Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/33/20230235

A review-based approach to user profiling

Yudi Wu

School of Intelligent Science and Engineering, Harbin Engineering University, Harbin City, Heilongjiang Province,China.

2783906941@qq.com

Abstract. With the popularity of social media, sentiment analysis and text categorisation by analysing the information people post online has become an effective method to study personality prediction. This paper focuses on how to use a personality prediction model based on Bidirectional LSTM for personality prediction. Accurate personality prediction results can provide personalised recommendation services for individuals, which has certain commercial value. In this paper, the dataset of Kaggle is first preprocessed, and then the Bidirectional LSTM model is constructed and the hyperparameters are set.The processed data are then put into the model for training and testing. Finally, the above steps are repeated using other different machine learning models. After comparison experiments with other common machine learning models, it was found that the Bidirectional LSTM model showed significant advantages in the personality prediction task, and its accuracy reached 93.5%, which was significantly higher than the traditional machine learning model.

Keywords: Natural Language Processing, User Profiling, Bidirectional Long Short-Term Memory.

1. Introduction

The popularity of the Internet has led to a large number of remarks. Through natural language processing and emotional analysis, information with business and social value can be obtained from these remarks. Emotional analysis can also be used to screen depression and suicide risks [1]. In addition, by analyzing the activities and language use models on social networks, users can predict the personality characteristics and behavior style of users. Personality prediction helps to understand human behavior and promote the development of personalized service technology. Such technologies can reduce subjective prejudice and accurately reflect the individual's true personality characteristics.

Some related studies have proposed methods to perform personality prediction. Among them, To determine the distance between samples, detect outliers, and apply the maximum interval criteria and support vector machine method for personality prediction, Zhong et al. recommended using Mahalanobis distance and Z-value[2]; By building an end-to-end video analytics network based on 3D-ConvNet, Xu et al. improved the accuracy of personality prediction by addressing the issue of network overfitting[3]; Wang et al. used Attention Recurrent Neural Networks (AttRNNs) to solve the problem of neglecting the temporal features of digital footprints by commonly used classification models, and proposed an effective method for predicting personality features [4]. Si et al. introduced a Bert-LSTM model to address the challenge of limited text utilization in textual sentiment analysis tasks. Their proposed model not only outperformed other existing methods but also demonstrated superior accuracy, precision, recall, and F-Measure metrics. This approach effectively mitigates the limitations associated with text analysis and showcases improved performance across multiple evaluation criteria [5]. There are some other approaches that have also obtained some results, such as the JUHA model proposed by Liu et al, the 2CLSTM model proposed by Sun et al, and the model proposed by Zhu to analyse user behaviours on social media platforms using artificial neural networks and stochastic wandering [6][7][8].

There are also studies that propose ways to improve personality prediction methods. For example, Bharadwaj et al. proposed a feature selection method using feature vector construction such as TF/IDF, word frequency/inverse document frequency and SVM classifier for training and testing [9].Gjurković et al. used the MBTI dataset and Reddit social media to obtain features for personality prediction [10].Yu and Markov suggested a strategy for mean pooling layer alignment and a CNN model for personality prediction [11].Li et al. introduced a CNN-based multi-task learning framework for simultaneous detection of personality features and emotions [12].Tandera et al. used MLP deep learning architecture for personality prediction and achieved good accuracy [13]. These methods are helpful in improving personality prediction.

Bidirectional LSTM models have strong modelling capabilities in personality prediction, being able to handle sequential data, capturing long range dependencies, considering both past and future contextual information, and being able to handle variable length sequential data, as well as the advantages of extracting rich features and providing interpretability. These features make Bidirectional LSTM models perform well in personality prediction tasks. In this paper, the author use the mbti dataset on kaggle to make predictions using the Bidirectional LSTM model and compare it with several other common machine learning models, and finally the Bidirectional LSTM model gets 93.5% accuracy, which is significantly better than other machine learning models, indicating that the Bidirectional LSTM model is more suitable for the task of sentiment classification of text.

2. Principle of the method

LSTM is a model used to solve the long-term dependency problem in RNN. It improves the performance of the model by introducing a gating mechanism to remember and utilise distant information [14].

Traditional LSTM models can only read the input sequence sequentially from front to back step by step when processing sequence data, which may result in failing to capture some key contextual information in the sequence. In order to consider the contextual information more comprehensively, Bidirectional LSTM introduces the inverse LSTM to process both the forward and backward directions of the sequence.

2.1. The formula and structure of Bidirectional LSTM

The input layer, the forward LSTM, the reverse LSTM, the hidden state connection, and the output layer make up the Bidirectional LSTM. The sequence $x = [x_1, x_2, \ldots, x_t]$ is entered via the input layer. The forward LSTM processes the input sequence x step by step along the front-to-back direction. During each time step, the hidden state from the previous time step and input x_t from the current time step are provided to the forward LSTM. The forward LSTM then proceeds to update the cell state using a set of gating mechanisms and activation functions. Afterward, both the cell state and the hidden state are updated accordingly. The specific computational procedure of the forward LSTM is as follows:

Table 1. The specific computational procedure of the forward LSTM

Basic Unit	Equation
Input gate	$i_t = \sigma(W_i * [h_{forword_{t-1}}, x_t]+b_i)$
Forget gate	$f_t = \sigma(W_f * [h_{forword_{t-1}}, x_t]+b_f)$
output gate	$o_t = \sigma(W_o * [h_{forword_{t-1}}, x_t]+b_o)$
Cell state	$c_t = f_t * c_{forword_{t-1}} + i_t * \tan h\,(W_c * [h_{forword_{t-1}}, x_t]+b_c)$
Hidden state	$h_{forword_t} = o_t * \tan h(c_t)$

The backward LSTM processes the input sequence x step by step in a back-to-front order.The specific computation procedure of the backward LSTM is similar to that of the forward LSTM and will not be repeated here.The hidden state connection process is to connect the hidden states of the forward($h_{forword_t}$) and backward LSTM ($h_{backword_t}$) within each time step.This connection is established to obtain the final hidden state h_t.

$$h_t = [\, h_{forword_t}; h_{backword_t}].$$

According to the needs of the task, different activation functions and loss functions can be used in the output layer of the Bidirectional LSTM for classification, regression, or other specific tasks.

In summary, Bidirectional LSTM captures contextual information in sequence data by running forward and backward LSTM networks. Forward LSTMs process input sequences from front to back and backward LSTMs process input sequences from back to front. By employing a series of gating mechanisms and activation functions, the cell state and hidden state are updated accordingly. Eventually, The Bidirectional LSTM combines the hidden states from both the forward and backward LSTMs, enabling the acquisition of more comprehensive contextual information. In the inference phase, the forward and backward parts need to run simultaneously to generate Bidirectional hidden states.

2.2. *Network Structure of Bidirectional LSTM*

The following is a simple schematic of the Bidirectional LSTM, as shown in the Figure 1.The Bidirectional convolutional neural network utilizes two hidden layers: A for the forward computation and A' for the backward computation. The result y is determined by both A and A'. Specifically, during the forward calculation, the hidden layer's current state s_t is influenced by its previous state s_{t-1}. On the other hand, during the backward calculation, the hidden layer's current state s_t is influenced by its subsequent state s_{t+1}. This approach enables the network to capture and analyze the input sequence from both directions, enhancing its ability to identify overlapping and repeated content.

The relevant formulas have been provided below:

$$o_t = g(V_{s_t} + V's'_t)$$

$$s_t = f(Ux_t + Ws_{t-1})$$

$$s'_t = f(U'x_t + W's'_{t+1})$$

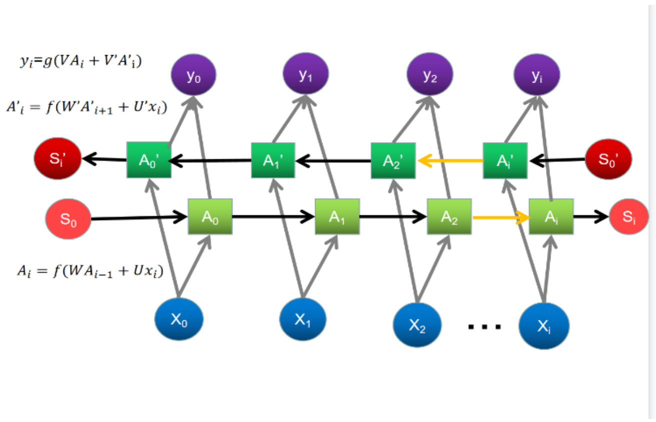

Figure 1. Network structure diagram of Bidirectional LSTM

3. Experiments

3.1. *data set*

This experiment used MBTI personality types as categorical indicators [15]. This is a popular personality test used to examine the association between personality type and writing style. The dataset comes from the PersonalityCafe forums and contains the MBTI types of over 8,600 people and the last 50 messages. Each person's message is represented by one line of data. The specific form of this dataset is as follows:

$$Type|||Post1|||Post2\ |||...|||Post50$$

Each row represents one person's information, including their MBTI type and the last 50 messages they posted in the past. Where "Type" is the person's MBTI type, and "Post1" to "Post50" are the person's past postings, which are separated by "|||".

The author conducted a simple analysis of the above data and drew a few charts to facilitate the next more in-depth analysis:

The Table 2 shows some of the data and it can be seen that the dataset consists of a column for personality type and a column for job title, which totals 8,674 samples.

Table 2. Partial presentation of the dataset

	type	posts
5729	ENTJ	'I'm questioning my extroversion...
5730	INFJ	'Always striving for more interesting possibilities ...
5731	INFP	'That's how Keirsey figures it in his books ...
5732	INTJ	'Yeah, exactly! Either that or reading books...
5733	ENFP	'7w6 ENFP here...

By conducting a count of each personality type in the dataset, a graph was generated to visualize the distribution across the different classes as Figure 2. The analysis revealed that the dataset exhibits an evident class imbalance, with certain personality types having a greater number of instances compared to others. Notably, the most prevalent personality type within the dataset was found to be INFP. The visual representation vividly illustrates the varying frequencies of each personality type, providing valuable insights into the distribution patterns within the dataset.

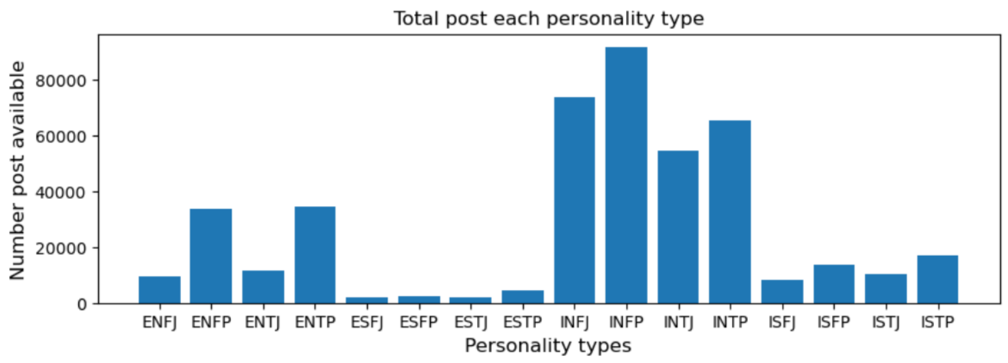

Figure 2. Total post each personality type

The Figure 3 shows the histogram distribution of the data in a single column, and as can be seen from the graph, most of the posts are between 7,000 and 9,000 words. The graph includes a line that represents the kernel density estimate, which is employed to address a fundamental challenge in data analysis: making inferences about the overall population based on a limited sample of data. The kernel density estimate is a function that is determined by summing the contributions of the kernel functions at each individual data point. With the use of this smoothing approach, it is possible to get insights into the data's underlying distribution and gain a deeper knowledge of the traits of the population. By visualizing the kernel density estimate line, valuable insights can be gained into the overall data pattern and informed inferences can be made about the broader population.

To expand the MBTI dataset and capture individual personality indices, the author adopts a Binary encoding approach by adding columns for the personality type indicator. This involves assigning a value of 1 to the corresponding axis (IE, NS, TF, JP) if a person exhibits the specific trait (I, N, T, J), while a value of 0 is assigned if the trait is absent. This encoding enables the computation of various metrics within the dataset, such as determining the number of introverted or extroverted posts present, by summing the corresponding values across all entries. By extending the dataset in this manner, a more comprehensive analysis of individual personality traits becomes possible, allowing for deeper insights and potential correlations in the data. The Table 3 shows the expanded dataset.

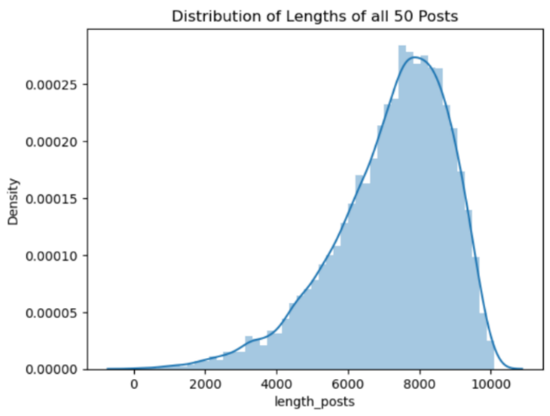

Figure 3. Distribution of lengths of all 50 posts

Table 3. Expanded MBTI dataset

	type	posts	IE	NS	TF	JP
5729	ENTJ	'I'm questioning my extroversion...	0	1	1	1
5730	INFJ	'Always striving for more interesting possibilities ...	1	1	0	1
5731	INFP	'That's how Keirsey figures it in his books ...	1	1	0	0
5732	INTJ	'Yeah, exactly! Either that or reading books...	1	1	1	1
5733	ENFP	'7w6 ENFP here...	0	1	0	0

3.2. Pre-processing methods

Before building the model, the first step is to preprocess the data, including performing unique heat encoding of tags, creating a dictionary mapping of MBTI types, data cleaning of posts, creating lookup tables, populating and truncating posts, dividing the dataset, etc., and the following are the specifics:

Firstly, use the LabelBinarizer class to uniquely encode MBTI types into numeric representations. Create a dictionary that maps each MBTI type to its corresponding numeric representation. And then perform data cleansing on posts, such as converting to lowercase, removing special characters, and other operations. Next,create a lookup table that maps words to integer indexes. Replace words in posts with integer representations and generate a list of posts with integer representations. Fill or truncate the posts to have the same length as needed. Finally, divide the dataset into training and validation sets for model training and evaluation.

3.3. Training and Testing

In the process of model construction and training, the hyperparameters of LSTM were first set. The experiment sets the hidden state size of each LSTM unit to 256. 1 layer of LSTM is used. batch_size is set to 256 here, meaning that each training batch will contain 256 samples. Here the learning rate is set to 0.01 which is a common initial learning rate. Here the word embedding dimension is set to 250. the number of words in the vocabulary is calculated by adding 1 to the size of the vocabulary to denote additional unknown words.

A graph object is then created and the various nodes of the model are defined in it, such as input placeholders, label placeholders, embedding matrices, forward and backward LSTM units, and fully connected layers. Input placeholders (input_data) and label placeholders (labels_) are used to store input and label data. Embedding matrix (embedding) is used to convert the input sequence into a series of word embedding vectors. Forward and backward LSTM units (LSTM_fw, LSTM_bw) are used to build the basic units of the LSTM layer. LSTM units with dropout (drop_fw, drop_bw) are used to prevent overfitting. Forward and backward multilayer LSTM cells (cell_fw, cell_bw) are used to construct multilayer LSTM models. Initial state (initial_state_fw, initial_state_bw) is used for the initial state of the LSTM cell. Outputs and final states (outputs, final_state) are computed by calling tf.nn.bidirectional_dynamic_rnn().The outputs of the LSTM are classified using the fully connected layer (pre, predictions).

The construction and compilation of the Bidirectional LSTM model have been completed. The following demonstrates the training and testing of the model.

To carry out the training process, a session is first created using tf. Session and all the variables of the model are initialised. epoch is set to 3 and in each epoch, initial_state_fw and initial_state_bw are run using sess.run to initialise the state of the forward and backward units. Iterate through each data batch using the enumerate function and generate training data using the get_batches function. Print the training losses every 5 iterations. Evaluate the accuracy of the validation set every 25 iterations. Run the final_state node using sess.run to get the final state and update it to val_state_fw and val_state_bw. Then use the get_batches function to generate the validation set data, populate the data with placeholders, and run the accuracy node to calculate the accuracy.

This experiment also discusses hyperparameter settings for several common machine learning models.The hyperparameters test_size=0.33 and random_state=7 were consistently set during training and testing of the models. The other hyperparameters for each model are as follows:

For eXtreme Gradient Boosting, the reg_lambda is set to 1, with 100 estimators, a subsample of 0.8, max_depth of 5, gamma of 0, learning_rate of 0.1, colsample_bytree of 0.8, and reg_alpha of 0. Logistic Regression uses L2 penalty with a C value of 1, lbfgs solver, and a maximum of 100 iterations. Random Forest employs 100 estimators, 1 minimum sample leaf, max_depth of 5, random_state of 7, and a minimum split sample of 2. Stochastic Gradient Descent includes a learning_rate of 0.1, alpha of 0.0001, maximum iterations of 1000, and an automatic multi_class setting. SVM has a C value of 1, rbf kernel, gamma set to scale, degree of 3, and shrinking enabled. Lastly, KNN has 2 neighbors, uniform weights, automatic algorithm selection, a leaf size of 2, and minkowski metric.

These hyperparameter settings are chosen based on the characteristics of the model and experience. By setting sensible hyperparameters, the performance and complexity of the model can be adjusted to obtain better predictions. Following training and testing using these predetermined hyperparameters, the models are compared for performance based on their accuracy.

4. Experimental results

After each model is trained, it is tested using the test set and finally the test set correctness of each model is printed out and the following results are obtained:

Table 5. Accuracy of each model

Model	Accuracy(%)
Bidirectional LSTM	93.5
Random Forest	73.5
XGBoost	73.2
Stocastic Gradient Descent	75.1
KNN	64.4
SVM	75.6

From the table, it is evident that the Bidirectional LSTM model achieves the highest accuracy rate. This rate is significantly superior to that of the other models, indicating its strong performance in the prediction task of MBTI personality types. Therefore, the Bidirectional LSTM model proves to be a more desirable choice in this regard.

5. Conclusion

It was found that Bidirectional LSTM models perform better in the MBTI personality type classification task. Bidirectional LSTM is better at handling sequence diversity in natural language processing tasks by capturing contextual information and alleviating the long-term dependency problem. Compared to traditional unidirectional models, Bidirectional LSTM can use more information for decision making and prediction, adapt to complex tasks, and have higher accuracy and generalisation. The next step could be to optimise the Bidirectional LSTM model, such as increasing depth, introducing an attention mechanism, pre-training word vectors, and fusion with other models, to further improve the model performance.

References

[1] Y. Hosgoren, H. Oztoprak, J. Hasanli, B. Özel, A. Bagcaz,Prediction of suicidal behavior by evaluating personality traits with machine learning techniques,European Neuropsychopharmacology, vol. 44, no. Supplement 1, 2021, pp. S34-S35.

[2] Zhong, X., Guo, S., Gao, L., Shan, H., & Xue, D 2018 A General Personality Prediction Framework Based on Facebook Profiles. In Association for Computing Machinery ,pp. 269-275.

[3] CONTRACTOR, D., PATRA, B., MAUSAM, et al. (2021). Constrained BERT BiLSTM CRF for understanding multi-sentence entity-seeking questions. In Proceedings of the Conference on Natural Language Processing,pp. 65-87.

[4] Wang, L., Cui, L., Liu, X., Lu, Y., & Li, Q. (2020). "Personality Traits Prediction Based on Users' Digital Footprints in Social Networks via Attention RNN." 2020 IEEE International Conference on Services Computing (SCC), Beijing, China. pp. 54-56.

[5] Dhanalakshmi, R., Sudalaimuthu, T., & Radhakrishnan, K. R. (2022). Early Detection of Sepsis Using LSTM and Reinforcement Learning. In Applications of Computational Methods in Manufacturing and Product Design: Select Proceedings of IPDIMS 2020 pp. 297-306.

[6] Yi, Y., Bian, Y. (2021). Named Entity Recognition with Gating Mechanism and Parallel BiLSTM. In Proceedings of the 2021 Conference,pp. 1219-1237.

[7] Joshi, V. M., Ghongade, R. B., Joshi, A. M., et al. (2022). Deep BiLSTM neural network model for emotion detection using cross-dataset approach. In Proceedings of the 2022 Conference ,pp. 103407.1-103407.10.

[8] Xiong, R., Wu, C., Li, X., et al. (2019). Character-level Based Conference Named Entity Recognition Using BiLSTM. In International Conference on Computer, Network, Communication and Information Systems: CNCI 2019, Qingdao, China, 27-29 March 2019,pp. 83-88.

[9] S. Bharadwaj, S. Sridhar, R. Choudhary, R. Srinath,Persona traits identification based on myers-briggs type indicator (MBTI)-a text classification approach,Proceedings of the2018 international conference on advances in computing, communications and informatics(ICACCI, IEEE (2018), pp. 1076-1082.

[10] M. Gjurković, J. Šnajder,Reddit: a gold mine for personality prediction,Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media (2018), pp. 87-97.

[11] J. Yu, K. Markov,Deep learning based personality recognition from facebook status updates,IEEE 8th International Conference on Awareness Science and Technology (iCAST).IEEE (2017), pp. 383-387.

[12] Li, H., Luo, S., Wang, H., et al. (2020). Target Classification Based on Radar Track Using BiLSTM. In IET International Radar Conference: IET IRC 2020, Online, 4-6 November 2020, Volume 3 of 3,pp. 1555-1559.

[13] Tandera T, Hendro Suhartono D, Wongso R, Prasetio YL. Personality prediction system from facebook users. In: Procedia computer science, vol. 116; 2017. p. 604–11.

[14] Young-Jin Nam and Ha-Hyun Jo, "Prediction of Weekly Load using Stacked Bidirectional LSTM and Stacked Unidirectional LSTM," The Journal of Korean Institute of Information Technology, vol.18, no.9, pp.9-17.

[15] Eicher, O., Farmer, D., Li, Y., et al. (2021). Handwritten Chess Scoresheet Recognition Using a Convolutional BiLSTM Network. In Document Analysis and Recognition - ICDAR 2021 Workshops: Part I, pp. 245-259.