Metric Comparison Performance For Text Classification
Amri Muhaimin *, Tresna Maulana Fahrudin1, Trimono1, Prismahardi Aji Riyantoko1, Kartika Maulida Hindrayani1

Abstract: Text classifications have been popular in recent years. To classify the text, the first step that needs to be done is to convert the text into some value. Some values that can be used, such as Term Frequencies, Inverse Document Frequencies, Term Frequencies – Inverse Document Frequencies, and Frequency of the word itself. This study aims to get which metric value is best in text classification. The method used is Naïve Bayes, Logistic Regression, and Random Forest. The evaluation score that is used is accuracy and Area Under Curve value. It comes out that some metric values produce similar evaluation scores. Another finding is that Random Forest is the best method among others, also the best metric for text classification is Term Frequencies – Inverse Document Frequencies.

Keywords: classification, metric, text

1. Introduction

In this era of big data, data is no longer in the form of numbers, but the text is also data that can be analyzed. In its development, texts have been widely researched and used as the primary material for artificial intelligence. The methods used are also quite varied, such as predictive text for the google search field, sentiment analysis on specific subjects and objects, and topic modeling in the text.

In the process, the text is first processed into a metric. Then the metric is processed using algorithms. In converting text into metrics, several types of metrics can be used, including Term Frequencies (TF), Inverse Document Frequencies (IDF), binary numbers (0, 1), TF-IDF, and word frequency. We are free to choose which numbers are used in classifying the text we have.

The research is generally done by classifying texts using various methods. Meanwhile, this study aims to compare the results of modeling performance using different input metrics. In short, it is to compare metrics such as TF, IDF, TF-IDF, Frequency, and Binary numbers using several classification algorithms such as Naïve Bayes, Logistics Regression, Random Forest, and Possibly Neural Network. So that information related to metrics can be obtained that may be used in classifying the public.

In this research, only supervised learning schemes are used. So that the measuring tools used to evaluate the goodness of the model with different metrics are accuracy and Area Under Curve (AUC). Then the tool used in this research is R, and the data used comes from Kaggle about coronavirus tweets and their sentiment. For the rest of the section of this study, there are Related Works, Experiment and Analysis, and Conclusions.

2. Related Works

Some papers that used to similar this study are hard to find. Therefore, we used the classification method to create this section. We review some papers that used classification and use the feature in various metrics.
The model-based called fastText is used in [1]. This algorithm used neural networks as the main based algorithm. The features that are used are the n-gram and also the bag of words from the text. In this research, they used 4 baselines to compare the model. And fastText has overcome those models. The main idea of this model is to create a low dimensional feature but useful.

Then, researchers [2] used BERT-BIGRU for text classification. The output activation function that is used is softmax in this model. BERT-BIGRU contains 2 structures, which are BERT and BIGRU. The BERT structure is similar to the Transformer structure. It only used the Encoder of the Transformer for its structure. BERT structure is used to generate the words for the feature in the model. On the other hand, BIGRU is just the layer that contains GRU from RNN. Furthermore, this model just creates some new way to generate the feature for text classification.

A comprehensive study is carried out by [3]. In short, it explains some methods that can be used for text classification. Such as Rocchio Classification, KNN Classification, Naïve Bayes Classifier, SVM Classification, Regression-Based Classification, Neural Network-Based Classification, Rule-Based Classification, and Decision Tree Classification. Each method contains pros and cons in its application. Other than the method, the important thing of the text classification is the preprocessing stage. The stage contains building the feature for the model and also cleaning the data.

Another Researcher uses the deep learning method to classify text classification [4]. The architecture is Long Short-Term Memory. The feature that is used to classify the data is the GloVe feature. It is a combination between word2vec and their algorithm so the GloVe can overcome the word2vec method

The last one is Multilabel Text Classification done by [5]. The method that is used is Long-Term Memory using word2vec as the feature. The performance is quite good. But it will be better if the researchers use another method as a baseline. In short, the Long-Term Memory can be classified as a multilabel case.

This study aims to compare the performance of the metric used in the classification model. To some of the related works above, this study is kind of a new thing. The expectation for this study is that get the information on which metric is the best metric for text classification.

3. Experiment and Analysis

This section is quite simple. After we get the data then we proceed into some algorithms for each metric and evaluate it using the confusion matrix score such as accuracy and AUC. Before that, this section will explain the data and the method used in this study.

3.1. Dataset

We get the data from Kaggle. The data is called coronavirus tweet sentiment. So we are going to classify the sentiment from each tweet using the different metrics with the same algorithms or method. The data structure can be seen in Table 1. It is about the sentiment of the coronavirus event that hit society in 2020. We use the training testing concept and not cross-validation because we want to make it simple. The number of training data is 41157 and the number of testing data is 3798. There are 5 categories for the target, namely extremely negative, negative, neutral, positive, and extremely positive. Even though this category is about sentiment, we still handle it just like a classification case and do not use rule-based classification.

<table>
<thead>
<tr>
<th>ID</th>
<th>Document</th>
<th>Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Doc 1</td>
<td>Positive</td>
</tr>
</tbody>
</table>

Table 1. Text Classification Data Structure
3.2 Analysis Stage

This study provides 5 stages. Simply, from the first stage, we get the data from the Kaggle, then we preprocess it as text data processed as usual. After the preprocessing stage, we calculate the 5 metrics for classification. They are TF, IDF, TF-IDF, Word of Frequency, and Binary number. From each metric we model the data using at least 3 algorithms, they are Naïve Bayes Classifier, Logistic Regression, and Random Forest. We only use popular and simple algorithms. After that, we evaluate each result using the testing set, we calculate the accuracy and AUC from each metric.

In preprocessing stage, we just use simple cleaning techniques. Such as case folding, removing stopwords, removing punctuation, and tokenizing. We don’t apply n-gram here, because it will be more complex if we use n-gram. We only use a bag of words or it means one gram or one word that we got from tokenizing process. After we are done with the tokenizing process, we can get the frequency of the word, with that value then we calculate the TF, IDF, and TF-IDF values. For the binary number, we just convert the frequency value into zero one, if the value is more than one then it should be one and otherwise. To elaborate on the metric value, the formula is written in Table 2.

<table>
<thead>
<tr>
<th>Metric Value</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term Frequency (TF)</td>
<td>$TF(t,d) = \frac{\text{number of times } t \text{ appears in } d}{\text{total number of terms in } d}$</td>
</tr>
<tr>
<td>Inverse Document Frequency (IDF)</td>
<td>$IDF(t) = \log \frac{N}{1 + df}$</td>
</tr>
<tr>
<td>$N$ is the total number of documents</td>
<td></td>
</tr>
<tr>
<td>$df$ is the number of documents with the term $t$</td>
<td></td>
</tr>
<tr>
<td>TF-IDF</td>
<td>$TF - IDF(t,d) = TF(t,d)IDF(t)$</td>
</tr>
<tr>
<td>Frequency of Word</td>
<td>Number of a word occurring in each document</td>
</tr>
</tbody>
</table>

The modeling stage is the stage where each data with each metric is modeled. We use the train-test scenario. The number of training and testing sets is mentioned. After we create the model, we test it into the testing set. Because this case is not binary classification, so we use accuracy and area under curve (AUC) value to compare each model.

3.3 Experimental Result

The experiment uses the feature from the token and creates the Document Term Matrix (DTM) using the tm package in R. After we create the DTM, the sparsity comes out 100%. So we reduce the sparsity percentage, by removing the column that has sparse terms over 95%. After that step, there are only 16 features left. To avoid different features on the training and testing set, we create the DTM simultaneously. Then we split the data frame into training and testing sets.
The modeling part is done by modeling each metric one by one. We create the model from the metric within the design. Let’s say, first we model the classification model using the Frequency metric, and then we calculate the accuracy and the AUC value. That’s step is repeated until all the metrics are modeled and evaluated. The result is stored in Table 3.

Table 3. Experimental Result

<table>
<thead>
<tr>
<th>Metric</th>
<th>Method</th>
<th>Accuracy</th>
<th>AUC</th>
<th>Average Accuracy</th>
<th>Average AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF</td>
<td>Naïve Bayes</td>
<td>0.2753</td>
<td>0.6151</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>0.3169</td>
<td>0.6169</td>
<td>0.2991</td>
<td>0.6157</td>
</tr>
<tr>
<td></td>
<td>Logistic</td>
<td>0.3051</td>
<td>0.6151</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDF</td>
<td>Naïve Bayes</td>
<td>0.2426</td>
<td>0.5872</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>0.2937</td>
<td>0.5592</td>
<td>0.2757</td>
<td>0.5842</td>
</tr>
<tr>
<td></td>
<td>Logistic</td>
<td>0.2908</td>
<td>0.6063</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Regression</td>
<td>0.2924</td>
<td>0.6068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF-IDF</td>
<td>Naïve Bayes</td>
<td>0.2753</td>
<td>0.6151</td>
<td>0.2988</td>
<td>0.6161</td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>0.3159</td>
<td>0.618</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Logistic</td>
<td>0.3051</td>
<td>0.6151</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Naïve Bayes</td>
<td>0.246</td>
<td>0.495</td>
<td>0.2766</td>
<td>0.5535</td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>0.2914</td>
<td>0.5586</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Logistic</td>
<td>0.2924</td>
<td>0.6068</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We calculate the AUC score using the multiclass rule, not the binary rule. If we look at the average accuracy and average AUC, the TF-IDF metric overcomes all the metrics. But if we look at the method in each metric, it seems the TF metric and TF-IDF metric are not quite different. The worst one is the Frequency metric in terms of average AUC, and the IDF metric in terms of average accuracy. There is additional information here, the Random Forest is the best method among others in each metric score.

4. Conclusions

From this study, we conclude that the TF-IDF metric is the best in terms of text classification. Besides, there is no big difference between the performance of the TF metric and the TF-IDF metric. TF-IDF metric is formed by multiplying between TF metric and IDF metric. In terms of sparsity, all metrics produce the same sparsity. By methodology, in this case, Random Forest comes out as the best method to classify the text. For future work, we can adjust the preprocessing step, and the method to define the feature that is used in the text classification. Also, the new metric can be developed to add more references in the text classification.

Author Contributions: Muhaimin and team members are writing this paper
Funding: This research received no external funding
Data Availability Statement: The data can be downloaded from [here](#)
Conflicts of Interest: The author declares no conflict of interest
References


