1
GPT-2 Acrostic	0.37	0.05	1	1
GPT2 Quatrain overall	0.40	0.07	0.99	0.97
GPT2 Quatrain few keywords	0.42	0.11	1	1
GPT2 Quatrain more keywords	0.38	0.03	0.98	0.94
GT	0.63	0.02	NA	NA

Table 1: Automatic evaluation results

During the evaluation process, we observed an interesting trend in the performance of GPT2 Quatrain regarding vocabulary repetition rate. Specifically, we noticed that when the input keywords were relatively few, the vocabulary repetition rate was relatively high. However, when the number of input keywords

increased, the vocabulary repetition rate exhibited a significant decrease. To gain a clearer understanding of this behavior, we decided to analyze these two scenarios separately.

During the evaluation, we also observed that the keyword coverage rate and keyword distribution rate metrics did not exhibit significant meaning when the number of input keywords was relatively low (one or two keywords). Specifically, when using the Mixpoet model with only one input keyword, or when GPT2 Quatrain had a small number of input keywords, these metrics did not provide substantial insights. On the other hand, GPT-2 Acrostic consistently achieved a high keyword coverage rate due to its adherence to the rules of generating acrostic poems.

Similarly, the keyword distribution rate metric did not offer significant insights in these cases. As GPT2 Quatrain and Mixpoet had a limited number of input keywords, the distribution of these keywords throughout the poems did not reveal meaningful variations.

Considering these observations, we can conclude that for the specific cases where only one or a few keywords were used, metrics such as rhyme accuracy and vocabulary repetition rate held more meaningful implications. These metrics provided valuable information about the models' ability to maintain rhyming patterns and generate diverse vocabulary, respectively.

6.6.2 Human evaluation

In addition to automated evaluation, we conducted a human evaluation to assess the Chinese ancient poetry generation models based on several key criteria: coherence, imagery and metaphor, emotional expression, and thematic content.

Coherence refers to the smoothness and logical consistency between the lines and stanzas of the poems. Evaluators examined how well the lines and stanzas were connected, as well as the overall coherence within the context of the poem.

Imagery and metaphor evaluation focused on the vividness and imaginative qualities of the poems. Evaluators observed how effectively the poems conveyed vivid and compelling imagery, as well as the presence of skillful use of metaphorical language. They assessed whether the poems evoked sensory experiences and painted vivid mental pictures for the readers.

Emotional expression evaluation aimed to determine the poems' ability to convey and evoke emotions. Evaluators assessed the extent to which the poems effectively expressed specific emotions and elicited emotional resonance in the readers.

The thematic content evaluation focused on the clarity, and depth of the poems' themes and overall content. Evaluators examined whether the poems had discernible and well-developed themes, as well as whether they conveyed meaningful insights, profound meanings, and thought-provoking ideas.

In the human evaluation phase, we randomly selected 20 poems generated by each of the three models, resulting in a total of 60 poems. These poems were distributed to volunteers in the form of a questionnaire, where the volunteers were asked to rate the poems on the four evaluation criteria mentioned earlier: coherence, imagery and metaphor, emotional expression, and thematic content. Each criterion was assigned a maximum score of 25 points, allowing for a comprehensive assessment of the poems' quality.

Due to time constraints, we were able to collect a total of 75 completed questionnaires from the volunteers. The table 2 presents the results of the human evaluation, indicating the average scores given by the volunteers for each model on each evaluation criterion.

Model	Coherence	Imagery and Metaphor	Emotional Expression	Thematic Content	Total Score
Mixpoet	18.3	20.7	18.8	22.3	80.1
GPT2 Acrostic	12.1	17.0	16.8	14.3	60.2
GPT2 Quatrain	21.7	22.1	17.4	17.7	78.9

Table 2: Results of the human evaluation

7 Conclusion

In this project, after reading a lot of literature and searching a lot of related materials, we settled on using GPT2 and Mixpoet to build our Bard system to deal with the ancient poem generation problem. In addition, we create a poetry recommendation system based on the BERT algorithm and Elasticsearch, in which the cosine similarity calculation is applied. For the GPT2, we also make fine-tuning to create of 2 submodels, one for Quatrain poem generation and another for acrostic poem generation. For the GPT2, we have successfully generated desirable quatrain poems and acrostic poems according to keywords. For the MixPoet, based on keyword input and sentiment selections, poems with high artistic value can be created. Besides, the poem recommendation system has successfully found poems matching the keywords from users. In general, our system is well designed and feasible.

8 Future Outlook

Data preparation: Mark the style of each verse or the style of the whole and generate it using the summary model. A dataset containing the content of the poems and the corresponding style markers is collected. For each poem, the style markers for each verse can be used as additional input information, or the overall style markers can be used as part of the poem. I believe that the existing data is relatively sufficient for training the summarizer(BERT, GPT).

The GPT2 model can also consider training for generating the title of the poem and the emotion. But finding the data is much more difficult, and if the input of the model is too much, information will lose while generated. So that needs more data, more labels.

MixPoet can generate poems with high artistic value and rhymes that fit the rules. However, sometimes MixPoet will generate poems that are not closely linked to keywords. We think the main reason is the limitation of the labeled dataset. The author of this model labeled around 50000 poems to train the model, achieving high quality and coherence. But the dataset published by the author plus our manually labeled dataset has only about 10,000 poems. Manually labeling a dataset is a very time-consuming and labor-intensive task. So next, we will try to find some methods to label datasets automatically to train the MixPoet again.

9 Assistance of ChatGPT

Use ChatGPT to explore ideas, ask questions to explore the feasibility, clarify concepts, and learn about knowledge points. Code error correction, used to analyze code error reporting. Thesis understanding, use it to search for some content in the thesis to speed up the time spent to understand it and to search for other related knowledge and links to the thesis.

References

[1] Natural Language Processing Lab at Tsinghua University. Jiuge, 2022. Accessed: 2023-05-20. URL: `https://github.com/THUNLP-AIPoet`.

[2] Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. Chinese poetry generation with planning based neural network. *arXiv preprint arXiv:1610.09889*, 2016.

[3] 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 *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013.

[4] Naoko Tosa, Hideto Obara, and Michihiko Minoh. Hitch haiku: An interactive supporting system for composing haiku poem. In *Entertainment Computing-ICEC 2008: 7th International Conference, Pittsburgh, PA, USA, September 25-27, 2008. Proceedings 7*, pages 209–216. Springer, 2009.

[5] Xiaofeng Wu, Naoko Tosa, and Ryohei Nakatsu. New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system. In *Entertainment Computing–ICEC 2009: 8th International Conference, Paris, France, September 3-5, 2009. Proceedings 8*, pages 191–196. Springer, 2009.

[6] 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.

[7] Hugo Gonçalo Oliveira. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence*, 1:21, 2012.

[8] Erica Greene, Tugba Bodrumlu, and Kevin Knight. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 524–533, 2010.

[9] 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.

[10] 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*, pages 1183–1191, 2016.

[11] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[12] 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 Thirty-Fourth AAAI Conference on Artificial Intelligence*, New York, USA, 2020.

[13] Natural Language Processing Lab at Tsinghua University. Bert-ccpoem, 2022. Accessed: 2023-05-20. URL: `https://github.com/THUNLP-AIPoet/BERT-CCPoem`.