MBTI Personality Prediction Based on BERT Classification

Hanwen Zhang *

Georgetown Preparatory School, North Bethesda, United States

* Corresponding Author Email: hzhang@gprep.org

Abstract. Young people today tend to express their feelings and socialize on the internet instead of in real life, which makes social media practical in defining one's personality since their expressions usually exhibit their personalities. Predicting people's personalities based on their posts is a relatively challenging task requiring large quantities of processing data and modeling. This paper uses two word-embedding methods, BERT classification and TF-IDF Vectorizer, and three models, including Logistic Regression, K-Nearest Neighbors, and Random Forest Classifier, to find this task's state of the art method. In this case, with BERT classification, the state-of-the-art method for most of the Natural Language Processing(NLP) tasks, Logistic Regression is the best-performing model with an average accuracy of 87 percent.

Keywords: BERT Classification, Logistic Regression, TF-IDF Matrix, NLP, MBTI.

1. Introduction

With psychology's development, more research has gradually appeared and divided personalities into different genres. One of the most influential and supported personality dividers is Myers–Briggs Type Indicator (MBTI) [1]. Katharine Cook Briggs and her daughter Isabel Briggs Myers first constructed the original version of MBTI. With more perfection, the most delinquent version divides the personalities using four aspects--mind, energy, nature, and tactics--and one identifier [1, 2]. Each element has two categories that contradict each other. It constitutes 16 distinct personalities in its entirety. With the prevailing psychological examination, young people, to an increasing extent, are interested in figuring out their personality types. Instead of doing long and tedious questionnaires, people prefer an easier way to know their personalities. Social media has been a prominent venue for people's sharing. People express their feelings, thoughts towards events, and preferences; they also communicate with each other on social media. According to researchers, people's words reflect what kind of people they are [3]. Therefore, their claims on social media provide evidence of their personalities.

Predicting personalities based on users' tweets is a task of text classification. In daily life, people use some words frequently, such as "like," "think," "people," etc. A vector mainly filled with vectors representing these repeating words does not fit this task. Term frequency-Inverse document frequency [4] was once the state-of-the-art method. Even though its performance remains terrific, BERT classification [5] overcomes it by six percent of accuracy. It is still a relatively rugged vectorizer for text classification, so this paper compares it with BERT classification, the state-of-the-art method for many NLP tasks that recently emerged for vectorizing.

Since this task belongs to classification, restricted models remain available. This paper applies three distinct models, including Logistic Regression [6], K-Nearest Neighbors [7], and Random Forest Classifier [8], to find the state-of-the-art method for this task. K-Nearest Neighbors [7] is a supervised model that assigns new data to the closest indicated cluster. It fits this task because the task requires classifications. However, the common usage of some words may have resulted in a vague line between clusters in a binary classification, which lowers the model's accuracy. Random Forest is also a powerful method for a classification task. However, it takes around ten times longer than Logistic Regression and gets a lower accuracy. Logistic regression turns out to be the best-performing model. Dividing the classifications of sixteen personalities into four binary classifications, this model takes the least time and has the highest accuracy.

2. Related Work

2.1. Toxic Text Classification

As the SOTA method, BERT classification is used in multiple NLP tasks. A German paper [9] applies it to identifying German toxic, engaging, and fact-claiming comments. The authors obtained the dataset from a Facebook page of a political talk show of a German television broadcaster from February to July 2019. The dataset contains 3244 comments. This paper divides the comments into three types, toxic, engaging, and fact-claiming. Some comments have multiple labels, and some have no labels.

They used GBERT [10], a BERT model pre-trained for German, as one of their models. Then they add a classification head on top of the first output vector of both pre-trained models. In the GBERT architecture, the model generates the output vector by inserting a classification token at the start of each input sentence. This vector is used for the next sentence prediction task during the pre-training. The classification head comprises a linear layer with the same hidden size as the transformer model, tracked by a tanh activation function and another linear layer. The first linear layer is initialized with the weights learned during the pre-training task. The other layers are initialized randomly. For a single-label classifier, the final linear layer comprises two outputs followed by a softmax function. For a multi-label classifier, the final linear layer comprises three outputs followed by a sigmoid function. This model has succeeded with an F1 score of 0.726.

2.2. Word2vec for NLP Tasks

Google has been a pioneer in Machine Learning algorithms in NLP tasks. Despite the fantastic BERT classification, they also created other Machine Learning algorithms that once were the SOTA methods. Word2vec [11] is an excellent instance of those algorithms. People were using different neural networks for NLP tasks. Using both the syntactic and semantic similarities between the words, word2vec made considerable improvements. Two models train word2vec.

One is the Continuous Bag Of Words model(CBOW). CBOW is a form of the Neural Network Language Model, one past method people use for NLP tasks, without a linear hidden layer. This model will predict the middle word of a sentence with already N words while the first N/2 history words and N/2 future words are provided. The best result of the model appears when N = 8, which means that this model does not fit NLP tasks with long sentences too well. The N vectorized words will be averaged when the input experiences the projection layer. Then, the averaged vector will go on to the output layer, followed by hierarchical softmax. The other model is the Continuous Skip-gram Model, which serves as a supplement to the first model. It predicts N words for context before and after the middle word. More words closely related to the input word will be sampled more for generating output labels. When there is a number N, a random number R is generated from 1 to N. And the R words for history and future will act as correct labels for the model.

3. Methodology

3.1. Data

The data is found on Kaggle [12]. The dataset includes 8675 samples with MBTI personalities and the corresponding users' tweets. The creator of the dataset collected the last 50 tweets the users had sent and separated them with blocks. The tweets include anything, including normal posts, replies with usernames, and shares of links.

3.2. Data Pre-Processing

The target value is set up in the way of binary classification. A table of target values shown below is generated with the help of pandas by setting up binary values for each of the four traits of MBTI

personalities [2]. In the trait, Mind, the target value is 0 if it is “I”, 1 if it is “E”. The same strategy for target values is applied to the other three traits of MBTI personality.

A graph made using Matplotlib, Figure 1, is shown below for visualization of the number of samples for each personality. The tweets include words, URLs, and usernames. Each tweet is separated by "|||." Thus, it is necessary to pre-process the data by cleaning the text. Using the method, sub() in the re module, it replaces the URLs and usernames with strings "URL" and "username." It also replaces the blocks with space. Using wordcloud module, Figure 2 below shows bigger font size if the words have higher frequencies. Because of the different needs for different vectorizing methods, this paper applied two ways of processing.

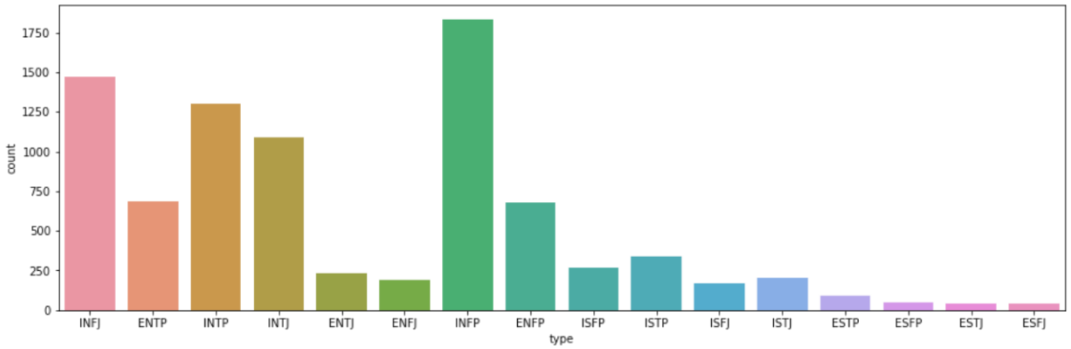

Figure 1. The Number of Samples in the Dataset

Figure 2. Top Word Choices in the Samples

3.3. Vectorizing

3.3.1 TF-IDF Vectorizer

For TF-IDF Vectorizer [4], since it generates an array based on the term frequency and inverse document frequency, it does not need the context of a word. Thus, this paper removes the punctuations and the stopwords. Stopwords, such as "the, "a," and "in," are words that do not have a specific meaning [4, 13]. Then the word_tokenize method from the nltk module is used to tokenize the words. Simplifying the text, WordNetLemamatizer from the nltk module returns the words to their bases or roots, helping better count the frequencies. Determining the term frequency and inverse document frequency, the TF-IDF vectorizer [4] takes in a set of texts and returns an array of weights representing the text's importance, which will be the direct input for the classifiers. TF-IDF is the abbreviated form of *Term Frequency-Inverse* Document *Frequency*. *Term frequency* calculates the frequency within which a specific word appears in the document, whereas *Inverse Document Frequency* finds the regularity of a word in the corpus. In the scikit-learn module, the TF-IDF vectorizer solves a problem where words do not appear in the corpus and cause a divided-by-zero error by adding 1 to the numerator and the denominator of the equation. The scikit-learn equation is shown as below[4]:

$$idf(t,d) = log\left(\frac{1+n}{1+df(t,d)}\right) + 1 \quad (1)$$

Where is t the term to measure its commonness, and n is the number of documents d; and $df(t,d)$ is the document frequency of the term.

By multiplying the two matrixes representing Term Frequency and Inverse Document Frequency, the TF-IDF Vectorizer generates an array representing the weights of the texts' importance as presented below:

$$tfidf(t,d) = tf(t,d) \cdot idf(t,d) \quad (2)$$

Where $tf(t,d)$ is the term frequency, and the times the term t appears in the document d. $idf(t,d)$is the inverse document frequency.

3.3.2 BERT Classification

BERT Classification [5], a Transformer-architecture model, is the abbreviation of Bidirectional Encoder Representations from Transformers. It has been the state-of-the-art model for NLP tasks since it came out. This study fine-tunes BERT to make the state-of-the-art model for this specific task.

As Figure 3 shown above, BERT Classification [5] takes in complete sentences and generates an array based on the left and right contexts. Thus, keeping the punctuation is necessary. In addition, URLs become useless and are deleted. Then, a function that applies the sent_tokenize method is developed to separate the sentences. The sequences of the pre-processing result provide the input for BERT. The texts enter the encoders of BERT and become vectors. Then the vectors enter the decoder layers and come out as a feature vector with a size of n x 768 [5], which are ready for training in the four classifiers for MBTI personalities, including mind, nature, energy, and tactics [2].

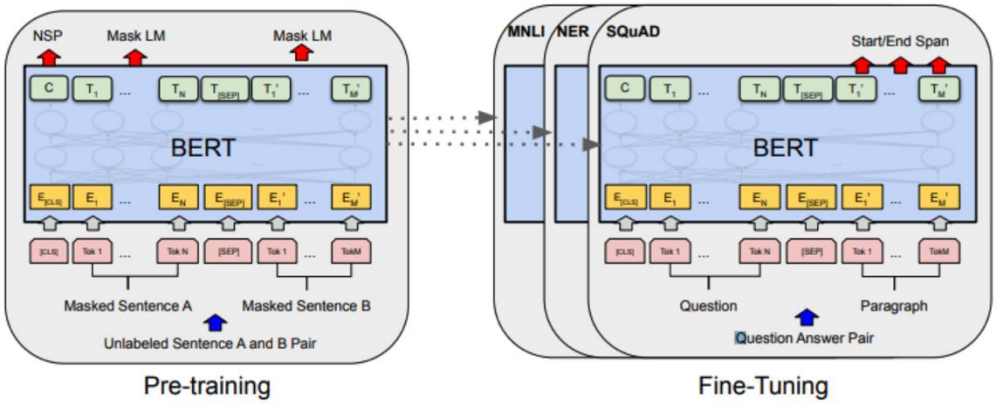

Figure 3. Procedures of BERT Classification in NLP Task [5]

The loss function of BERT Classification is Cross Entropy [14], which quatifies the difference between between two probability distributions. The equation is listed below [2, 14]:

$$L_C = y\log(\hat{y}) + (1-y)\log(1-\hat{y}) \quad (3)$$

Where the loss function L_C is for each type $C \in M, N, E, T$; y stands for the binary value in each trait; and $\hat{y}$ stands for the prediction of the binary value.

$$L = \sum_{C\in(M,N,E,T)} L_C \quad (4)$$

Where L is the sum of the loss in all four traits, which is the total loss.

3.4. Models

3.4.1 Logistic Regression

Unlike Linear regression, Logistic Regression [6] fits this task the most because it is a binary classification method for categorical variables, where this task can diverge into four binary classification tasks [2]. This method applies a curve fit that fits the data into a sigmoid function, as

the equation shows below, with the y-value, probability, between 0 and 1. The line that divides the dependent variable results is also customizable by defining values.

$$S(x) = \frac{1}{1+e^{-x}} \tag{5}$$

3.4.2 K-Nearest Neighbor

K-Nearest Neighbor(KNN) algorithm [7] is a supervised machine learning algorithm; the data is clustered based on its type. While predicting the test data, the algorithm selects a number, K, used as the number of the nearest data points. Then it calculates the Euclidean distances between the new data points in the test data and the train data. The algorithm ranks the distances in ascending order and selects the first K number of data points. Finally, it assigns a cluster, the most frequent cluster among the K number of data, to the testing data. This paper applies KNN algorithm to four binary classification tasks.

3.4.3 Random Forest Classifier

A Random Forest classifier [8] is made of multiple decision trees. These decision trees will narrow the variety of outcomes by having multiple steps to determine more situations. The entire random forest classifier includes many decision trees. Unlike having one answer from one perspective, the random forest classifier takes n-number outputs of decision trees to decide the majority, which will be the final output. However, since there are many decision trees, it takes more steps to run the entire model and more time to get the answer.

4. Result

Table 1 shows the result of the experiment. With both vectorizers, Logistic Regression has the highest accuracy among all three models. And the accuracy of Logistic Regression with fine-tuned BERT classification is higher than the one with TF-IDF vectorizer.

Table 1. Overall Modeling Average Accuracy

Model Accuracy	TF-IDF Vectorizer	Fine-tuned BERT Classification
Logistic Regression	0.843	0.877
K-Neareset Neighbor	0.741	0.805
Random Forest	0.788	0.838

With the vectors generated by both TF-IDF vectorizer and fine-tuned BERT classification, the accuracy of Logistic Regression is higher than Random Forest Classifier that has higher accuracy than K-Nearest Neighbor. As mentioned above, because of this paper's binary classification mindset and imbalanced data, K-Nearest Neighbor and Random Forest Classifier cannot claim a more apparent line in each binary classification. As a result, the accuracy is lower. Logistic Regression is the best fit for this task. In this paper, the classification of 16 personalities is divided into four binary classifications, which tremendously contributes to the model's success.

Table 2. Specific Accuracy for Logistic Regression Model with Fine-tuned BERT Classification

Trait	Accuracy
Mind	0.89371
Energy	0.90256
Nature	0.90232
Tactics	0.81211

Table 2 shows the specific accuracies for each trait of MBTI personalities of the SOTA method. The accuracy of Tactics has a massive gap between any other traits, which is due to the imbalance inside the dataset. As Figure 1 shows, the samples are not evenly distributed.

Some personalities have over a thousand samples, whereas some have no more than two hundred. In the case of Tactics, as Figure 4 shown below, there are 3,434 samples of Js but 5,241 samples of Ps. The vast difference in the sample quantities results in the trait's low accuracy.

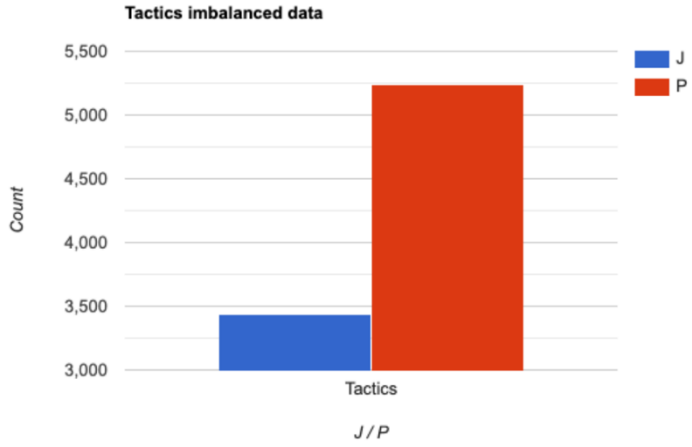

Figure 4. Imbalanced Data of Tactics Trait

5. Conclusion

Myers-Briggs Type Indicator in personality typology indicates how people think of the world and make decisions in their lives. The percentage of the four traits, mind, energy, nature, and tactics, decides people's MBTI personalities. Even though most psychologists do not recognize the indicator due to its poor validity, reliability, comprehensiveness, and correlated categories, MBTI personality prevails in both psychology and daily life. This paper has used fine-tuned BERT classification and TF-IDF vectorizer for the word embedding task to vectorize the tweets of different personalities. Famous machine learning models are also used to classify the 16 characters of MBTI in four binary classifications. The existing challenge throughout the task is the imbalanced dataset. The samples' numbers of each personality are not evenly distributed. Some personalities have more significant numbers of samples that are ten times more than other ones. As a result, any model's performance would decline in the training process because of the lack of training samples. Upsampling and downsampling are specific and robust solutions to the imbalance of data. They combine or disassemble the samples to get a balance in every category. However, this paper has tried both and gets more subpar performances than before, upsampled or downsampled. The negative impact of the imbalance stays along with the experiment. This research provides an alternative to lengthy and monotonous questionnaires. People can use their tweets or posts on other social media platforms to predict their personalities. It delivers a more accessible way for young people to follow the current trend.

References

[1] The Myers & Briggs Foundation. 2019. MBTI® Basics.https://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/

[2] NERIS Analytics Limited. 2013. Our Framework. https://www.16personalities.com/article/our-theory

[3] M.Farouk Radwan, MSc. 2022. How the words people say reflect their personalities. https://www.2knowmyself.com/How_the_words_people_say_reflect_their_personalities

[4] Rajaraman, A.; Ullman, J.D. (2011). "Data Mining". Mining of Massive Datasets. pp.1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.

[5] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv:1810.04805v2

[6] Cramer, J. S. (2002). The origins of logistic regression (Technical report). Vol. 119. Tinbergen Institute. pp. 167–178. doi:10.2139/ssrn.360300

[7] Altman, Naomi S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression" (PDF). The American Statistician. 46 (3): 175–185. doi:10.1080/00031305.1992.10475879. hdl:1813/31637

[8] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original (PDF) on 17 April 2016. Retrieved 5 June 2016.

[9] Tobias Bornheim, Niklas Grieger, and Stephan Bialonski. FHAC at GermEval 2021: Identifying German toxic, engaging, and fact-claiming comments with ensemble learning. In Proc. GermEval 2021 Workshop on Identification of Toxic, Engaging, and Fact-Claiming Comments: 17th KONVENS 2021, pages 105–111, Online (2021).

[10] Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's Next Language Model. In Proceedings of the 28th International Conference on Computational Linguistics, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[11] Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. ICLR.

[12] Kaggle.com. 2022 (MBTI) Myers-Briggs Personality Type Dataset. https://www.kaggle.com/datasets/datasnaek/mbti-type

[13] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze (2008). Introduction to Information Retrieval. Cambridge University Press. p. 27.

[14] Mannor, Shie & Peleg, Dori & Rubinstein, Reuven. (2005). The cross entropy method for classification. 561-568. 10.1145/1102351.1102422.