Personality Prediction Using RNN Leveraging the Big Five Model and Its Application in Personality-Adaptive Chatbots

Xuhan Tong *

Weiyang College, Tsinghua University, China

* Corresponding Author Email: tongxh23@mails.tsinghua.edu.cn

Abstract. This study proposes a method to predict users' personality traits, using Recurrent Neural Networks (RNNs) based on the widely used Big Five personality model, and its application to generative AI-based personality-adaptive chatbots. Text data from essays are used as dataset for the RNN model to predict these personality traits, showcasing the innovativeness of using RNN model to capture certain personality patterns from long, coherent texts instead of short posts on social media. The results show that the model can predict some personality traits effectively, but it has certain difficulties with others, which could be a potential area that may need further improvement. The study also apply the predicted personality traits into generative AI models to examine personalized chatbot responses. The findings demonstrate that after considering the predicted traits from the prediction model, generative AI models can generate more targeted responses according to the given traits. This research underscores the importance of adjusting chatbot responses based on users' personality traits, contributing to more empathetic and user-centered digital interactions.

Keywords: Personality prediction; RNNs; Big five personality; Personality-adaptive chatbot.

1. Introduction

In recent years, increasing attention has been paid on people's personality, the MBTI [1] test and the Big Five Personality [2] test has been increasingly popular among people, and people tend to take others' personality into consideration when having a conversation with others. This trend extends to interactions with chatbots, where users often seek personality-adaptive responses. As a result, designing personality adaptive chatbots has become increasingly important [3].

Several attempts have already been made in this area. In personality prediction, Ahmad et al. [4] have identified the difficulties in recognizing users' personalities and proposed deep learning methods to address them. Additionally, empathetic dialogue systems [5] were also developed in order to make computer better understand and remember users' emotions. Regarding datasets, the increasing volume of social media users has prompted researchers to turn to social media posts as a valuable source of personality data [6]. For instance, Hernandez and Scott [7] created a deep learning classifier using an MBTI dataset to predict the author's personality type using texts on tweet, leveraging RNNs. As for the model using, RNNs are widely used in personality prediction models due to their ability to retain information from previous timesteps and their effectiveness in text analysis. However, there has been insufficient focus on using long, coherent essays as input data and on applying the Big Five Personality traits in practical applications.

Inspired by this research, and recognizing the gaps in the study of the Big Five Personality in the domain of personality prediction, as well as the limited use of essays as dataset input, this research focuses on leveraging RNNs to analyze the Big Five Personality traits.

2. Methods

2.1. Overview

This research utilizes a simple Recurrent Neural Network (RNN) model to predict personality traits based on the Big Five personality framework (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism). The dataset consists of essays written by individuals exhibiting different levels of these traits. The evaluation indicators include accuracy, log loss and confusion matrix.

2.2. Principle of RNNs

2.2.1. Recurrent Structure

RNNs are distinguished by their ability to maintain previous memory through recurrent connections. This is achieved by feeding the hidden layer's output into itself again, allowing the network to consider information from former steps (see Fig. 1). This is particularly useful for tasks that involve sequences, such as language modeling or forecasting in time series data.

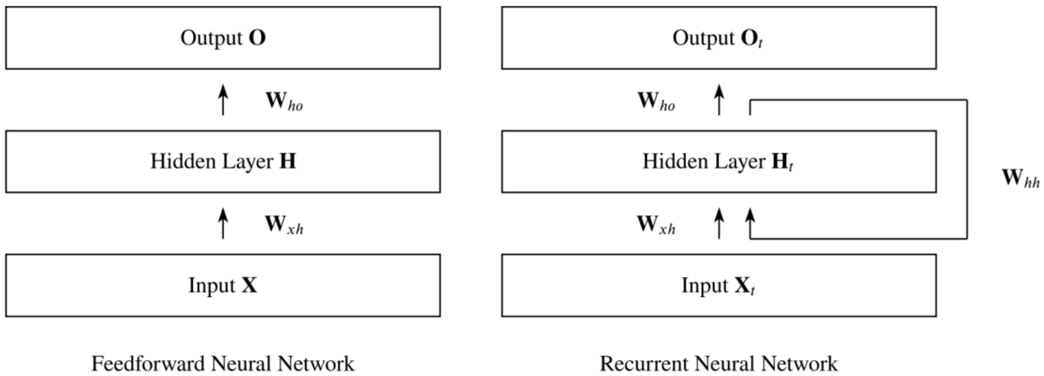

Fig 1. Visualization of differences between Feedfoward NNs and Recurrent NNs [8].

2.2.2. Time Steps and States

In an RNN, the input is processed in separate time steps. When dealing with text, each word is treated as a separate time step. The state of an RNN at each time step is determined by both the current input and the state from the previous time step. This state vector captures the context accumulated by the network up to that point.

2.2.3. Computation in the Hidden Layer

At each time step, the hidden state is updated based on the weighted current input, the weighted former hidden state, and the application of a nonlinear activation function. The equation for this update can be expressed as follow.

$$h_t = \sigma(W_{xh} \cdot x_1 + W_{hh} \cdot h_{t-1} + b_h) \quad (1)$$

Where h_t is the hidden state at the current time step; x_t is the input at the current time step; W_{xh} is the weight matrix from input to hidden layer; W_{hh} is the weight matrix from the previous hidden state to the current hidden state; b_h is the bias term; σ is the activation function.

2.2.4. Output Layer

The output of an RNN can be generated either by using the hidden state of the last time step or by aggregating information across all the time steps. The output is usually a function of the final or combined hidden states, and it can be calculated as follow.

$$y_t = \sigma(W_{hy} \cdot h_t + b_y) \quad (2)$$

Where $y_t$the output is at the current time step; W_{hy} is the weight matrix from the hidden layer to the output layer; b_y is the bias term.

2.2.5. Backpropagation and the Vanishing Gradient Problem

Training an RNN relies on backpropagation through time (BPTT), which can be achieved by applying the chain rule. However, this process can encounter problems due to the vanishing or exploding gradient issue, where gradients become too small or too large as they propagate back through many time steps, making them difficult to use effectively.

2.3. Model Assessment

To evaluate the model, the Test Loss and Accuracy of the predictions are calculated on the test dataset. Additionally, according to Chen et al. [9] and Opper et al. [10], Cross-Entropy Loss is commonly used for classification and prediction problems, designed to measure the difference between the model's predicted probability distribution and the true distribution. Thus, in order to further assess the precision of classification, Log Loss and Cross-Entropy Loss are used to evaluate the precision of the RNN model.

3. Experiment and Results

3.1. Dataset and Data preprocessing

The dataset consists of a large collection of essays (see Table 1). To make it available for processing and transmission, the dataset is split into three smaller files initially, each containing about 1,000 lines. In the main program, these files are combined as a single dataset, which is split into 60% for training and 40% for testing later. Six columns of the dataset are assigned to“texts” (the essay content) and “labels” (the five personality traits). The text data is then vectorized and standardized before being converted to tensors. Finally, the data is loaded into DataLoader objects to facilitate efficient batch processing during the process of model training and evaluation.

Table 1. Dataset Example.

Text	cEXT	cNEU	cAGR	cCON	cOPN
Well, right Now I just woke up from a mid-day nap. It's sort of weird, but ever since I moved to Texas, I have had problems concentrating on things. …… Well it's been fun writing, I don't know if you go anything out of this writing, but it helped me get some of my thoughts into order.	0	1	1	0	1

3.2. Model Construction and Training

An RNN model is built using the PyTorch module, with a certain dropout rate in order to prevent overfitting. As for optimization, the Stochastic Gradient Descent (SGD) algorithm is chosen, because it performed better in convergence speed and stability here compared to the Adam and RMSprop optimizers. The model is then trained over 100 epochs, during each epoch, the model will adjust the target labels to ensure they matched the shape of the model's previous output, to be specifically, one-hot encoded labels are converted into index labels to directly compare them with the output classes. Before feeding the inputs into the RNN, an additional dimension is added to the input tensor to match the expected input format. At the end of the epoch, the train loss for each prediction is calculated, the gradients are computed through backpropagation, and the model parameters will be updated by the optimizer.

3.3. Accuracy and Loss

Table 2. Accuracy and Loss.

Test Loss	1.0029
Test Accuracy	0.6464
Test Log Loss	1.0043
Test Cross-Entropy Loss	1.0029

The model's loss values and the test accuracy (see Table 2) suggest that while the model is minimizing errors, there is still significant room for improvement in calibration and prediction certainty. The test accuracy suggests a moderate performance level but highlights a potential need for further optimization. The consistent values of Test Loss, Test Log Loss, and Test Cross-Entropy Loss imply that the model is experiencing difficulties in learning from the data effectively, possibly due to under fitting, indicating that the model might not be capturing the underlying patterns in the data adequately. This might be attributed to factors such as insufficient model complexity, lack of proper hyper parameter tuning, or the need for more diverse or larger datasets.

3.4. Predicting Outcomes

Table 3. Predicting Outcomes.

	Labels	Amount
0	Extraversion	215
1	Neuroticism	772
2	Agreeableness	0
3	Conscientiousness	0
4	Openness	0

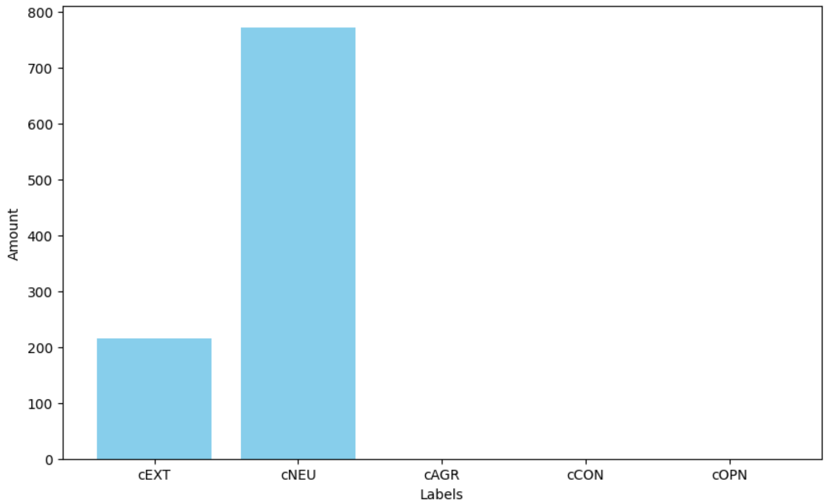

Fig 2. Distribution of Personality Predictions.

The lack of predictions for Agreeableness, Conscientiousness, and Openness (see Table 3 and Fig. 2) is a significant limitation, indicating that the model struggles to identify these traits effectively, which could affect the overall utility of the model in applications requiring a balanced assessment of all personality dimensions.

3.5. Responce Generation (based on generative model Chat-GPT 4)

The user input example: "Tom failed at the exam, please give him some comfort."

Features of the Response Generated Before Using the Prediction Model:

1. Expresses an apology in a straightforward manner.
2. Uses others as examples to encourage users.
3. Assures users of their own abilities.

Features of the Response Generated After Using the Prediction Model (Extraversion, Agreeableness, and Openness):

1. Shows understanding toward the user.
2. Suggests that the user surround themselves with friends.
3. Reassures users about their own abilities.

Clearly, after implementing the prediction model, the AI-generated responses have become more personalized and targeted to the user's personality traits. In this example, the user is identified as Extraverted, Agreeable, and Open, which leads to suggestions like encouraging them to stay with friends (point 2), aligning more closely with their typical relaxation methods.

4. Conclusion

This result highlights the advantages of leveraging advanced neural networks for nuanced and personalized predictions, providing a valuable foundation for future research and applications in personalized AI and human-computer interaction. This approach enhances the accuracy and relevance of personality assessments, paving the way for more targeted and empathetic digital interactions.

Future work could include the following parts. The first one is to utilize more advanced architectures, for example, Long Short-Term Memory networks or Gated Recurrent Units. These architectures are specifically designed to mitigate the problems of the vanishing and exploding gradient. Additionally, incorporating techniques like attention mechanisms could help the model focus on the input sequence, which could improve its ability to capture long-term dependencies. Last but not least, optimizing the training process by using more sophisticated optimization algorithms is another area for future exploration.

References

[1] Park S., Lee S. Y., Kim S. H. Personality types based on the Myers-Briggs Type Indicator and User Experience Design. Journal of Usability Studies, 2020, 15(3): 123-138.

[2] John O. P., Naumann L. P., Soto C. J. Paradigm Shift to the Big Five Trait Taxonomy: History, Measurement, and Conceptual Issues. Handbook of Personality: Theory and Research, 4th edition, 2021.

[3] Nass C., Moon Y., Fogg B. J., Reeves B., Dryer D. C. Can computer personalities be human personalities? Int J Hum Comput Stud. 1995, 43(2): 223–239.

[4] Ahmad H., Asghar M. Z., Khan A. S., Habib A. A systematic literature review of personality trait classifcation from textual content. Open Comput Sci. 2020, 10(1): 175–193.

[5] Ma Y., Nguyen K. L., Xing F. Z., Cambria E. A survey on empathetic dialogue systems. Inf Fusion. 2020, 64: 50–70.

[6] Kaushal V., Patwardhan M. Emerging trends in personality identifcation using online social networks—a literature survey. ACM, Trans Knowl Discov Data. 2018, 12(2): 1–30.

[7] Ait Baha, T., El Hajji, M., Es-Saady, Y. et al. The Power of Personalization: A Systematic Review of Personality-Adaptive Chatbots. SN Computer Science, 2023, 4, 661: 1-25.

[8] Schmidt R. M. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. arXiv: 1912.05911.

[9] Chen Qian, Shuang Li, Jianmin Wang. MM-CPred: A Multi-task Predictive Model for Continuous-Time Event Sequences with Mixture Learning Losses, Lecture Notes in Computer Science, 2021, 1268: 509–525.

[10] Opper M., Haussler D. Worst Case Prediction over Sequences under Log Loss, the Mathematics of Information Coding, Extraction and Distribution, 1999, 107: 81-90.