JISTE (Journal of Information System, Technology and Engineering)
Volume 1, No. 2, pp. 49-53
E-ISSN: 2987-6117
http://gemapublisher.com/index.php/jiste
Received: April 2023
Accepted: May 2023
Published: June 2023

Application of Naïve Bayes Classification to Analyze Performance Using Stopwords

Jefriyanto Jefriyanto[1], Nur Ainun[2], Muchamad Arif Al Ardha[3]

Universitas Negeri Padang[1], Universitas Serambi Mekkah [2], Universitas Negeri Surabaya [3]

Correspondence Email: jefriyanto@fe.unp.ac.id[1]

Abstract

Based on current data, there has been an increase in social media users, which shows that more and more people are using social media as a place to express themselves and their emotions. This will generate thousands of tweets within a day. The tweet data is processed so that it is useful for stakeholders who need it to help them make a decision. Because sentence structures on social media are often irregular, pre-processing is necessary to make tweet sentences normal. Stemming and Stopwords are pre-processing techniques that are widely used in sentiment analysis. In previous studies, there were indications that its use did not have a significant effect on accuracy. In this study, the authors divide it into four models: using stemming and stopwords and without using stemming and stopwords. Data using stemming gets the best results with an f1-score of 65%. These results indicate an increase in performance in the use of stemming and stopwords using Multi-class Naive Bayes.

Keywords: social media, stopwords, naïve bayes.

INTRODUCTION

According to Simon Kemp (2022) there were an additional 2.1 million internet users and more than 21 million active social media users. Twitter is one of the social media accounts for 18.45 million users in Indonesia. This shows that more and more people are using social media as a place and means to express themselves in the form of tweets or opinions in all aspects. So that thousand and even millions of tweets can occur within a day on social media. Twitter is widely used by government agencies, industry, education, and business to solve daily problems. Compared to other social media, this provides an API that helps its users collect tweet data in the form of text. When a post or tweet occurs, it often contains information or the emotional state of the person who made it. Even when someone criticizes a policy, they indirectly show their emotions. Basically, a sentiment is not only between positive and negative (Binary Class), but can also be in the form of love, joy, anger, sadness, or fear (Multi Class).

Sentiment analysis is a research branch of text mining with a focus on analyzing opinions in a text. Sentiment analysis on social media attracts many researchers around the world. One of them is Twitter, a social media site that allows users to express themselves freely. In the business world, sentiment analysis is used to analyze customer opinions about products and services. The data that has been processed is information for stakeholders to assist in decision-making and in deciding the target market. The structure of the word comments on social media is irregular and contains many abbreviations in the sentence structure; this is a challenge in carrying out the sentiment analysis process, so the use of pre-processing is very important so that it can affect the accuracy of the sentiment analysis process. Pre-processing is also very critical in text analysis. Therefore, pre-processing aims to organize and clean text data originating from social media before being further processed in classification.

Stemming and stopwords are pre-processing techniques used in text analysis. Stemming is useful for changing a word into its base word. Meanwhile, stopwords eliminate words that often appear but have no meaning. In previous research, there was an indication that stemming and stopwords have no effect on accuracy. In previous research with Binary classes using the SVM Classifier, it was shown

that the use of stemming and stopwords did not significantly affect accuracy; using stemming without stopwords only increased accuracy from 81.4% to 81.44%. Whereas in other studies with Binary classification using SVM and the classification method, the results showed that the use of stemming and stopwords had no significant effect and did not even increase accuracy with either SVM or the classification method. The use of KNN and SVM is widely used for the classification process in sentiment analysis. The classifier method is considered a potentially better method for classifying data than other classification methods in terms of accuracy. The classification method was also chosen as a classifier because it is easy and fast at predicting data. Therefore, based on the background that has been described, the authors plan to use a classification method with multiple classes, considering accuracy based on previous research, which has good potential, is easy and fast to implement, and is widely used for classification. So, the authors intend to conduct research with the title Performance Analysis of the Use of Stopwords and Stemming in Sentiment Analysis with a Classification Approach. In this study, before the data is processed, pre-processing will be carried out using Stemming and Stopwords as the focus of the research, which will then be processed using the Nave Bayes method.

METHOD

In supporting research, a research design is needed to describe the processes that will be carried out with the aim of achieving maximum results. This design is based on the Cross-Industry Standard Process for Data Mining (CRISP-DM), which includes Data Understanding, Data Preparation, modelling, and Evaluation. The author uses the data collection method, which is carried out by studying literature and finding and studying references and information from various journals, books, and websites as references related to online research information. The dataset used in this study is the Indonesian Twitter Emotion Dataset obtained from Mei Silviana Saputri's github repository with the username meisaputri21. This dataset has a size of 355 KB, along with an abbreviation dictionary. The database consists of two columns, namely labels and tweets. The label column contains love, anger, sadness, joy, and fear as parameters for classification. Meanwhile, the tweet column contains tweets from Twitter users. Based on the review results from these journals, this research has more value in the process that will be carried out later. This research divides it into four models; there are studies that divide it into four models. However, this research uses the Binary Classification Technique using SVM as a Classifier, whereas in this study the authors used the Multi-Class Classification Technique using Nave Bayes as a Classifier. There is also research that uses the Multi-Class Technique, but this research only compares classifiers, or, in other words, does not focus on Stemming and Stopwords.

RESULT AND DISCUSSION

At this dataset input stage, the author inputs the dataset using data that has been retrieved from the GitHub repository under the name Indonesian-Twitter-Emotion-Dataset. This dataset also has two columns, namely labels and tweets, with a total of 4401 rows. The label column contains love, anger, sadness, joy, and fear as parameters for classification. After the data has been successfully entered, it will be prepared before being used for model training. Basically, the existing data has a different letter form. Therefore, at this stage, the data will be converted into lower-case or lower-case letters with the aim of avoiding unwanted errors. The existing dataset still has a lot of noise, so this data cleaning function is to remove the noise that is still present in each sentence. Data cleaning includes the process of removing special tweets (Mentions, Links, Hashtags, numbers, punctuation marks, and excessive spaces). So that the output obtained from the cleaning process becomes cleaner data and reduces noise in sentences. After the data is cleaned, it is processed by separating text into pieces called tokens, which are then analyzed. Words, numbers, symbols, punctuation marks, and other important entities can be considered tokens. In NLP, tokens are defined as words," although tokenizing can also be done on paragraphs and sentences. In this process, the researcher uses the NLTK library by using a function called word tokenize. The use of word tokening is followed by the apply () function, which works on the Pandas Series. The output of this process is in the form of sentences that have been split into words per word with separators in the form of commas ",".

Normalization is used to equate a term that has the same meaning but is written differently; it can be caused by misspellings, shortening of words, or "slang". The initial stage in this process is to create a dictionary with words that have the same meaning, then create a variable as a placeholder for the

words in the dictionary. After that, the looping process is carried out as much as the amount of data in the dictionary so that it will produce output. Then the data will go through a selection process one by one to determine whether the word in the nth term document is found in the abbreviation dictionary. If true, then replace the nth term; if false, then continue the loop. At this stemming stage, the author uses the stemmer function taken from the Literary library to return the word to its basic form. Because the stemmer. Stem () function in the Sastrawi library is slow, here the author uses the Swifter library to speed up the stemming process on Data frames by running tasks in parallel. Processing speed can be twice as fast or even faster if you use Swifter.

Stopwords aims to remove the words from the tokens generated by the previous process. Common words that usually appear in large numbers are considered to have no meaning, so they will be removed. Researchers used the stopword () function taken from the NLTK library to get a list of Indonesian stopwords. The following is a list of Indonesian stopwords generated by the stopword () function. Not only that, the researcher also added several words that the writer considers to have no important meaning and also have a lot of frequency. After the stopwords are collected, the researcher creates a function that will be called in the future if used. At this TF-IDF stage, the data will then go through a weighting process with the intention of making words that appear frequently have a value that tends to be small, while words that occur less frequently will have a value that tends to be large. After going through the pre-processing process, the dataset is then modelled by dividing it into training data and testing data. In terms of dataset division, the authors divide it into four different models. Then the dataset will be divided into 8:2 with the information that 80% of the dataset is used as training data and 20% is used as testing data. But before dividing the dataset, the data must be token-fitted, or, in other words, combining sentences that have been broken word for word into complete sentences. Then, the dataset can be divided for training and testing. In dividing the dataset, there is a train test split () function taken from the sklearn model-selection library. This function has several parameters that can be used: the x parameter is taken from the tweet column that has been joined by tokens, the y parameter is taken from the label column, and the test size parameter functions to share testing data. After that, the data will go through the training stage, which functions to train the data before it is tested. After the dataset model has been completed through the data sharing process and the training process, the data will go through the classification stage, where the data will be tested, or what can be called the predicting process. In this process, the researcher uses the predict () function taken from the sklearn library. The parameter used in the predict () function is the variable x, which has been divided into 20% of the dataset in the previous stage. So that this x variable will be tested to find out the final result. After all models have been trained and tested, the performance evaluation of each model will be based on precision, recall, and f1-score. The reason for using it is because the dataset used by each class is unbalanced. Precision is the ratio of true positive predictions compared to all positive predicted results, and Recall is the ratio of true positive predictions compared to all true positive data. While the f1-score is a harmonic average of precision and recall.

The first step to be taken in outline is to prepare the dataset. Then the authors divide it into four models to determine the performance of each model. After that, the dataset is designed to be divided into data for training and testing. The author has explained the implementation process from start to finish and then discussed the results of the implementation. Based on the results of the previous implementation, it appears that there is an imbalance in the dataset with a total of 881 testing data points; therefore, the performance of a model will be taken based on the f1-score. The value on the f1-score is taken from the results of the confusion matrix for each class. As one example, here the author uses the label anger (0) in calculating TP, FP, and FN. After the f1-score for each class is calculated, the next step is to calculate the average f1-score for all classes. So that the obtained f1-score in the data nstem nstop model is 0.627 and the time needed in the classification process is 385.3 ms.

Based on the results of the previous implementation, it appears that there is an imbalance in the dataset with a total of 881 testing data points; therefore, the performance of a model will be taken based on the f1-score. The value on the f1-score is taken from the results of the confusion matrix for each class. As one example, here the author uses the label anger (0) in calculating TP, FP, and FN. After the f1-score for each class is calculated, the next step is to calculate the average f1-score for all classes. So that the obtained f1-score in the data stop model is 0.633 and the time needed in the classification process is 265.5 ms. Based on the results of the previous implementation, it appears that there is an

imbalance in the dataset with a total of 881 testing data points; therefore, the performance of a model will be taken based on the f1-score. The value on the f1-score is taken from the results of the confusion matrix for each class. After the f1-score for each class is calculated, the next step is to calculate the average f1-score for all classes. So that the obtained f1-score in the data stem model is 0.65 and the time needed in the classification process is 208.9 ms. Based on the results of the previous implementation, it appears that there is an imbalance in the dataset with a total of 881 testing data points; therefore, the performance of a model will be taken based on the f1-score. The value on the f1-score is taken from the results of the confusion matrix for each class. After the f1-score for each class is calculated, the next step is to calculate the average f1-score for all classes. So that the obtained f1-score in the data stem stop model is 0.648 and the time needed in the classification process is 141.7 ms.

This study has implemented sentiment analysis by comparing datasets without stemming and stopwords (data nstem nstop), datasets with stopwords (data stop), datasets with stemming (data_stem), and datasets with stemming and stopwords (data stem stop) with a classification approach. Based on the results of the previous implementation, it can be seen that the model without stemming and stopwords gets an evaluation value of a f1-score of 62.7% with a processing time of 385.3 ms, the model with stopwords gets an evaluation value of a f1-score of 63.3% with a processing time of 265.5 ms, the model with stemming gets an evaluation value of a f1-score of 65% with a processing time of 208.9 ms, and the model with stemming and stopwords gets an evaluation value of 64.8% with a processing time of 141.7 ms. So, this research is inversely proportional to research conducted by Hidayatullah (2015), who used Nave Bayes and SVM with Binary classification. There is a decrease in accuracy when the model uses stemming and stopwords. Whereas in another study conducted by Pradana & Hayaty (2019), SVM was used with Binary classification. The increase in performance is only found in the No Stop (data stem) model by 0.04%, and there is a decrease in performance in the other models.

CONCLUSION

The results obtained from the previous implementation show that the use of Stopwords and Stemming pre-processing can improve performance in sentiment analysis. With the results of the model without stemming and stopwords getting an f1-score of 62.7%, the model with stopwords getting an f1-score of 63.3%, the model with stemming getting an f1-score of 65%, and the models with stemming and stopwords getting an f1-score of 64.8%. The use of Stopwords and Stemming also improves performance in the classification process. With data nstem nstop processing time of 385.3 ms, data stop processing time of 265.5 ms, data stem with processing time of 208.9 ms, and data stem stop with processing time of 141.7 ms. Using a balanced dataset to get better results. Using more datasets so that the performance of each model in processing time is more visible. Using other techniques such as binary and multi-Label to determine the effectiveness of pre-processing. The selection of pre-processing techniques is important for processing text data to improve sentiment classification performance.

REFERENCES

Akella, J. O., & Akella, L. N. Y. (2018). Sentiment Analysis Using Naïve Bayes Algorithm: With Case Study. *Proceedings of the 3rd International Conference on Inventive Computation Technologies, ICICT 2018*. https://doi.org/10.1109/ICICT43934.2018.9034394

Alam, S., & Yao, N. (2019). The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. *Computational and Mathematical Organization Theory*, 25(3), 319–335. https://doi.org/10.1007/s10588-018-9266-8

Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment Analysis of Review Datasets Using Naïve Bayes‘ and K-NN Classifier. *International Journal of Information Engineering and Electronic Business*, 8(4), 54–62. https://doi.org/10.5815/ijieeb.2016.04.07

Fitri, V. A., Andreswari, R., & Hasibuan, M. A. (2019). Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm. *Procedia Computer Science*, 161, 765–772. https://doi.org/10.1016/j.procs.2019.11.181

Hidayatullah, A. F. (2015). *The Influence of Stemming on Indonesian Tweet Sentiment Analysis*. In Computer Science and Informatics. http://www.website.com

Pradana, A. W., & Hayaty, M. (2019). The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts. *Kinetic: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 4(3), 375–380. https://doi.org/10.22219/kinetik.v4i4.912

Saputri, M. S., Mahendra, R., & Adriani, M. (2018). *Emotion Classification on Indonesian Twitter Dataset*. IEEE.

Simon Kemp. (2022, February 15). DIGITAL 2022: INDONESIA. Datareportal.Com. https://datareportal.com/reports/digital-2022-indonesia

Sudarsa, D., Kumar.P, S., & Jagajeevan Rao, L. (2018). Sentiment Analysis for Social Networks Using Machine Learning Techniques. *International Journal of Engineering & Technology*, 7(2.32), 473. https://doi.org/10.14419/ijet.v7i2.32.16271