Whoosh User Sentiment Analysis on Social Media Using Word2Vec and the Best Naïve Bayes Probability Model

Muhammad Dinan Islamanda 1,*, Yuliant Sibaroni2)
1,2) School of Computing, Telkom University, Indonesia
1) dinanislamanda@student.telkomuniversity.ac.id, 2) yuliant@telkomuniversity.ac.id

Submitted : Jun 10, 2024 | Accepted : Jun 28, 2024 | Published : Jul 1, 2024

Abstract: By using the Twitter microblogging feature, users can post short tweets with limited characters that express their thoughts and opinions regarding a matter. The newest transportation in Indonesia, a high-speed train namely Whoosh is one of the things that Twitter users responded to. This latest transportation has led to the emergence of opinions from the Indonesian people which are shared publicly in various media, one of which is social media. Therefore, to make it easier for business people or companies to understand public opinion regarding service improvements in the future, sentiment analysis on social media is needed to determine user opinions regarding high-speed train transportation. In this research, sentiment analysis of high-speed train users will be carried out on social media Twitter using Word2Vec and Naïve Bayes as classification methods. In this research, a comparison of Naïve Bayes models will also be carried out to find out the best Naïve Bayes method opportunity model. Simultaneously, the Word2vec feature extraction method was chosen because Word2Vec can be used to improve model performance and increase the accuracy of sentiment classification. This research found that the Word2Vec Skip-Gram model outperformed the Word2Vec CBOW model. The best model obtained was the use of the Gaussian Naïve Bayes and Word2Vec Skip-Gram models with an accuracy score of 77.18%, precision 70.35%, recall 76.09%, and f1-score 73.10%.

Keywords: high-speed train, naïve bayes, sentiment analysis, twitter, word2vec

INTRODUCTION

By using the Twitter microblogging feature, users can post short tweets with limited characters that express their thoughts and opinions regarding a matter (Malik et al., 2019). The newest transportation in Indonesia, a high-speed train namely Whoosh is one of the things that Twitter users responded to. Whoosh travels from Jakarta to Bandung and vice versa. High-speed trains have technological advances compared to conventional trains which are able to shorten travel time, especially for medium distance travel (Lestari et al., 2018). However, this transportation raises many opinions from the public. According to Dhea, a resident of Bandung, high-speed train transportation must have affordable prices and facilities commensurate with those offered (Awla Rajul, 2023). Affordable and efficient mass transportation is very important for the survival of society (Gershon, 2005). Therefore, to make it easier for business people or companies to understand public opinion regarding service improvements in the future, sentiment analysis on social media is needed to determine user opinions regarding high-speed train transportation.
Many studies discussing sentiment analysis use classification methods and various feature extraction methods. Word2Vec can produce a high f1-score value in Indonesian news categorization using the Naïve Bayes algorithm with an average f1-score value of 90% (Santoso et al., 2018). In e-wallet sentiment analysis research using Naïve Bayes and N-Gram, the use of 2 grams made the data tested better than before. Obtained 94.90% accuracy with the Naïve Bayes algorithm for OVO and 94.70% accuracy for DANA (Krisitianyanti et al., 2020). In sentiment analysis research using Random Forest and Word2Vec, the accuracy results using Word2Vec were 86.87%, higher than using Bag-of-Word and TF-IDF with accuracy of 83.49% and 84.49% (Hitesh et al., 2019).

Applying Naïve Bayes machine learning classification to sentiment analysis obtained quite good accuracy compared to Random Forest and SVM. The proposed algorithm performs better for higher volume tweets (Shamantha et al., 2019). Naïve Bayes provides accuracy results of 80.9% compared to KNN 75.58% and SVM 63.88% in sentiment analysis for Twitter analysis of the 2019 Presidential Candidates of the Republic of Indonesia (Wongkar & Angdresey, 2019).

Naïve Bayes has several models and each model has its characteristics. Multinomial and Bernoulli Naïve Bayes classifiers are suitable for discrete data. The difference is, when classifying test documents, Multinomial Naïve Bayes works by tracking several events, while Bernoulli Naïve Bayes uses binary event information, ignoring the number of events (Schütze et al., 2008). Complement Naïve Bayes is suitable for imbalanced datasets and this classifier improves classification accuracy by using data from all classes except the class on which it is focused (Sato et al., 2011). When dealing with continuous data, it is commonly assumed that each class’s continuous values are distributed using a Gaussian distribution. For instance, the continuous attribute x is present in the training set. The mean and variance x are computed for each class after the data has been divided into segments based on class (Gayathri & Sumathi, 2016).

From previous research, Naïve Bayes is a classification method that is quite good at classifying sentiment among other methods and Word2Vec can improve performance results in classification algorithms. In this research, sentiment analysis of high-speed train users will be carried out on social media Twitter using Word2Vec and Naïve Bayes as classification methods. In this research, a comparison of Naïve Bayes models will also be carried out to find out the best Naïve Bayes method opportunity model for analyzing the sentiment of high-speed train users on Twitter social media using Word2vec feature extraction. Simultaneously, the Word2vec feature extraction method was chosen because Word2Vec can be used to improve model performance and increase the accuracy of sentiment classification.

**LITERATURE REVIEW**

The research topic regarding sentiment analysis using machine learning classifiers has been carried out previously. Sentiment analysis research on tweets data using Naïve Bayes, SVM, and Random Forest classification. The application of Naïve Bayes and SVM machine learning classification to sentiment analysis obtained quite good accuracy compared to Random Forest. Increased accuracy also occurs with the use of extraction features (Shamantha et al., 2019).

Sentiment analysis research on transportation has been carried out on KAI (Indonesian Railway) using SVM and TF-IDF. Tweets containing user opinions on Twitter regarding the services provided by KAI were used in this research. The classification results using the OAA Multiclass SVM method in this study used different weighting features. The unigram TF-IDF model in conjunction with the OAA Multiclass SVM with gamma 0.7 yields the highest accuracy of 80.59 (Fitriana & Sibaroni, 2020).

Additionally, a number of studies employing the Naïve Bayes method have been conducted. Sentiment analysis research on KFC product reviews on Twitter and YouTube was carried out using Naïve Bayes. The application of the Naïve Bayes algorithm obtained quite good accuracy of 85.48% for data before endorsement and 86.48% for data after endorsement (Ramdhani et al., 2018). Sentiment analysis research on customer satisfaction with Gojek and Grab Indonesia using Naïve Bayes. The data used is the result of collecting tweets based on keywords about Gojek and Grab in Indonesian Language. Obtained accuracy of 72.33% with 73.95% and 73.24% of recall and precision on average (Sari et al., 2019).
Several studies using the Word2Vec method have also been carried out. Sentiment analysis research on hotel reviews was carried out using the LSTM and Word2Vec methods. There is a comparison of the Word2Vec architecture used in this research. It was found that the best scheme achieved an average accuracy of 85.96%, with Skip-Gram used as the Word2Vec architecture, and with the vector dimension value set to 300 (Muhammad et al., 2021). Comparison of Multinomial and Gaussian Naïve Bayes was carried out in sentiment analysis research on movie reviews using Word2Vec. There is a comparison of testing the number of dimensions in Word2Vec with a total of 300 dimensions being the best model. The Multinomial Naïve Bayes model provides the highest accuracy of 72.23% compared to the Gaussian Naïve Bayes model with an accuracy of 68.72% (Rizal et al., 2022).

From the results of previous studies, the Naïve Bayes classification method is a good method for classifying sentiment among other classification methods such as Random Forest and SVM. The Word2Vec method can also help improve the accuracy of sentiment classification. This research will analyze the sentiment of high-speed train users using Word2Vec and Naïve Bayes. In this research, a comparison of Naïve Bayes models will also be carried out to find out the best Naïve Bayes method opportunity model.

**METHOD**

The purpose of this research is to create a sentiment classification system for user feedback on high-speed train transportation, comparing four Naïve Bayes models and the Word2Vec architecture. Figure 1 shows the system design flowchart that is shown below.

**Dataset**

Crawling methods were utilized to obtain the dataset from Twitter for this research. The data that has been crawled is tweet data related to high-speed trains such as 'kereta cepat', 'whoosh', 'KCIC', and 'kereta cepat jakarta bandung' and saved in a CSV file. The dataset used is 12,923 data containing user opinion tweets regarding high-speed train transportation in the Indonesian language.

**Preprocessing**

The data resulting from data crawling will be pre-processed so that the data becomes clean data and is easier to carry out sentiment classification. This stage consists of several processes. Table 1 shows several stages in data preprocessing. Stopword Removal and Tokenization stages using the NLTK library. The indoNLP library is used in the Replace Slang Words, Remove Elongation, and Replace Emoji stages, while the Stemming stage uses the Sastrawi library.
Table 1. Example of Preprocessing Text Result

<table>
<thead>
<tr>
<th>Stages</th>
<th>Explanation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Text</td>
<td>Example of user tweets</td>
<td>Kemarin sudah mencoba kereta cepat, kagumm bgt. Pemandangan yg keren dan indah 😊. Perjalanan ga berasa lama tiba tiba udh sampe #whoosh</td>
</tr>
<tr>
<td>After Case Folding</td>
<td>Change all letters to lowercase</td>
<td>kemarin sudah mencoba kereta cepat, kagumm bgt. pemandangan yg keren dan indah 😊. perjalanan ga berasa lama tiba tiba udh sampe #whoosh</td>
</tr>
<tr>
<td>After Stopword Removal</td>
<td>Removal of all stopwords</td>
<td>kemarin mencoba kereta cepat, kagumm bgt. pemandangan yg keren indah 😊. perjalanan ga berasa udh sampe #whoosh</td>
</tr>
<tr>
<td>After Replace Slang Words</td>
<td>Change of abbreviated words</td>
<td>kemarin mencoba kereta cepat, kagumm banget. pemandangan yang keren indah 😊. perjalanan enggak berasa sudah sampai #whoosh</td>
</tr>
<tr>
<td>After Remove Elongation</td>
<td>Changing words that have elongation</td>
<td>kemarin mencoba kereta cepat, kagum banget. pemandangan yang keren indah 😊. perjalanan enggak berasa sudah sampai #whoosh</td>
</tr>
<tr>
<td>After Emoji to Word</td>
<td>Turns an emoji into a word</td>
<td>kemarin mencoba kereta cepat, kagum banget. pemandangan yang keren indah 😊!wajah_gembira!. perjalanan enggak berasa sudah sampai #whoosh</td>
</tr>
<tr>
<td>After Cleansing</td>
<td>Removal of URLs, numbers, symbols, usernames, and punctuation</td>
<td>kemarin mencoba kereta cepat kagum banget pemandangan yang keren indah wajah gembira perjalanan enggak berasa sudah sampai</td>
</tr>
<tr>
<td>After Stemming</td>
<td>Convert a word to its base word</td>
<td>kemarin coba kereta cepat kagum banget pandang yang keren indah wajah gembira jalan enggak asa sudah sampai</td>
</tr>
<tr>
<td>After Tokenization</td>
<td>Breaking tweet text into tokens</td>
<td>['kemarin', 'coba', 'kerja', 'cepa', 'kagum', 'banget', 'pandang', 'yang', 'keren', 'indah', 'wajah', 'gembira', 'jalan', 'enggak', 'asa', 'sudah', 'sampai']</td>
</tr>
</tbody>
</table>

**Word2Vec Word Embedding**

In Natural Language Processing, word embedding is a crucial step that involves taking unlabeled text data and extracting syntactic and semantic features from it. (Li & Yang, 2018). Word2Vec word embedding is used in this research and is carried out after the data preprocessing process. Word2Vec converts words into vectors according to several parameters, including vector dimensions and window size. Similar words are typically grouped in the same block and have similar vector values (Jatnika et al., 2019). Continuous Bag of Words (CBOW) and Skip-Gram are two architectures in Word2Vec. In contrast to Skip-Gram architecture which predicts words around the current given word, CBOW architecture predicts current words based on context. The following figure is the Skip-Gram and CBOW architecture.
**Split Data**

Following Word2Vec word embedding, the data will be divided into training and testing groups. 90% of the training data and 10% of the testing data make up the 90:10 division ratio. Data division is carried out randomly.

**Cross Validation**

Rerunning the entire procedure \( k \) times using distinct training and testing datasets is the goal of the \( k \)-fold cross-validation technique. (Saud et al., 2020). The optimal model is then chosen by obtaining the minimum error using a variety of statistical tools for error estimation. This research uses the formula \( k = 10 \) fold to calculate \( k \) in \( k \)-folds. The following figure is the cross validation illustration.
Naive Bayes Model Learning

Naive Bayes classification is a supervised learning classification algorithm in machine learning. This classification is rooted in the Bayesian theorem by Thomas Bayes in determining probability. These probabilistic models make it possible to capture model uncertainty in principle by specifying probabilities (Parveen & Pandey, 2016). From a classification point of view, the main goal of this classification is to determine the optimal mapping between new data and a set of classifications in a certain problem domain (Yang, 2018). The following is the equation formula for the Naïve Bayes method.

\[
P(y_j|x_i) = \frac{P(x_i|y_j)P(y_j)}{P(x_i)}
\]  

Where, \( P(y_j|x_i) \) is the probability of category j when word i appears, \( P(y_j) \) is the probability of category j appearing, \( P(x_i|y_j) \) is the probability that word i is included in category j, \( P(x_i) \) is the probability of appearing a word.

The odds in the formula above can be calculated using several assumptions regarding the probability distribution, namely:

a. Multinomial Naïve Bayes

This Naive Bayes model uses a multinomial model to represent the word distribution in a document. Words are treated as a sequence in documents, and it's thought that every word position is generated separately from the others (Rennie et al., 2003). The Multinomial Naïve Bayes formula is as follows.

\[
P(x|C_k) = \frac{(\sum_{i=1}^{n} x_i)!}{\prod_{i=1}^{n} x_i!} \prod_{i=1}^{n} P_{kl}^{x_i} (1 - P_{kl})^{(1-x_i)}
\]

Where, \( P_{kl} \) := \( P(x_i|C_k) \).

b. Gaussian Naïve Bayes

When dealing with continuous data, it is commonly assumed that each class’s continuous values are distributed using a Gaussian distribution. For instance, the continuous attribute x is present in the training set. The mean and variance \( \mu \) and \( \sigma^2 \) are computed for each class after the data has been divided into segments based on class (Gayathri & Sumathi, 2016). Therefore, the following formula can be used to estimate the probability of a continuous dataset.

\[
P(x_i|y) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)}
\]

c. Bernoulli Naïve Bayes

The Bernoulli Naive Bayes model is very suitable for performing binary classification (McCallum & Nigam, 1998). Discrete data is also suitable using Bernoulli Naïve Bayes as with Multinomial Naive Bayes. However, the difference is that Multinomial Naive Bayes works with the number of events, the Bernoulli model uses binary event information, ignoring the number of events (Schütze et al., 2008). The following is the Bernoulli Naïve Bayes formula.

\[
P(x|C_k) = \prod_{i=1}^{n} P_{kl}^{x_i} (1 - P_{kl})^{(1-x_i)}
\]

Where, \( P_{kl} \) is the likelihood that the term \( x_i \) will be generated by class \( C_k \).

d. Complement Naïve Bayes

Complement Naïve Bayes classifier is a modification of the Multinomial Naïve Bayes classifier and is suitable for imbalanced datasets. This classifier improves classification accuracy by using data from all classes except the class on which it is focused (Sato et al., 2011). Complement Naïve Bayes estimation will be more effective because each class uses a more even amount of
training data per class, thereby reducing bias in weight estimation (Rennie et al., 2003). The Complement Naïve Bayes formula is as follows.

\[
 l_{CNB}(d) = \arg \max_c \left[ \log p(\vec{\theta}_c) - \Sigma_i f_i \log \frac{N_{ei}}{N_{e}} + \alpha_i \right] \tag{5}
\]

Where \( N_{ei} \) is the quantity of instances of the word \( i \) in documents from classes other than \( c \), \( N_{e} \) is the total count of words that appear in classes other than \( c \), and \( \alpha_i \) and \( \alpha \) are smoothing parameters.

**Evaluation**

Using a confusion matrix, the constructed model will be assessed to ascertain its performance. Confusion Matrix is a machine learning visual assessment tool, the actual class results are represented by the rows, and the predicted class results are represented by the columns (Xu et al., 2020). The following are some of the most given sizes.

Accuracy, this metric is used to calculate the proportion of all correctly predicted cases (Deng et al., 2016).

\[
 Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{6}
\]

Precision, this measurement is used to determine the level of accuracy of predictions for certain classes (Deng et al., 2016).

\[
 Precision = \frac{TP}{TP+FP} \tag{7}
\]

Recall, this measurement is used to identify examples of a particular class from a data set (Deng et al., 2016).

\[
 Recall = \frac{TP}{TP+FN} \tag{8}
\]

F1-Score, a measurement of the harmonic mean between precision and recall (Deng et al., 2016).

\[
 F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}
\]

There are 4 values in the confusion matrix, namely True Positive (TP) the positive (Actual) number which is categorized correctly as positive, True Negative (TN) the negative (Actual) number which is correctly categorized as negative, False Positive (FP) the negative (actual) number which are incorrectly categorized as positive, and False Negative (FN) the number of (actual) positives which are incorrectly categorized as negative (Zeng, 2020). These values are defined based on the following confusion matrix in Table 2.

<table>
<thead>
<tr>
<th>Table 2. Confusion Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actual Class</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Predicted Class</strong></td>
</tr>
<tr>
<td>Positive</td>
</tr>
<tr>
<td>Negative</td>
</tr>
</tbody>
</table>
RESULT

This research uses a dataset of 12,923 with a data composition of 7,651 negative sentiments and 5,272 positive sentiments. The preprocessed data is followed by feature extraction using Word2Vec to implement the processed data. In this study, an assessment was finished using the Naïve Bayes classification and Word2Vec Word Embedding by comparing four Naïve Bayes probability models (Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Gaussian Naïve Bayes, and Complement Naïve Bayes) and also the Word2Vec architectural model (CBOW and Skip-Gram) after the modeling stage to find the most optimal accuracy results. Evaluations of recall, accuracy, precision, and f1-score are included in the test results, are presented in Table 3 and Table 4.

Table 3. CBOW Result

<table>
<thead>
<tr>
<th>Naïve Bayes Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multinomial Naïve Bayes</td>
<td>73.70%</td>
<td>78.76%</td>
<td>48.57%</td>
<td>60.09%</td>
</tr>
<tr>
<td>Complement Naïve Bayes</td>
<td>74.24%</td>
<td>67.13%</td>
<td>72.10%</td>
<td>69.53%</td>
</tr>
<tr>
<td>Bernoulli Naïve Bayes</td>
<td>74.24%</td>
<td>67.25%</td>
<td>71.72%</td>
<td>69.42%</td>
</tr>
<tr>
<td>Gaussian Naïve Bayes</td>
<td>75.87%</td>
<td>70.09%</td>
<td>71.15%</td>
<td>70.62%</td>
</tr>
</tbody>
</table>

Table 4. Skip-Gram Result

<table>
<thead>
<tr>
<th>Naïve Bayes Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multinomial Naïve Bayes</td>
<td>75.25%</td>
<td>82.85%</td>
<td>49.52%</td>
<td>61.99%</td>
</tr>
<tr>
<td>Complement Naïve Bayes</td>
<td>76.10%</td>
<td>68.10%</td>
<td>77.79%</td>
<td>72.63%</td>
</tr>
<tr>
<td>Bernoulli Naïve Bayes</td>
<td>75.94%</td>
<td>67.70%</td>
<td>78.36%</td>
<td>72.64%</td>
</tr>
<tr>
<td>Gaussian Naïve Bayes</td>
<td>77.18%</td>
<td>70.35%</td>
<td>76.09%</td>
<td>73.10%</td>
</tr>
</tbody>
</table>

Based on the two tables above, it was found that using the Word2Vec Skip-Gram and Gaussian Naïve Bayes models produced the best results with an accuracy of 77.18%.

DISCUSSIONS

The Gaussian Naïve Bayes model and the Word2Vec Skip-Gram model obtained the most optimal results with accuracy of 77.18%, precision of 70.35%, recall of 76.09%, and f1-score of 73.10% with the confusion matrix, as can be seen in Table 5. These results are based on the evaluation results using the four Naïve Bayes models and the Word2Vec architecture model (Skip-Gram and CBOW).

Table 5. Confusion Matrix Result

<table>
<thead>
<tr>
<th>Predicted Class</th>
<th>Actual Class</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>401</td>
<td>169</td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>126</td>
<td>597</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 shows that, in comparison to Word2Vec CBOW and other Naïve Bayes models, Gaussian Naïve Bayes and Word2Vec Skip-Gram yield more accurate results for high-speed train user sentiment. These are the values of the confusion matrix that resulted.

1. True Positive (TP) totaled 401, wherein 401 data that the model both predicted and actually classified as positive.
2. True Negative (TN) totaled 597, wherein 597 data that the model both predicted and actually classified as negative.
3. The model classified 169 data that it had incorrectly predicted as positive but were actually negative, yielding a total of 169 False Positives (FP).
4. The model classified 126 data that it had incorrectly predicted as negative but were actually positive, yielding a total of 126 False Negatives (FP).
CONCLUSION

In research on high-speed train user sentiment analysis using the Naïve Bayes algorithm and Word2Vec word embedding, four variants of the Naïve Bayes model were compared in each test. Testing also compares the performance of Skip-Gram and CBOW Word2Vec models. The use of Naïve Bayes and Word2Vec models influences performance results. The test results show that the Gaussian Naïve Bayes model obtains better performance results than the other three Naïve Bayes models.

However, the use of the Word2Vec Skip-Gram model provides better performance than the Word2Vec CBOW model. The Skip-Gram model provides 2 percent superior accuracy and higher recall results compared to the CBOW model.

In this research, the best combination of Naïve Bayes and Word2Vec methods was found, namely the use of the Gaussian Naïve Bayes and Word2Vec Skip-Gram models which obtained accuracy results of more than 77%. This combination also gets precision, recall and F1-score results of 70.35%, 76.69%, and 73.10%.

Suggestions for further research, to produce better results could be to use a more complex classification model. Other word embedding methods can also be used to compare the Word2Vec method to find out which method is better.

REFERENCES


*name of corresponding author

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.