Exploring Deep Learning Architectures for Chinese Ancient Poetry Emotion Classification

Zhouhongyuan Hu *

College of Computer Science, Sichuan University, Sichuan, Chengdu, 610207, China.

Abstract

Emotion classification is a crucial research area in natural language processing. However, the complexity of emotions expressed in Chinese classical poetry, combined with the unique grammar and syntax of the language, poses a significant challenge for current models used in modern Chinese emotion classification tasks. In this paper, we propose a novel method for automatically annotating emotions in classical Chinese poetry using a human-annotated dataset and fine-tuning the BERT-CCPoem pre-training model. To achieve accurate emotion annotation, we develop a hierarchical term classification system comprising seven primary categories and 23 subcategories, which is used in a coarse-to-fine approach to fine-tune the BERT-CCPoem model on each dataset. A fully connected layer is added to the output of BERT-CCPoem for classification. Our experimental results demonstrate that our method outperforms other deep learning methods in terms of emotion classification accuracy, robustness, and scalability. Our proposed method can be applied not only to ancient Chinese texts but also to other types of texts.

Keywords

Chinese classical poetry, deep learning, emotion classification.

1. Introduction

The ancient Chinese poetry, a treasure of human culture, has a long and rich history[1], carrying abundant historical and cultural connotations. With millions of enthusiasts and creators worldwide, ancient Chinese poetry is not only widely circulated in East Asia, including China, Japan, and Korea, but also has a certain influence in Europe and America. Emotion expression is undoubtedly the most important aspect of ancient Chinese poetry, with its complex and diverse emotions broadly categorized into seven aspects, including farewell, love and resentment, homesickness, landscape and pastoral, chanting things, history-nostalgia and frontier wars. Therefore, sentiment analysis of ancient Chinese poetry is of great significance for understanding it.

However, sentiment analysis of ancient Chinese poetry text is inherently a complex task. Traditional sentiment analysis methods typically rely on bag-of-words or n-gram models, which treat each word or phrase in the text as an independent feature and classify the text using some classification algorithms such as Naive Bayes and Support Vector Machine. However, this approach faces challenges when applied to ancient Chinese poetry and other poetic texts.

Firstly, ancient Chinese poetry often employs rhetorical devices, such as metaphor, allusion, and symbolism, which endow words with not only literal meanings but also deeper connotations. Traditional sentiment analysis methods struggle to capture these more complex linguistic features.

Secondly, ancient Chinese poetry often contains polysemous words, such as "falling autumn leaf", which could refer to natural landscapes or express emotional nuances[2]. Traditional sentiment analysis methods usually treat each word as an independent feature and cannot
distinguish these different meanings. Furthermore, ancient Chinese poetry text is usually short, and the sentiment tendency is often subtle, making it difficult for traditional machine learning methods to accurately capture sentiment information from it.

With the development of technology, pre-trained language models with large parameters can learn the context of the text well. This makes it possible to obtain an accurate sentiment classification model for ancient Chinese poetry. In this paper, we propose a method for automatic sentiment annotation of ancient Chinese poetry based on the BERT-CCPoem pre-training model fine-tuning. We manually classify each ancient Chinese poem and use these data to fine-tune the BERT-CCPoem pre-training model to achieve sentiment classification of ancient Chinese poetry. The purpose of this paper is to propose a new and efficient solution for sentiment analysis of ancient Chinese poetry and to explore the effectiveness and practicality of this approach.

In this paper, we make the following contributions:
1. We develop a hierarchical emotion classification system, consisting of seven main categories and 23 subcategories, which can cover the common emotion types in classical Chinese poetry, and can reflect the complexity and diversity of emotions in classical Chinese poetry.
2. We adopt a coarse-to-fine approach to fine-tune BERT-CCPoem, that is, we first fine-tune it on the dataset of main categories, and then fine-tune it on the dataset of subcategories. This can improve the generalization ability and classification accuracy of the model.
3. We conduct experiments on a dataset that we built ourselves, which contains classical Chinese poems from different dynasties, forms, and styles. The results show that our method outperforms other deep learning methods in terms of emotion classification accuracy, robustness, and scalability.

2. Models Considered

In a recent study by Zhang Lingli et al.[3], a novel model for sentiment analysis of poetry was constructed by fusing BERT-wwm-ext and ERNIE[11] models to obtain word vectors, extracting features through BiLSTM[12] and TextCNN[13], and utilizing BiGRU to fuse the features. However, despite the promising results, the data source for this poetry sentiment analysis model remains the THUNLP-AIPoet dataset[4], and the sentiment labels are still limited to traditional polarity labels, such as positive, neutral, and negative[4].

Nevertheless, as the emotions conveyed in poetry are exceedingly delicate, the sentiment expressed in different types of poetry can be significantly nuanced, even if they share the same polarity. For instance, the sentiment in a patriotic poem and a pastoral poem may both be positive, but their connotations are vastly different. Therefore, for most people, the sentiment in poetry that they care about is by no means limited to simple polarity sentiment. Rather, they seek richer and more subtle emotions.

Experts have identified seven major categories and 23 subcategories of emotions in poetry, highlighting the need for sentiment analysis models that can capture the intricacies of poetic expressions. (see Table 1)

<table>
<thead>
<tr>
<th>Large class labels</th>
<th>Small class labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>farewell</td>
<td>nostalgic farewell</td>
</tr>
<tr>
<td>express ambition</td>
<td>scenery and lyricism</td>
</tr>
<tr>
<td>love and resentment</td>
<td>boudoir grievance</td>
</tr>
</tbody>
</table>

Table 1: Sentiment Granularity
This study aims to use deep learning methods to achieve automatic labeling of more subtle emotions in poetry. That is, given the text content of a poem, the model will output the major category of emotion and subcategory of emotion expressed in the poem. The models will be trained on both major and subcategories of emotion, from coarse to fine classification, in order to achieve automatic classification of sentiment in poetry.

We explored the prediction performance of three Deep Learning models based on:
BERT-CCPoem(BCC): BERT-CCPoem [5] is a BERT-based pre-training model designed for vector representation and downstream applications of Chinese classical poetry. It was developed by the Research Center for Natural Language Processing and Social Humanities Computing, Institute of Artificial Intelligence, Tsinghua University. It uses a corpus CCPC-Full v1.0 that contains almost all Chinese classical poems for training. The corpus has a total of 926,024 poems and 8,933,162 lines. It can provide a vector representation of any Chinese classical poem, which can be used for tasks such as intelligent retrieval, recommendation, and sentiment analysis.

Bert-base-Chinese(BBC): BERT-base-Chinese [6, 7] is a pre-trained language model based on Transformer, designed to address various tasks in Chinese natural language processing. It uses large-scale unlabeled text data to learn universal language representations, allowing for fine-tuning in various downstream tasks. BERT-Base-Chinese has billions of parameters and 12 hidden layers, allowing for a deep understanding and extraction of the final representation of Chinese text. Additionally, BERT-Base-Chinese uses bidirectional encoding to enhance the model’s semantic representation ability. Experimental results show that BERT-base-Chinese achieves state-of-the-art performance in various Chinese natural language processing tasks, including language inference, named entity recognition, sentiment analysis, and more. This
model has not only had a wide impact in academia, but has also been widely applied in various practical scenarios in industry.

GRU: Gated Recurrent Unit (GRU) [8] is a variant of Recurrent Neural Networks (RNN) [9], proposed by Cho et al. in 2014. GRU aims to address the problem of vanishing and exploding gradients that exist in standard RNN. Compared to traditional RNN, GRU introduces two gates (reset gate and update gate) to control the information flow in RNN. The reset gate determines whether to consider the previous hidden state in the current state, while the update gate controls how much information from the previous hidden state is included in the current state. By controlling these two gates, GRU can better handle the information transfer of long sequential data, and has been widely used in natural language processing, speech recognition and other tasks.

3. Methodology

3.1. Model Parameters and Hyperparameters

Hyperparameters are parameters that need to be manually specified in the deep learning model, which can affect the training effect of the model. In this article, we used three different models for experiments, namely BCC, Bert-base-Chinese and GRU. The following are the hyperparameter settings for each model:

**BCC:** We used the pre-trained BCC model for Fine-tuning. We specified a maximum sequence length of 512, a training batch size of 8, an Adam optimizer for training, a learning rate of 1e-6, no weight decay, no dropout, and a maximum number of training rounds of 50 epochs.

**BBC:** We used the BBC pre-trained model for Fine-tuning. We specified a maximum sequence length of 512, a training batch size of 8, an Adam optimizer for training, a learning rate of 1e-6, no weight decay, no dropout, and a maximum number of training rounds of 50 epochs.

**GRU:** We implemented a GRU-based sentiment classification model using the Pytorch deep learning framework. We set the embedding vector dimension is 512, the dimension of the GRU layer is 512, the number of GRU layers is 2, the Adam optimizer is used, the learning rate is 0.001, no dropout is used, and the maximum number of training rounds is 50 epochs.

All models use cross entropy as loss function. Specifically, assuming we have $K$ classes, for the $i$-th class, the predicted probability by the model is $\hat{y}_i$, and the true label is $y_i$. The formula for CrossEntropyLoss is:

$$\text{CrossEntropyLoss} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{K} y_{n,i} \log(\hat{y}_{n,i})$$

(1)

3.2. Dataset

All poetry texts used in this experiment are from the Chinese-poetry[14], and the emotion labels corresponding to the poems are manually annotated. Among them, there are 5812 poems with large category emotion labels, 1378 poems with small category emotion labels in the "farewell" category, 375 poems with small category emotion labels in the "love and resentment" category, 387 poems with small category emotion labels in the "homesickness" category, 928 poems with small category emotion labels in the "landscape and pastoral" category, 1329 poems with small category emotion labels in the "chanting things" category, 445 poems with small category emotion labels in the "history-nostalgia" category, and 479 poems with small category emotion labels in the "frontier wars" category.

Among them, the ratio of training set to validation set to test set is 6:2:2.

For convenience, in the following part of this article, A is used to represent the total category that has not been divided for the first time, B is used to represent the "farewell" category, C is
used to represent the "love and resentment" category, D is used to represent the "homesickness" category, E is used to represent the "landscape and pastoral" category, F is used to represent the "chanting things" category, G is used to represent the "history-nostalgia" category, H is used to represent the "frontier wars" category.

3.3. Choice of Comparison Metrics and Hypothesis Tests

In the performance evaluation of multi-classification models, some specific indicators need to be used. The following are four commonly used indicators:

Accuracy: Indicates the proportion of the number of samples predicted by the model to the total number of samples, the calculation formula is:

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN}
\]  

(2)

Precision: Indicates how many of the samples predicted as a certain class actually belong to this class, and the calculation formula is:

\[
Precision = \frac{TP}{TP + FP}
\]

(3)

The precision rate reflects the accuracy of the model's judgment on positive examples, and the higher the better. For multi-classification problems, it is necessary to calculate the accuracy rate for each category separately, and then average or weighted average to obtain the total accuracy rate.

Recall: Indicates how many of the samples that actually belong to a certain class are predicted to be of that class. The calculation formula is:

\[
Recall = \frac{TP}{TP + FN}
\]

(4)

The recall rate reflects how well the model covers the positive examples, the higher the better. For multi-classification problems, it is necessary to calculate the recall rate for each category separately, and then calculate the average or weighted average to obtain the total recall rate.

F1-score: Indicates the harmonic mean of precision and recall, the calculation formula is:

\[
F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall}
\]

(5)

The F1 value comprehensively considers two aspects of precision and recall, the higher the better. For multi-classification problems, it is necessary to calculate the F1 value for each category separately, and then calculate the average or weighted average to obtain the total F1 value.

In this paper, we use Accuracy and F1-score among the four commonly used evaluation indicators to evaluate the performance of our proposed emotion classification model on the ancient poetry emotion classification task. All evaluation metrics are calculated by direct averaging.
4. Results and Analysis

4.1. Which deep learning model is most effective for sentiment classification of ancient poetry?

This study uses three models, namely BCC, BBC and GRU, for sentiment analysis on poetry. To compare the performance of the three models, we selected the Accuracy and F1 models. After collecting enough data, we tested the three models separately and got their performance on each indicator. The specific results are shown in Table 2.

<table>
<thead>
<tr>
<th>Class</th>
<th>Accuracy</th>
<th>F1-score</th>
<th>Accuracy</th>
<th>F1-score</th>
<th>Accuracy</th>
<th>F1-score</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCC</td>
<td>0.8602±0.0082</td>
<td>0.7580±0.0122</td>
<td>0.7634±0.0095</td>
<td>0.6564±0.0188</td>
<td>0.6261±0.0108</td>
<td>0.4834±0.0128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBC</td>
<td>0.7554±0.0184</td>
<td>0.6512±0.0229</td>
<td>0.6789±0.0244</td>
<td>0.5933±0.0149</td>
<td>0.8088±0.0083</td>
<td>0.7394±0.0075</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRU</td>
<td>0.7469±0.0491</td>
<td>0.7439±0.0470</td>
<td>0.6724±0.0416</td>
<td>0.6558±0.0441</td>
<td>0.3214±0.0379</td>
<td>0.3155±0.0298</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We can find that the BCC model performs well in the emotion classification task of ancient poetry, which may be because it combines the characteristics of the CCPoem dataset and the advantages of the BERT model. The CCPoem dataset contains a large number of ancient poems, and the BERT model can learn the relationship between sentences and context. Therefore, the BCC model can better capture the emotional features in ancient poems, thus improving the performance of sentiment classification.

First, we compare the accuracy of different models on each emotion category, as shown in Figure 1. It can be seen that for most emotional categories, the BCC model has achieved the best performance, indicating that it can better capture the emotional information of ancient poems. However, for the sentiment category “farewell”, the GRU model performed best. We analyze the reason, which may be because poetry works of this emotional category tend to use relatively clear vocabulary and language to express emotions, and the GRU model has certain advantages in dealing with short texts and emotional vocabulary.

Next, we performed F1-score comparisons on the performance of different emotion categories, as shown in Figure 1. It can be seen that, similarly, the BCC model does not perform as well as GRU on the classification task of "farewell" emotion, and all other classification tasks have achieved the best performance.

![Figure 1: Accuracy and F1-score](image-url)
In order to compare the performance of the three models (BCC, BBC, GRU) on classification tasks, we used the F test to test whether the scores of the three models on each group of classification tasks are significantly different. The null hypothesis was that the scores of the three models were not significantly different, and the alternative hypothesis was that the scores of at least two models were significantly different at a significance level of 0.05. The F test can be expressed as:

\[ F = \frac{SSB/(k-1)}{SSE/(N-k)} \]  

where SSB is the sum of squares between groups, SSE is the sum of squares within groups, k is the number of groups (models), and N is the total number of observations (poems). If F is greater than the critical value from the F distribution table with degrees of freedom k - 1 and N - k, we reject the null hypothesis and conclude that there is a significant difference among the scores of the three models. We used the f_oneway function in the scipy library to calculate the F value and p value, and the results are shown in Table 3:

<table>
<thead>
<tr>
<th>Category</th>
<th>F-statistic</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>433.316</td>
<td>6.490e-12</td>
</tr>
<tr>
<td>B</td>
<td>100.871</td>
<td>3.131e-08</td>
</tr>
<tr>
<td>C</td>
<td>152.403</td>
<td>2.953e-09</td>
</tr>
<tr>
<td>D</td>
<td>24.748</td>
<td>5.521e-05</td>
</tr>
<tr>
<td>E</td>
<td>297.210</td>
<td>6.004e-11</td>
</tr>
<tr>
<td>F</td>
<td>197.442</td>
<td>6.580e-10</td>
</tr>
<tr>
<td>G</td>
<td>80.813</td>
<td>1.090e-07</td>
</tr>
<tr>
<td>H</td>
<td>77.894</td>
<td>1.338e-07</td>
</tr>
</tbody>
</table>

As the p-values for all categories were much less than 0.05, we rejected the null hypothesis and accepted the alternative hypothesis that the scores of the three models were significantly different.

To further compare the superiority of performance between models, we used a t-test to test whether there is a significant difference in the scores between models. The null hypothesis is that the scores of the two models are not significantly different, and the alternative hypothesis is that the scores of the two models are significantly different. The t-test can be expressed as:

\[ t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s_1^2/n_1 + s_2^2/n_2}} \]  

where \( \bar{x}_1 \) and \( \bar{x}_2 \) are the mean scores of the two models, \( s_1^2 \) and \( s_2^2 \) are the sample variances of the two models, and \( n_1 \) and \( n_2 \) are the sample sizes of the two models. We used the ttest_ind function from the scipy library to calculate the t and p values, and the results are shown in Table 4:
We can find that, except category B and category G, in all other categories, the p-values between all models are less than 0.05, and the t-values corresponding to BCC and BBC are positive. Therefore, the BCC model is significantly better than the BBC model and the GRU model in these categories; the BBC model is significantly better than the GRU model in score. For type B data, the BCC model is significantly better than the BBC model; the GRU model is significantly better than the BCC model and the BBC model. For G data, there is no significant difference between the BCC model and the BBC model (p<0.05), but the t value of the BCC model is positive, indicating that its results are slightly better than the BBC model; The BCC model and the BBC model are significantly better than the GRU model in score.

We also use boxplots to visualize the distribution of performance scores of the three models on different poetry categories. As can be seen from Figure 2, the BCC model has high performance scores on the vast majority of categories, while the GRU model has low performance scores on almost all categories. The BBC model has high performance scores in most categories, but is lower than BCC.
Figure 2: Boxplots on eight categories

In summary, our experimental results show that the BCC model has superior performance in most ancient poetry sentiment classification tasks.
4.2. Comparison of Deep Learning with Traditional Method

Support Vector Machines (SVM) have achieved excellent results in short text classification [10]. Therefore, we also compared the predictive performance of deep learning models with traditional regression and statistical methods such as support vector regression (SVR) and gradient boosted decision trees (GBDT), applied to the same dataset and the same train-test split is used to benchmark the performance. The F1-score of the classification results of different models on different categories is shown in the Table 5.

<table>
<thead>
<tr>
<th>Class</th>
<th>BCC</th>
<th>BBC</th>
<th>GRU</th>
<th>SVM</th>
<th>GBDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.7580±0.0122</td>
<td>0.6564±0.0188</td>
<td>0.4834±0.0128</td>
<td>0.2693±0.0105</td>
<td>0.2620±0.0074</td>
</tr>
<tr>
<td>B</td>
<td>0.6512±0.0229</td>
<td>0.5933±0.0149</td>
<td>0.7394±0.0075</td>
<td>0.7782±0.0215</td>
<td>0.7757±0.0230</td>
</tr>
<tr>
<td>C</td>
<td>0.7439±0.0470</td>
<td>0.6558±0.0441</td>
<td>0.3155±0.0298</td>
<td>0.2728±0.0221</td>
<td>0.3176±0.0184</td>
</tr>
<tr>
<td>D</td>
<td>0.5071±0.0609</td>
<td>0.4162±0.0176</td>
<td>0.3401±0.0150</td>
<td>0.4754±0.0256</td>
<td>0.4648±0.0231</td>
</tr>
<tr>
<td>E</td>
<td>0.8155±0.0352</td>
<td>0.6949±0.0074</td>
<td>0.4066±0.0305</td>
<td>0.3777±0.0157</td>
<td>0.3770±0.0289</td>
</tr>
<tr>
<td>F</td>
<td>0.7713±0.0110</td>
<td>0.6151±0.0601</td>
<td>0.2930±0.0281</td>
<td>0.7398±0.0241</td>
<td>0.7223±0.0231</td>
</tr>
<tr>
<td>G</td>
<td>0.7361±0.0702</td>
<td>0.6594±0.0364</td>
<td>0.3493±0.0392</td>
<td>0.2666±0.0309</td>
<td>0.2573±0.0335</td>
</tr>
<tr>
<td>H</td>
<td>0.6879±0.0403</td>
<td>0.6080±0.0558</td>
<td>0.3750±0.0412</td>
<td>0.3675±0.0546</td>
<td>0.3627±0.0549</td>
</tr>
</tbody>
</table>

As can be seen from the Table 5, the BCC model is still the best model except for class B, and the two machine learning methods have good performance on class B and class F.

5. Conclusion

Although BERT-CCPoem is considered to be the most advanced model for sentiment analysis tasks of ancient poems, we found that when poetry texts contain words with clear emotional tendencies, especially in the process of sentiment analysis of "farewell" poems, based on statistical SVM and GBDT for machine learning are more effective. But when the emotion of poetry is implicitly expressed through metaphors, allusions, symbols and other rhetorical means, deep learning methods such as BERT-CCPoem are more effective. In general, the BERT-CCPoem model can well adapt to this coarse-to-fine sentiment analysis method for ancient poetry, which is of great significance for the automatic annotation of Chinese classical poetry. For future work, we plan to extend our approach to other genres of ancient Chinese literature, such as ancient prose and history books. We also plan to explore other ways of incorporating lexical knowledge into deep learning models, such as using attention mechanisms or graph neural networks. We hope that our work can inspire more research on sentiment analysis and generation in Chinese literature.

References


