Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
15/2 (2022), 93-103. DOI: http://dx.doi.org/10.21609/jiki.v15i2.1052

# Myers-Briggs Type Indicator Personality Model Classification in English Text using Convolutional Neural Network Method

Joseph A. Sugihdharma[1], Fitra A. Bachtiar[2]

[1,2]Intelligent Systems Laboratory, Faculty of Computer Science, Universitas Brawijaya
Jl. Veteran No. 8, Kota Malang, 65145, Indonesia

*Email: [1]josephananda08@gmail.com, [2]fitra.bachtiar@ub.ac.id*

**Abstract**

Myers-Briggs Type Indicator (MBTI) is a personality model developed by Katharine Cooks Briggs and Isabel Briggs Myers in 1940. It displays a combination of preferences from four domains. Generally, test takers need to answer about 50 to 70 questions, and it is relatively expensive to know MBTI personality. The researcher developed a personality classification system using the Convolutional Neural Network (CNN) method and GloVe (Global Vectors for Word Representation) word embedding to solve this problem. The dataset used in this research consists of 8,675 data from the Kaggle site. The steps in this research are downloading the dataset from Kaggle, text preprocessing, GloVe weighting, classification using the CNN method, and evaluation using accuracy from the Confusion Matrix. Based on the tests carried out, using GloVe weighting can improve the model accuracy rather than random weighting. The best GloVe word dimensions depend on the metrics used to measure the model performance and the data of the classes contained in the dataset. From the CNN hyperparameter tuning test, the Adamax optimizer performs better and produces higher accuracy than the Adam optimizer. In addition, the CNN hyperparameter tuning increased model accuracy more significantly compared with the best GloVe word embedding dimensions.

**Keywords**: *classification, natural language processing, MBTI personality model, GloVe word embedding, convolutional neural network*

## 1. Introduction

Personality is one of the things that everyone needs to know. A person needs to know their character when interacting with people in everyday life, education, or a professional environment. 80 percent of the top 500 companies and 89 percent of the top 100 companies in the United States use Myers-Briggs Type Indicator (MBTI) personality test in the recruitment process. The MBTI personality test is a personality test that contains questions about a person's preferences in four different domains. The MBTI domains are distinguished based on how a person interacts with other people or the outside world, how a person learns and gathers information from the surrounding environment, how a person weighs and decides, and the way a person responds to the outside world around him [1]. By detecting and analyzing the personality, it can help the recruiter select the right candidate for the job. It also can help someone to choose the right career or academic courses and also bring out the best of themselves [2].

MBTI was developed in 1940 by Katharine Cook Briggs and her daughter Isabel Briggs Myers based on a personality typology by psychoanalyst Carl Jung [1]. This personality test instrument determines a person's personality by combining the dominant label of each MBTI domain [3]. Besides MBTI, there is another commonly used psychological test model, such as the Big Five. However, the Big Five model is more difficult to be classified either by machine learning or deep learning models compared to the MBTI model [4]. The difficulties of measuring the Big Five model are in the form of scores from each aspect, unlike the MBTI model, which classifies them into each domain class. Several obstacles were encountered in the previous research when trying to do the MBTI psychological test, including the relatively high cost and a large number of questions, which is around fifty to seventy questions as in [2]. It requires a lot of time to fill up all of those questions.

Researchers in previous studies have tried another way, namely by using user uploads on social media to find a person's character. Usually, researchers use posts on the PersonalityCafe.com forum to find a person's character, where every user has been labeled or asked to fill in their respective personality types. Several previous studies have tried to classify English text from the PersonalityCafe forum with various machine learning and deep learning algorithms. As in [2], the results show that Convolutional Neural Network (CNN) deep learning algorithm obtained higher accuracy than machine learning algorithm, namely Naïve Bayes, Random Forest, and Support Vector Machine. Other researchers compared the accuracy of the Naïve Bayes algorithm (NB), Support Vector Machine (SVM), and deep learning Recurrent Neural Network (RNN) Long Short Term Memory (LSTM) [5]. The results show that RNN-LSTM deep learning algorithm could surpass other machine learning algorithms, namely Naive Bayes and Support Vector Machine. To choose a better algorithm between CNN and LSTM to solve text classification, the previous researcher has conducted research that combines the pre-trained BERT model with other algorithms. The results show that BERT combined with CNN resulted in higher accuracy than the combination with LSTM [6].

Machine learning algorithms such as Naïve Bayes, Random Forest, and Support Vector Machine produce low accuracy due to many classification labels (multiclass classification). Besides, it caused high bias because several labels appeared more than the other labels (imbalance) [2]. Meanwhile, in deep learning Convolutional Neural Network (CNN), the results are pretty high because the researchers carried out a multilabel classification by breaking the labels of the MBTI personality into four binary classifications tasks and combining them. To overcome the low classification accuracy and high overfitting, the researcher suggests increasing the amount of data to improve the performance of the model [5]. In addition, previous researchers recommended trying other mechanisms for representing word embedding, such as word embedding char, k-char, or pre-trained GloVe (Global Vectors for Word Representation) word embedding.

In this study, we want to focus on maximizing the potential of the Convolutional Neural Network (CNN) algorithm in classifying the Myers-Briggs Type Indicator (MBTI) personality model. Based on the previous research, there are a lot of overlapped aspects in multiclass classification, and there is no obvious method to see the difference between those classes [5]. To overcome this problem, we tried to use binary classification, which shows more obvious differences and produces higher accuracy than multiclass classification. In this study, we implemented the suggestions from previous studies, namely the use of different word embedding. The word embedding we used in this study is a pre-trained word embedding, namely GloVe (Global Vectors for Word Representation). GloVe examines word relationship by calculating the ratio of the co-occurrence probability with other words [7]. To achieve our goal and keep focused on the research, we draw up some research questions and research objectives as shown in Table 1. Based on those research questions, we conducted four types of testing and analysis. The first and second tests are related to the GloVe weighting. The first test compared results from GloVe word embedding weighting with random weighting to find improvement of the accuracy when used in the Convolutional Neural Network model. And on the second test, we tested every GloVe dimension to find the best GloVe word embedding dimension. In the third test, we conduct a hyperparameter tuning test to measure the best accuracy that a CNN model can produce. The last test measures accuracy of the best GloVe dimension and best CNN hyperparameters combination.

The importance of knowing a person's personality makes researchers motivated to continue research on the MBTI model of personality classification without using questions that tend to be a lot and require a relatively long time, but by using English texts to find out a person's personality type. Based on previous research, we want to maximize the accuracy of the Convolutional Neural Network (CNN) algorithm by adopting the advantages of the previous research and combining it with GloVe (Global Vectors for Word Representation) word embedding. Thus, through this research, our contributions are:

1) We combine a CNN model and GloVe weighting, which was not implemented yet in previous research.
2) During the CNN hyperparameters tuning test, we found that optimizers could increase the accuracy in classifying MBTI domains. Previous research had never tuned this hyperparameter before.

## 2. Related Work

Previously, several researchers have tried to do some MBTI personality classification tasks with

**Table 1.** Research Questions and Research Objectives of This Research

| No. | Research Questions (RQ) | Research Objectives (RO) |
|---|---|---|
| 1 | How will the use of word embedding GloVe (Global Vectors for Word Representation) affect the accuracy of the Convolutional Neural Network model to classify MBTI personalities? | Identify the increase in the accuracy of the Convolutional Neural Network model to classify MBTI personalities by using GloVe (Global Vectors for Word Representation) word embedding. |
| 2 | Which GloVe (Global Vectors for Word Representation) word embedding dimensions can produce the best accuracy? | Identify which dimensions of GloVe word embedding (Global Vectors for Word Representation) produce the best accuracy. |
| 3 | How are the hyperparameters tuning on the Convolutional Neural Network that produces good accuracy? | Identify the hyperparameters combination on the Convolutional Neural Network that produces good MBTI personality classification accuracy. |
| 4 | How are the accuracy results from the combination of Convolutional Neural Network (CNN) best hyperparameters and best GloVe (Global Vectors for Word Representation) word embedding dimensions? | Identify the accuracy results from the combination of Convolutional Neural Network (CNN) best hyperparameters and best GloVe (Global Vectors for Word Representation) word embedding dimensions. |

various datasets and methods. Compared with this research, as in [2], the researchers use Convolutional Neural Network (CNN) as the deep learning classification method and gain the highest accuracy rather than other machine learning methods that the researchers use for comparison, which is 81,40%. The researchers used machine learning models as their model comparators, namely Naive Bayes, Random Forest, and Support Vector Machine, and could produce accuracy of 32,63%, 36,03%, and 57,90%, respectively. There are several differences even though the deep learning method is the same. The researchers use Term Frequency-Inverse Document Frequency (TF-IDF) as the feature extraction method, while in this research, we use values from Global Vectors for Word Representation (GloVe). As in [2], the researchers collected 8,000 data points from user posts at PersonalityCafe forums. Even though we used the dataset of user posts from the PersonalityCafe.com forum, we imported the dataset from the Kaggle site containing 8,675 data points.

There are previous researchers who had used the same dataset as ours but with different classification methods. As in [5], the researchers compared the accuracy of the Naïve Bayes algorithm (NB), Support Vector Machine (SVM), and deep learning Recurrent Neural Network (RNN) Long Short Term Memory (LSTM) [5]. The result shows that NB, SVM, RNN-LSTM could respectively produce accuracy of 32%, 34%, and 40% on training data, and 26%, 33%, and 38% on test data. Other researchers who use the same dataset as ours used BERT pre-trained model to classify MBTI personality. The best results from their experiments are 75,83% on Extrovert-Introvert (E-I) domain, 74,41% on Intuition-Sensing (N-S) domain, 75,75% on Feeling-Thinking (F-T) domain, and 71,90% on Judging-Perceiving (J-P) domain[8].

## 3. Theories

### 3.1. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm generally used to solve computer vision cases. However, this model can also be used to solve natural language processing cases [9]. As in [10], the researcher introduces a CNN model consisting of one convolutional layer with multiple filter widths and feature maps, max-pooling layer, and fully-connected layer with dropout and softmax output. The shallow and wide CNN architecture introduced by the researcher could surpass the accuracy of a CNN architecture with several or deep layers to do a text classification task [11].

The Convolutional Neural Network model for natural language processing accepts input of text or sentences represented in a matrix with rows of words amount in a sentence and columns of word embedding dimensions. Then, a convolution layer with a predetermined size and number of filters processes the sentence matrix and generates a feature map. After the sentence is processed, the highest value in each map will be taken at the max-pooling layer and combined with other feature maps to form a feature vector. The final layer receives input in the form of feature vectors and uses it for sentence classification [12].

### 3.2. Global Vectors for Word Representation (GloVe)

Word embedding or known as word representation can take the meaning of a word either semantically or syntactically from a word corpus that does not have a label [13]. GloVe (Global Vectors for Word Representation) is a word embedding developed by Stanford University researchers in 2014.

GloVe identifies the relation of a word by calculating the ratio of word occurrence probability together with other words. There are four types of GloVe (Global Vectors for Word Representation) word embedding trained with data from different sources. Each GloVe type has a different amount of word tokens and different variations of vector dimensions. GloVe word embedding used in this research comes from Wikipedia and Gigaword 5 (a collection of English-language news source networks) with 400 thousand words, 822 MB in size, and have four dimensions, namely 50, 100, 200, and 300 vector dimensions [7].

# 4. Methodology

There are several stages carried out in this research, namely data preprocessing, deep learning modeling, training and testing, and model evaluation. Each step will be explained in more detail as follows:

## 4.1. Dataset

The data used in this study is data in the form of English text uploaded by Mitchell Jolly on the kaggle.com site which consists of 8,675 data [14]. As in [15], this is one of the most used datasets to do the MBTI personality classification task. The dataset consists of two columns, which are posts and type. Each row of posts column contains the last fifty posts from a user in the PersonalityCafe.com forum. The second column contains the user MBTI type consisting of four letters representing four MBTI domains. As shown in Fig 1, the dataset consists of 1,999 and 5,676 rows of data from Extrovert and Introvert class respectively; 7,478 and 1,197 rows of data from Intuition and Sensing class respectively; 4,094 and 3,981 rows of data from Feeling and Thinking class respectively; 3,434 and 5,241 rows of data from Judging and Perceiving class respectively.

## 4.2. Data Preprocessing

After collecting the data, the first step we took was preprocessing the data. Several steps of text mining were carried out, such as removing special characters with regular expressions (RegEx), removing stopwords, and returning a word to its basic form with a lemmatization process. Then, we convert the MBTI label into zero and one, depending on the trained domain. Next, every sentence in the dataset is broken down into single elements or words in an array, then a word index in the form of key and value pairs is formed based on the occurrence of the word in the dataset. We convert each word into a numeric value based on the word index that has been created previously. The sentence lengths in the dataset will be made equal to a predetermined length, this process is known as the padding process. The last step in text preprocessing is to represent the studied text using GloVe word embedding. The GloVe equation is shown in Equation 1. The preprocessing steps can be seen in Fig 2.

$$GloVe = \sum_{i,j} f(X_{ij})({w_i}^T \widetilde{w}_j + b_i + \widetilde{b}_j - logX_{ij})^2 \tag{1}$$

## 4.3. Train, Validation and Test Data

In the next step, the data will be shuffled and divided into training data, validation data, and test data with a percentage of 70:15:15. We use training data for the model training process, validation data to evaluate the model performance during the training process, and test data to test the accuracy of the trained model.

## 4.4. Convolutional Neural Network Modelling and Training

After the data is divided, the next step is to create a Convolutional Neural Network (CNN) model. At this stage, we select and arrange the model layers. We define the hyperparameters of each layer of the CNN model. In Fig. 3, the Convolutional Neural Network model accepts input in the form of text or sentences that have been encoded and represented in a matrix with rows of words amount in a sentence and columns of word embedding dimensions. The sentence matrix will be processed in a convolution layer and the highest value from each feature map will be taken at the max-pooling layer and concatenated with other feature maps value to form a feature vector. The feature vector will be used for sentence classification. The output form of this developed CNN is the probability of a sentence belonging to each of the available classes. The CNN process can be seen in Figure 3.

## 4.5. Testing and Analysis

Furthermore, we conduct several tests on the trained model with the test data to determine whether the results of the trained model are good enough or can be improved again. If the results of the trained model still have the potential to be improved, the model will be re-trained with different hyperparameters to produce higher accuracy. Specifically, in this

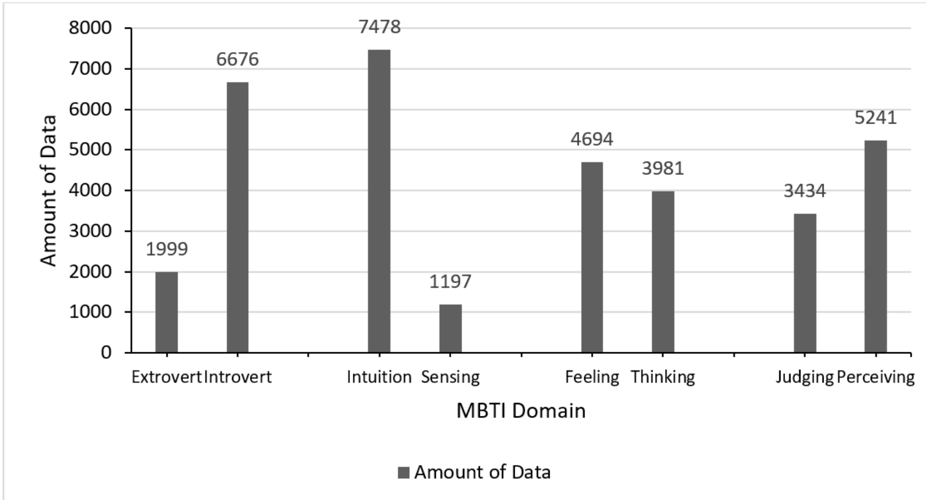


**Figure 1.** MBTI domains data ratio

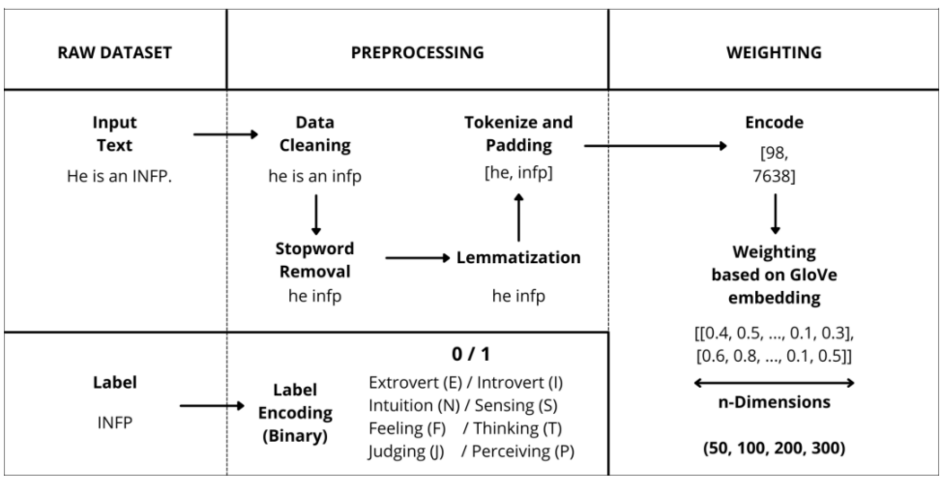


**Figure 2.** Data preprocessing diagram

study, we conducted four tests, which are comparing results from random and GloVe weighting, GloVe word embedding dimension testing, CNN hyperparameter tuning test, and combining the best GloVe word embedding dimensions and CNN hyperparameters. The values of the hyperparameters in the CNN hyperparameter tuning test are shown in Table 2.

**Table 2.** Tested Hyperparameter Values

| Hyperparameter | Values |
|---|---|
| Optimizer | Adam, Adamax |
| Learning Rate | 0,1; 0,07; 0,05; 0,03; 0,01 |
| Dropout | 0,5; 0,3; 0,2; 0,1 |
| Filter Size | (3,4,5), (2,3,4) |

## 4.6. Comparing to Other Methods Performance

To measure the performance of the developed model, which was a CNN model trained with GloVe word embedding, the researchers compare the accuracy produced by this model with other machine learning models. We choose Naïve Bayes as the baseline model, Decision Tree, and Random Forest classifier. In addition, we compare the proposed method with LSTM and BERT results from previous research.

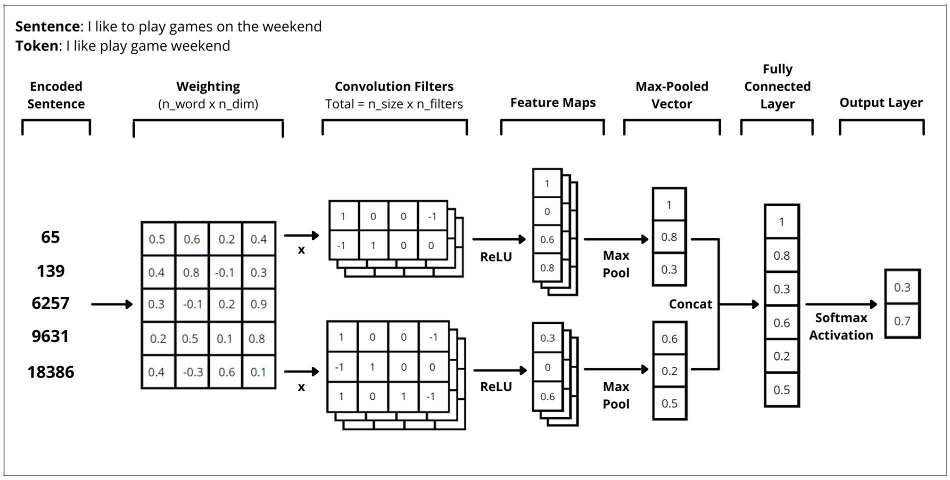


**Figure 3.** Convolutional Neural Network diagram

## 4.7. Model Evaluation

We evaluate the proposed method result on test data using accuracy and F1-Score metrics to measure the model performance. The equation of the metrics is shown in Equation 2 and 3. Accuracy measures the proportion of true results to total cases, while F1 Score measures the balance of Precision and Recall.

$$Accuracy = \frac{\textit{correctly classified data}}{\textit{all test data}} \quad (2)$$

$$F1 - Score = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}} \quad (3)$$

# 5. Results and Discussion

We divide the results and discussion section into several parts. This section will discuss the tests carried out in this research, comparison with machine learning methods and previous research, and error analysis. The detailed results and discussion are as follows:

## 5.1. Random and GloVe Weighting Test

In the random and GloVe weighting tests using the CNN algorithm, we tested different dimensions available in the GloVe word embedding, namely 50, 100, 200, and 300. We examine each word embedding dimension using random weighting and GloVe weighting. In this test, we use baseline hyperparameters, which are 0,5 dropout; 0,1 learning rate; and 3,4,5 filter sizes. Table 3 shows the accuracy of the CNN model when trained with random and GloVe weighting.

Based on the test results, we concluded that the CNN model trained using GloVe weighting could produce higher test data accuracy than the CNN model trained using random weighting or weighting without GloVe. Because GloVe word embedding had previously trained with large corpus from Wikipedia and Gigaword 5 (a collection of English-language news source networks), its vector representation brings external knowledge to our classification task[16]. The MBTI model only needs a slight update of the embedding weight value to reach the convergence point.

## 5.2. Word Embedding Dimension Test

In this section, we tested the available GloVe word embedding dimensions. We use baseline hyperparameters in this test, namely 0,5 dropout; 0,1 learning rate; and 3,4,5 filter sizes. The result of this test is the same as shown in Table 3 because it tested using various GloVe dimensions. Based on the results, we could see Intuition-Sensing (N-S) domain accuracy decreased when the GloVe word embedding dimensions increased. On the other hand, other MBTI domains' accuracy gains the highest accuracy on the largest GloVe dimensions. When we look at the F1-Score, all MBTI domains reached the highest F1-Score on the largest GloVe dimensions. Even if the accuracy of the N-S domain decreased as we increased the dimensions, oppositely, the F1-Score increased. As in [17], a higher number of word embedding dimensions could provide better semantic representation, but on the other hand, it increases the computation cost. Therefore, we conclude that

**Table 3.** Random and GloVe Weighting Test Result

| No. | Weighting | Dimensions | Accuracy | | | | F1-Score | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | E-I | N-S | F-T | J-P | E-I | N-S | F-T | J-P |
| 1 | Random | 50 | 61,81 | 78,22 | 53,53 | 50,72 | 73,29 | 24,52 | 44,13 | 54,5 |
| 2 | Random | 100 | 61,58 | 77,99 | 52,84 | 50,55 | 73,16 | 23,18 | 51,94 | 53,11 |
| 3 | Random | 200 | 58,59 | 80,55 | 53,1 | 49,64 | 68,34 | 24,12 | 47,17 | 51,06 |
| 4 | Random | 300 | 51,27 | 76,51 | 53,74 | 49,21 | 61,07 | 27,81 | 49,89 | 50,84 |
| 5 | GloVe | 50 | 70,38 | **85,15** | 63,11 | 55,36 | 79,84 | 23,28 | 55,78 | 63,47 |
| 6 | GloVe | 100 | 71,67 | 83,67 | 60,98 | 53,36 | 81,73 | 22,27 | 59,72 | 59,85 |
| 7 | GloVe | 200 | 70,91 | 84,16 | 64,53 | 59,46 | 80,25 | 29,89 | 63,94 | 67,41 |
| 8 | GloVe | 300 | **73,41** | 83,43 | **66,09** | **60,37** | **83,11** | **30,73** | **64,77** | **68,71** |

the largest GloVe word embedding dimensions are the best GloVe dimensions.

## 5.3. Hyperparameter Tuning Test

The next test is tuning the hyperparameter of the CNN model. We adjust the optimizer, learning rate, dropout, and filter size hyperparameters value in this test. The word embedding dimension used in this test is 50 dimensions. The value of the hyperparameters tested in this test is shown in Table 2.

After we did hyperparameter tuning to our model, we concluded that the Adamax optimizer could perform better than the Adam optimizer to solve this classification task. As in [18], Adamax performs better when used in a classification task with sparse parameter updates such as word embeddings. This optimizer only updates the variable value used on a forward pass, and the other variable value remains the same. As a result, the model could reach the convergent point faster. Based on the results, we concluded that a smaller learning rate value could increase the accuracy of the developed model. Oppositely, a higher dropout value can increase the developed model accuracy. The filter size that can produce the best accuracy depends on the ratio of the data or dataset used. By combining all the best values from each hyperparameter, we could achieve results as shown in Table 4.

## 5.4. Combination of Best GloVe Dimension and CNN Hyperparameters

The fourth test combines the best GloVe word embedding dimensions with the best CNN hyperparameters. The optimizer used in this test is the Adamax optimizer. We chose this optimizer because it produced better accuracy than Adam in the CNN hyperparameter tuning test. The best GloVe word embedding dimension for all MBTI domains is 300. In the Extrovert-Introvert (E-I) domain, the Feeling-Thinking (F-T) domain, and the Judging-Perceiving (J-P) domain, the CNN hyperparameter that can produce the best accuracy is 0,01 learning rate; 0,1 dropout; and 2,3,4 filter sizes. Unlike the Intuition-Sensing (N-S) domain, the best combinations of CNN hyperparameters of this domain are 0,01 learning rate; 0,2 dropout; and 3,4,5 filter sizes. We got the best combinations of CNN hyperparameters above by doing a manual grid search.

As shown in Table 5, the results obtained by combining the GloVe dimensions and the best CNN hyperparameters are significantly improved compared to the accuracy at baseline using Adam. Based on the results, there is a slight difference between the best hyperparameter combinations accuracy with the highest accuracy. Therefore, we conclude that hyperparameter tuning increased the model accuracy more significantly, compared to the effect of the best GloVe word embedding dimensions. Even though the N-S domain's F1-Score is still low compared to the other MBTI domains, we could see that the F1-Score is two times higher than the baseline value.

## 5.5. Model Performance Compared to Other Methods

The result shown in Table 6 is the comparison of accuracy reached by the best CNN models trained by the combination of best hyperparameters and GloVe dimensions with other machine learning methods. Based on Table 6, we conclude that the accuracy from the proposed model that we develop could exceed the maximum accuracy of machine learning models. Table 7 below shows the comparison of our work with the previous research. Based on the result, we concluded that CNN combined with GloVe word embedding could improve the accuracy to classify MBTI domains and exceeded the accuracy of the BERT pre-trained neural network.

## 5.6. Error Analysis

After we did model training and tuning, additionally we did an error analysis by testing our model with some random hand-picked sentences

**Table 4.** Best CNN Hyperparameters Combination Result

| No. | Combinations | Accuracy | | | | F1-Score | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | E-I | N-S | F-T | J-P | E-I | N-S | F-T | J-P |
| 1 | Baseline (Adam) | 70,38 | 85,15 | 63,11 | 55,36 | 79,84 | 23,28 | 55,78 | 63,47 |
| 2 | Best Hyperparameters (Adam) | 78,76 | 87,66 | 74,54 | 71,18 | 87,14 | 33,03 | 71,5 | 77,63 |
| 3 | Best Hyperparameters (Adamax) | **81,15** | **87,75** | **80,08** | **75,1** | **88,37** | **34,95** | **78,14** | **79,96** |

**Table 5.** Combinations of best GloVe dimension and CNN hyperparameters

| No. | Combinations | Accuracy | | | | F1-Score | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | E-I | N-S | F-T | J-P | E-I | N-S | F-T | J-P |
| 1 | Baseline (Adam) | 70,38 | 85,15 | 63,11 | 55,36 | 79,84 | 23,28 | 55,78 | 63,47 |
| 2 | Best GloVe Dimensions (Adam) | 73,41 | 83,43 | 66,09 | 60,37 | 83,11 | 30,73 | 64,77 | 67,41 |
| 3 | Best Hyperparameters (Adamax) | 81,15 | 87,75 | 80,08 | 75,1 | 88,37 | 34,95 | 78,14 | 79,96 |
| 4 | Best GloVe + Hyperparameters (Adamax) | **81,73** | **89,75** | **80,81** | **76,02** | **88,84** | **47,24** | **79,04** | **80,84** |

**Table 6.** Comparison with Other Methods

| Methods | Best Accuracy | | | |
|---|---|---|---|---|
| | E-I | N-S | F-T | J-P |
| Naive Bayes | 26,79 | 18,02 | 48,35 | 39,49 |
| Decision Tree | 72,19 | 82,52 | 51,24 | 55,79 |
| Random Forest | 76,69 | 86,43 | 55,37 | 60,81 |
| CNN + GloVe (Our Proposed Model) | **81,73** | **89,75** | **80,81** | **76,02** |

**Table 7.** Comparison with Previous Work

| Methods | Best Accuracy | | | |
|---|---|---|---|---|
| | E-I | N-S | F-T | J-P |
| LSTM[5] | **89,51** | **89,85** | 69,09 | 67,65 |
| BERT[8] | 75,83 | 74,41 | 75,75 | 71,90 |
| CNN + GloVe (Our Proposed Model) | 81,73 | 89,75 | **80,81** | **76,02** |

from the raw dataset. We took five random hand-picked sentences from each MBTI domain from the raw dataset. Table 8 shows some of our hand-picked sentences with their labels and the amount of correctly predicted MBTI domains. Usually, in sentiment classification, we could easily see whether a word in a sentence tends to be categorized as positive or negative. On the other hand, in personality classification especially by using text dataset, it is not easy to determine whether a word is mostly used in Extrovert or Introvert class. By seeing text data in Table 8, one of the patterns that cause the model to classify a text incorrectly is when a text mention or contains an MBTI label in it. As in sentences number 3 and 4, the text contains ”INTJ” and ”ENTP” within the text, which makes the model misclassify this sentence as INTJ and ENTP.

We did some simple Exploratory Data Analysis (EDA) that shows the 15 most common words found in a certain class in a domain. As shown in Fig. 4 and Fig. 5, we could see that the top 5 of most common words found in this domain are the same words and same order between both classes, which are ”like”, ”think”, ”get”, ”people”, and ”know”. Between 15 top common words, 14 words appeared in both classes. That is one of the reasons why the model misclassifies the final class. Additionally, while testing our model with some random hand-picked sentences previously, we got results of classification as shown in Fig. 6. Surprisingly, if we take a closer look at this figure, we could see that this figure has a similar graphical pattern with the dataset ratio shown in Fig. 1. We conclude that the data ratio in the dataset affects the accuracy of the model. For example, we took the first domain, the number of sentences classified as an introvert is much higher than extroverts. The data ratio also shows that introverts' data are much higher than extroverts. This result was affected by the frequency of the word appearing in the dataset. As shown in Fig. 4 and Fig. 5, we could see that ”like” is the most common word of both classes, but the frequency of this word between both classes is far apart, which ”like” in introvert class appeared three times higher than in extrovert class.

## 6. Conclusion

Based on the conducted tests, we conclude a few things. We conclude that the model trained using GloVe word embedding weighting resulted produced higher accuracy than the model trained using random weighting. The best GloVe word embedding dimensions depend on the data of the classes contained in the dataset. When measuring the model performance using accuracy, the smallest dimension of GloVe word embedding is the best dimension when used in the Intuition-Sensing (N-S) domain. Meanwhile, the largest GloVe dimension is the best when used on the other MBTI domains, which are the Extrovert-

**Table 8.** Error Analysis Data on Random Hand-Picked Sentences

| No. | Original Text | True Labels | Predicted Labels | Correct Domains |
|---|---|---|---|---|
| 1 | "I feel a need for power. If there's a group of people that I'm in, I have to be in charge at some level, if not the top (the top only if I'm qualified). I will not be a bottom rank. If I think it is..." | INTP | INTP | 4 |
| 2 | "What about being a professor for an online course? At the college I'm at, I never see my online professors, all work is done online, and I go up to the college to the testing center for tests. Just..." | ISTP | INTP | 3 |
| 3 | "Yes to this!! My INTJ coworker is my favorite for those same reasons. Also, my old ESTP roommate and I would have killed each other if not for the INTJ next door. :tongue:" | ESTJ | INTJ | 2 |
| 4 | "Something funny to go along with that... I'm dating an ENTP and he's starting to buy me random gifts. I don't think he quite understands that I go for functional things rather than just random toys...." | ESTJ | ENTP | 2 |
| 5 | "I had a hard time choosing between self-confidence, self-esteem, healthy relationship, and maintaining a calm, peaceful mind. I went with self-confidence in the end." | ISFJ | INTP | 1 |
| 6 | "I haven't met many people who deny being extroverts, but again most people are not aware of what being extroverted means." | ESFJ | INTP | 0 |

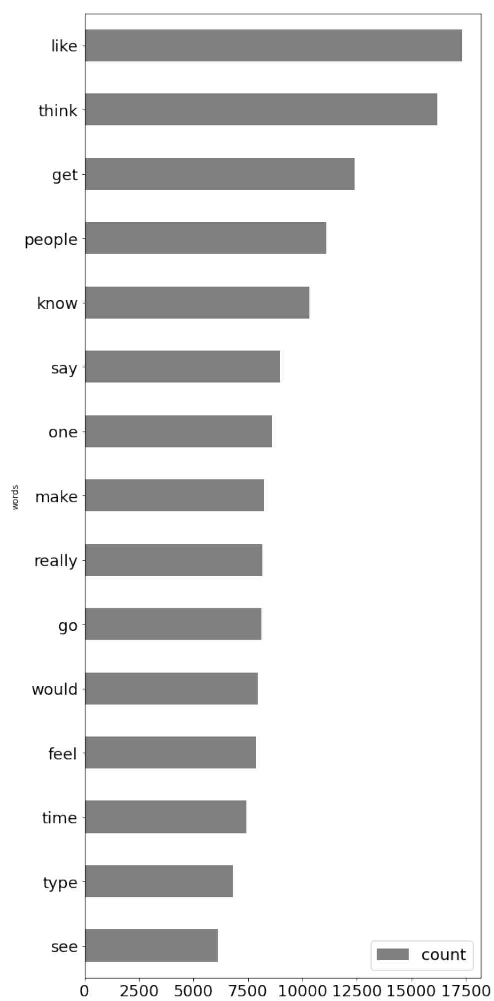


**Figure 4.** Common words found in dataset (Extrovert)

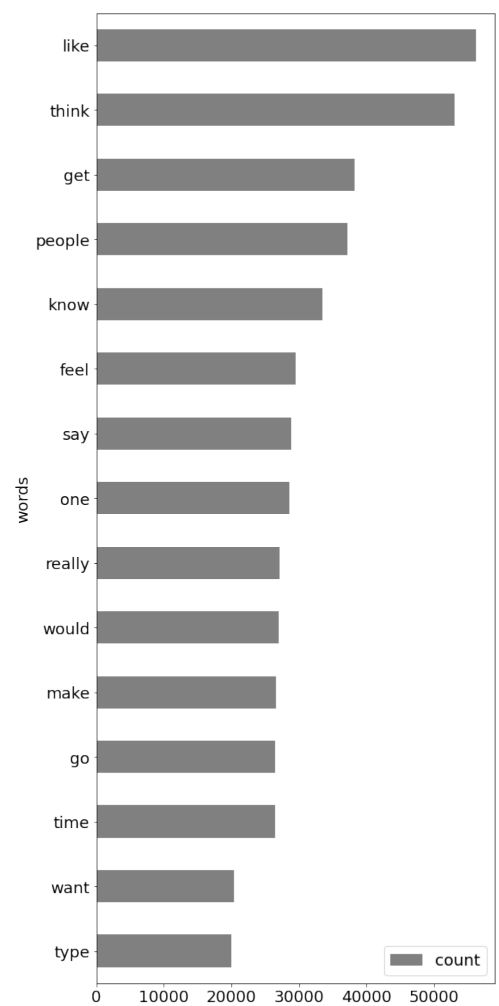


**Figure 5.** Common words found in dataset (Introvert)

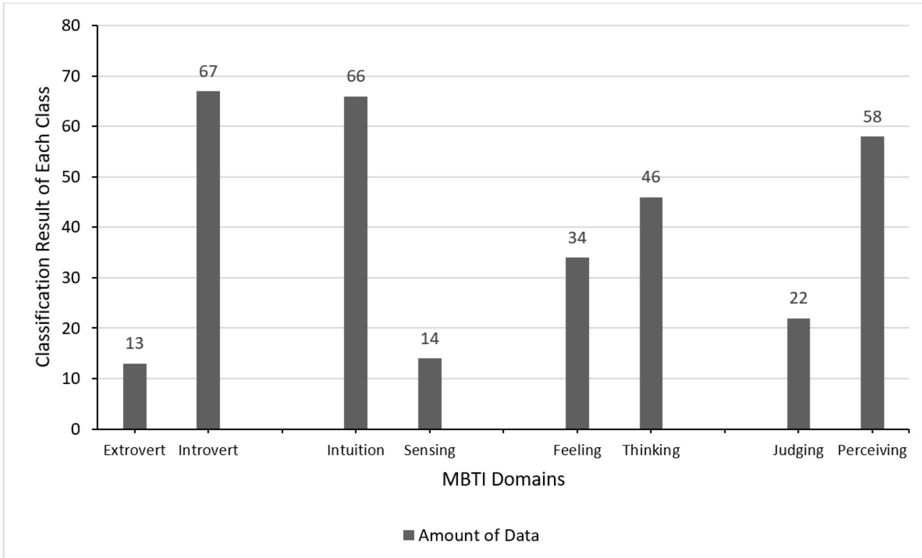


**Figure 6.** Classification results of random hand-picked sentences

Introvert (E-I) domain, Feeling-Thinking (F-T) domain, and Judging-Perceiving (J-P) domain. When we take F1-Score into account, the largest dimension of GloVe word embedding produces the highest F1-Score in all MBTI domains. In the CNN hyperparameter tuning test, we concluded that the Adamax optimizer could perform and generate higher accuracy than the Adam optimizer. When the best GloVe word embedding dimensions and the best combination of hyperparameters are combined, the model could generate higher accuracy than using only one of them. In addition, the CNN hyperparameter tuning increased model accuracy more significantly compared with the best GloVe word embedding dimensions.

We recommend several suggestions for model improvement in further research. First, we recommend using over-sampling or under-sampling methods to balance the data ratio, especially in a domain that has severe data ratio imbalance. Second, we recommend using different data sampling or splitting methods such as K-Fold Cross Validation to better accuracy. Third, the One-Hot Encoding method can be used on labels from the dataset and compared with the existing system that uses Label Encoding. Fourth, additional features, such as emotional features, may determine the relationship between a personality type and emotions. We expect these suggestions to increase the evaluation value of the developed system.

# References

[1] S. Cloninger, *Theories of Personality: Understanding Persons*, 4th ed. Pearson-/Prentice Hall, 2004. [Online]. Available: https://books.google.co.id/books?id=wN_5nQEACAAJ

[2] T. Pradhan, R. Bhansali, D. Chandnani, and A. Pangaonkar, "Analysis of personality traits using natural language processing and deep learning," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, Jul. 2020, pp. 457–461.

[3] D. Schultz and S. Schultz, *Theories of Personality*, 11th ed. Cengage Learning, 2016. [Online]. Available: https://books.google.co.id/books?id=oStTCwAAQBAJ

[4] F. Celli and B. Lepri, "Is big five better than mbti? a personality computing challenge using twitter data." in *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it)*, Torino, Italy, Dec. 2018, pp. 93–98.

[5] B. Cui and C. Qi, "Survey analysis of machine learning methods for natural language processing for mbti personality type prediction," 2017.

[6] Z. Ren, Q. Shen, X. Diao, and H. Xu, "A sentiment-aware deep learning approach for personality detection from text," *Information Processing & Management*, vol. 58, no. 3, p. 102532, 2021.

[7] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representa-

tion," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, vol. 14. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.

[8] S. S. Keh and I. Cheng, "Myers-briggs personality classification and personality-specific language generation using pre-trained language models," *CoRR*, vol. abs/1907.06333, 2019.

[9] W. Agastya, H. Haryanto *et al.*, "Ekstraksi emosi majemuk kalimat bahasa indonesia menggunakan convolutional neural network," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 10, no. 2, pp. 148–155, 2021.

[10] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751.

[11] H. T. Le, C. Cerisara, and A. Denis, "Do convolutional networks need to be deep for text classification?" in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*. New Orleans, USA: AAAI Press, Feb. 2018.

[12] C. Yao, Y. Qu, B. Jin, L. Guo, C. Li, W. Cui, and L. Feng, "A convolutional neural network model for online medical guidance," *IEEE Access*, vol. 4, pp. 4094–4103, 2016.

[13] S. Lai, K. Liu, S. He, and J. Zhao, "How to generate a good word embedding," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 5–14, 2016.

[14] M. Jolly, "(mbti) myers-briggs personality type dataset." [Online]. Available: https://www.kaggle.com/datasnaek/mbti-type

[15] Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria, "Recent trends in deep learning based personality detection," *Artificial Intelligence Review*, vol. 53, Apr. 2020.

[16] P.-H. Arnoux, A. Xu, N. Boyette, J. Mahmud, R. Akkiraju, and V. Sinha, "25 tweets to know you: A new model to predict personality with social media," in *Proceedings of the international AAAI conference on web and social media*, vol. 11, no. 1, 2017, pp. 472–475.

[17] A. Borah, M. P. Barman, and A. Awekar, "Are word embedding methods stable and should we care about it?" in *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*, 2021, pp. 45–55.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.