

Effects of using wordnet and spelling checker on classification methods in sentiment analysis for datasets using Bahasa

Rizky Andika¹, Suharjito²

¹Computer Science Department, Binus Graduate Program, Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

²Computer Science Department, Binus Online Learning, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Aug 17, 2021

Revised Dec 6, 2021

Accepted Jan 11, 2022

Keywords:

Bahasa

Sentiment analysis slang words

wordnet

Spelling checker

ABSTRACT

Sentiment analysis was a system for recognizing and extracting opinions in documents. There were two weaknesses in sentiment analysis. The first weakness was preprocessing in sentiment analysis can't recognize slang words so that important words that should have been recognized became unrecognizable. The Second was the feature extraction process in sentiment analysis only recognized words based on the form of the word but can't recognize the similar word. In this paper, we proposed spelling checker and wordnet to fix these weaknesses. We also used k-nearest neighbor (KNN), Naïve Bayes, and decision tree as methods for check classify the text. The purpose of this research was first to know the effects of used Wordnet and spelling checkers in sentiment analysis and second was to improve the data processing process in the existing sentiment analysis. The dataset that we used in the research was a list of tweets in Bahasa. The results showed wordnet and spelling checker make a decrease in the valued of false positives, false negatives, and true negatives in the calculation of the confusion matrix. It can increase the accuracy of the K-NN from 43% to 72%, Naïve Bayes from 41% to 71% and decision tree from 47% to 75%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Rizky Andika

Computer Science Department, Binus Graduate Program, Master of Computer Science

Bina Nusantara University

Jakarta, Indonesia

Email: rizky.andika@binus.ac.id

1. INTRODUCTION

Sentiment analysis labels a text or some texts as expressing either a positive or negative opinion in a document and can be considered the challenge of building a classification from the text [1]. Because sentiment analysis consists of many opinion datasets, the dataset of sentiment analysis consists of non-standard or unstructured data, thus making the text classification process in the data set inaccurate. There are two weaknesses in sentiment analysis. The first weakness is slang words can make sentiment analysis can't process raw data. Slang is a type of language of non-standard words and phrases [2]. Slang words can interfere with the algorithm process in sentiment analysis to process datasets in social media texts [3]. We provide an illustrative example of the effect of slang words in sentiment analysis, an example is a tweet that contains slang words like "at de moment he cnt just put me in da better zone thoughhhh happy bday mic, ur a legend". The tweet contains some slang words, and you will recognize some terms which doesn't belong to decent vocabulary, so that the tweet is not suitable to be used as a dataset for classifying texts in analysis, because it must be used as a standard first for each word. Second is not every list of words in the dataset is easy to classify by counting words with the dictionary. For example, two words "beautiful" and "pretty" in

the sentence “She's beautiful, but her sister's not pretty”. If humans read the sentence, the two words in the sentence have two different meanings, because the word “pretty” is preceded by the word “not”, human readers would recognize these words. But algorithms in preprocessing that count words out of context would be fooled because the algorithm only recognizes words based on the existing dictionary, not the meaning of the whole sentence, so the algorithm only assumes that the two words have the same meaning [4].

In this research, provides innovation to correct the shortcomings of the sentiment analysis process that cannot process slang words and recognize words, synonyms and their opposites. Solutions for these weaknesses are to use the spelling checker to recognize slang words so that spelling checker can solve the non-word error problem [5] and wordnet to recognize words, synonyms and their opposites [6]. Wordnet is one of some dictionaries that contain a lexical database for sentiment analysis for English and it is manually built by a group of lexicographers at Princeton University [7]. Wordnet can calculate and expand the meaning of some words that have the same meaning as synonyms and antonyms [8]. Spelling checker is a process of detecting spelling errors or slang words in word writing using an existing language dictionary [9]. The spelling checker is very helpful in this research because can help recognize spelling errors or slang words. Spelling errors or slang words are words that are not in the language dictionary [10]. We propose k-nearest neighbor (KNN), Naive Bayes, and decision tree classification methods for evaluating it. The purpose of this research is to evaluate the effect of using wordnet and spelling checker with dataset Bahasa in sentiment analysis. This research contributes to correct the weaknesses of sentiment analysis that are currently unable to process documents containing slang words and similar words with wordnet and spelling checker. It is important because slang words can invalidate the calculation of the classification process in sentiment analysis. So that the calculations in the sentiment analysis process are incorrect. Our research is different from previous research because we developed sentiment analysis with spelling checker and wordnet. The effect of using spelling checker and wordnet is that it can reduce the value of false negatives, false positives and true negatives, thereby increasing the value of the accuracy of sentiment analysis.

2. RELATED WORK

In this paper, we have taken several reference journals and articles as references in our research. Spelling checker is very helpful in the process of eliminating slang words in the dataset in the process of sentiment analysis [5], [9]. Ababneh [11] explain that Naive Bayes is better than decision tree and k-NN. Soleh and Purwarianti [12] explains that slang words can make the calculation of the confusion matrix in sentiment analysis inaccurate. Dey *et al.* [13] that explain Naive Bayes is better than other classification methods for hotel review case studies, Suhariyanto *et al.* [14] that explain support-vector machine (SVM) is better than other classification methods for detection fake movie [15], decision tree is better than Naive Bayes and k-NN [16], k-NN is better than Naive Bayes [17], k-nearest neighbor classifier is transparent, consistent, straightforward, simple to understand, high tendency to possess desirable qualities and easy to implement than most other machine learning techniques [18]. The process of extracting data sets on social media is more suitable for using unsupervised machine learning than supervised learning [19], subversion (SVN) has the lowest accuracy than k-NN and Naive Bayes [20], sentiment analysis is widely applied in fields related to opinions describing satisfaction or dissatisfaction such as case studies of tourist reviewers, film reviewers and the like [21], [22]. It is these various case studies that produce many conclusions, because each of these cases also uses various datasets, both training data and testing data, so that they have the potential to produce various kinds of conclusions. The interesting thing about the related work that we studied is that there are studies that say that slang or non-standard words can interfere with and falsify natural language processing tasks carried out on social media texts and sentiment analysis algorithm is currently not able to detect the similarity of words in the training and testing data so that the calculation process in sentiment analysis becomes less valid. Several studies have mentioned that spelling checkers can improve slang words into standard words [23], [24]. It is very good because by improving slang words in training data and testing data in sentiment analysis, the calculation process becomes better and more valid [25]. Wordnet is a language dictionary to find out the similarities, synonyms and antonyms of a word [26]. Wordnet can be applied in natural language processing (NLP) to search for synonyms so that sentiment analysis can identify synonyms and opposites in the dataset. The similarity of our research with research is that we want to learn about sentiment analysis. The reason we study sentiment analysis is because we believe that the methods in sentiment analysis can be further improved in terms of accuracy. The difference in sentiment analysis that we studied compared to sentiment analysis that has been done by studies is that we want to use spelling checker and wordnet before the preprocessing process in sentiment analysis, because based on the research we have done, spelling checker and wordnet can reduce false negative, false positive and true negative values in the confusion matrix in sentiment analysis. This makes the accuracy, precision, recall and F1 score values increase compared to sentiment analysis that does not use spelling checker and wordnet.

3. METHOD

The research methodology starts from building a framework in the process of solving problems. The framework of this research consists of two processes. The first process is spelling checker and wordnet process. The second process is text classification.

3.1. Slang word

Slang words is a non-standard language for easier communication and instantly in social group, so that when the researcher wants to conduct sentiment analysis, the dataset that contains slang words will interfere with the text classification process [27]. A dataset containing slang words will interfere with the text classification process, especially in the calculation process in the confusion matrix, because in the confusion matrix formula to find the precision, recall, F1 score and accuracy values, it requires false positive, false negative and true negative values. If the slang word is still in the dataset, it can cause the accuracy, precision, recall and F1 score values to be invalid, because the true positive (TP), true negative (TN), false negative (FN), and false positive (FP) values do not match the data in the dataset [28].

3.2. Wordnet process

Semantic similarity measure is a problem that often occurs in artificial intelligence. It has been widely used in NLP, information retrieval, word sense disambiguation, recommended system, and information extraction. Wordnet can solve this problem. It is a lexical database of English. In wordnet nouns, verbs, adverbs and adjectives are organized by a variety of semantic relations into synonym sets, which represent one concept. Semantic relations used by wordnet are antonyms, synonyms and hyponyms [29]. The wordnet derived knowledge base makes semantic knowledge available which can be used in overcoming many problems associated with the richness of natural language. A measure of semantic similarity is very suitable to be used as an alternative to pattern matching in the comparison process in sentiment analysis [30].

3.3. Spelling checker and wordnet in preprocessing

In this paper, the spelling checker and word expansion using wordnet were used before the pre-processing and confusion matrix processes. The result of this process is a dataset that does not contain slang words and a dataset where every word has a synonym and antonym, so that the words from the dataset and the training data can be identified in the confusion matrix. Details of the process can be seen in the Figure 1.

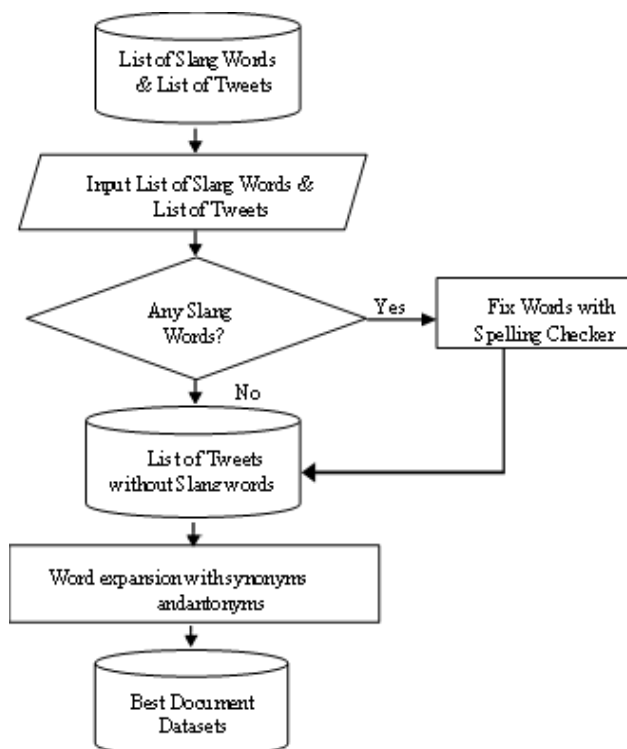


Figure 1. Flow chart spelling checker process and wordnet in preprocessing

Figure 1 is presented, a flow process of spelling checker and wordnet. The following steps are taken: i) The first process is to input list slang words and list tweets. Example, we have a list of tweets in Bahasa “makanan ini endess sekali”. Because the tweet contains one slang word “endess” the algorithm will replace the word from “endess” to be “enak”; ii) the second process is the algorithm will search slang words "endess" in the list of slang words and replace it be “enak”. The result from the spelling checker process is list tweet without slang words “makanan ini enak sekali”; iii) the third process is the expansion of each word using wordnet to search for every synonym and antonym of each word in the tweet; and iv) After expanding the word using wordnet, you will get the best documents containing two datasets. The first dataset is a list of synonyms and antonyms of each word in each tweet. These synonyms and antonyms will be grouped into positive and negative words. The second is a list of tweets that do not contain slang words that will be tested for the text classification process in sentiment analysis.

3.4. Text classification

Text classification is the process of classifying a large text document to obtain important information in the document [31]. Text classification can help to evaluate a case study related to the text such as the classification process for film reviews, hotel reviews and news reviews. Classification text consists of several processes such as preprocessing, feature extraction and classification [32]. Details of the text classification process can be seen in Figure 2.

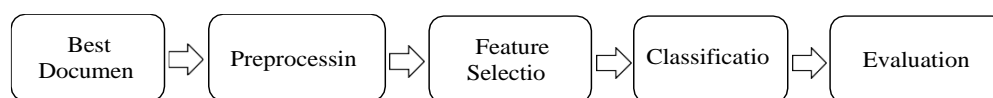


Figure 2. Text classification process phases

Figure 2 is presented, a flow process of text classification such as preprocessing, feature selection, indexing and classification. Preprocessing will contain several processes such as case folding, tokenizing, filtering and stemming [33]. Feature selection in this research using the term's frequency and inverse document frequency (TF-IDF) algorithm [34]. Feature extraction in this research using the TF-IDF algorithm.

The result of the feature extraction of this research is TF-IDF weighting results for each tweets along with positive and negative label categories. Classification for this research is to determine the matrix confusion to find the true positive, negative true, false positive and falsenegative values that will be used in the calculation recall, precision, accuracy and F1 score [35]. Several text classification algorithms from machine learning and data mining communities exist such as: Naïve Bayes, decision trees (DTs), and k-nearest neighbors (KNN).

3.5. Data preprocessing

Each dataset, both training data and testing data, will go through a preprocessing process consisting of several processes. Data preprocessing is the process of preparing the dataset that has been taken by eliminating the unused part so that only important data is produced which will later be used as a feature that will be used in the process of making sentiment analysis models [36]. Data preprocessing consists of several processes such as:

- a) Case folding, is a process of changing all the letters in a text into all lowercase letters.
- b) Tokenizing/parsing, the stage of cutting the input string based on each word that makes it up. In addition, spaces are used to separate the words.
- c) Filtering is the stage of taking words that are considered important from the results of tokenizing. filtering can use a stoplist algorithm (discarding less important words) or wordlist (save important words).
- d) Stemming is the process of changing a word into its standard form.

3.6. Feature extraction

The TF-IDF algorithm is a common method for feature extraction entries in the text classification process and it is simple and highly efficient algorithm for text representation [37]. TF-IDF is a statistical method for assessing the importance of a word for one document or one of some corpuses. TF-IDF is an algorithm that is used to calculate the text weight value based on the frequency and frequency of the reverse

document of a word or phrase which is a feature item TF/IDF has a formula based on the weighting value. The formula is can be seen in (1)-(3).

$$TF - IDF = TF \times \log IDF \quad (1)$$

$$IDF = TF \times \log IDF \quad (2)$$

$$W = TF \times IDF \quad (3)$$

The formula is the formula of the TF-IDF algorithm, where term's frequency (TF) represents the frequency of feature items, inverse document frequency (IDF) represents the frequency of inverted documents from feature items. When a word or phrase appears in all documents, the IDF value will be 1, and 0 makes the entire result zero after taking the logarithm. The TF-IDF term weighting is a combination of the standard TF calculation formula with the IDF formula by multiplying the TF value with the IDF value. weight (W) is the term weight (TF) of the document.

3.7. Classification

Based on the related work that we have studied, in this paper we will use k-NN, Naive Bayes, decision tree and to classify the results of our research. K-NN is a classifier in sentiment analysis that uses distance-based measures to classify the main idea is that documents that fall into the same class category are more likely to be "similar" or close to each other based on a similarity measure such as cosine defined in the dictionary [38]. DTs use the concept of a hierarchical structure to carry out the classification process. The decision tree algorithm process starts from the root node to the leaf node which is done recursively. The process in the decision tree is changing the shape of the data into a tree model and then changing the tree model into rules [39]. The algorithm used to build the decision tree is ID3. The algorithm in this decision tree uses the concept of calculating the entropy value. The concept of entropy is used to measure "how informative" a node is. The formula for finding entropy and information gain is as shown in:

$$Entropy = \sum_{i=1}^n -p_i \times \log_2 p_i \quad (4)$$

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{S_i}{S} \times Entropy(S_i) \quad (5)$$

Naive Bayes classifier is a classification method based on Bayes theorem calculations. Naive Bayes is a machine learning method that utilizes the concept of probability values and statistical calculations [40]. The basic concept of Naive Bayes is to predict future opportunities based on values that have existed in the past, so it is known as Bayes Theorem [41].

3.8. Evaluation

In evaluating the algorithm performance of our study, we use the reference confusion matrix. The confusion matrix represents the predictions and actual conditions of the data generated by the classification method algorithm in our study [42]. Based on the value of the confusion matrix, we can determine the value of the accuracy, precision, recall, and F1 scores from the sentiment analysis classification method.

$$Precision = \frac{TP}{(TP+FP)} \quad (6)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (7)$$

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (8)$$

$$F1\ Score = \frac{2 \times Recall \times Precision}{Recall+Precision} \quad (9)$$

Recall and precision are often used to evaluate the effectiveness of machine learning algorithms. Accuracy is the ratio of true predictions (positive and negative) to the whole data. Precision value is true positive prediction ratio compared to overall positive predicted results. Recall is a true positive prediction ratio compared to overall true positive data. F1 score is a comparison of weighted average precision and recall.

4. RESULTS AND DISCUSSION

4.1. Spelling checker

After take 1026 tweets [43] from twitter with API Twitter. The 1026 tweets contain some slang words, so that we needed to correct it with spelling checker. We had got 113 slang words from training data and 139 slang words from testing data. The word calculation process after and before cleaning can be seen Table 1.

Table 1. Slang words correction by spelling checker

Dataset	Slang words without spelling checker		Slang words with spelling checker	
	Before	After	Before	After
Training Data	113	113	113	-
Testing Data	139	139	139	-
Total	242	242	242	0

From Table 1, it could be seen that before the spelling checker process, training data had slang words totaling 113 words and testing data had slang words totaling 139 words. After the correction process was done used the listed of slang word. The result was there were no more slang words in the training data and testing data. The Spelling Checker process will check the slang words in each dataset, both training data and testing data. Each slang words will be matched with the slang words dictionary to be converted into standard words, so the effect is that all slang words become non-existent or have a value of 0, because all slang words have been converted into standard words.

4.2. Expansion words with wordnet

From 1026 tweets, we take sample positive and negative words. The samples were 75 positive words and 75 negative words. The reasoned we took samples only 75 words from each positive and negative words was if we took a large sample of positive and negative words, the results of synonyms and antonyms produced by wordnet were few, because sometimes wordnet produces duplication of positive and negative words similar to the samples. Details of the calculation of word expansion with wordnet in the training data and testing data can be seen in the Table 2.

Table 2. Expansion word by wordnet

Corpus list positive and negative words	Expansion word by wordnet		
		Synonyms	Antonyms
Positive Word	75	117	133
Negative Words	75	134	115
Total	150	251	248

From Table 2 it could be seen the results of the expansion of the corpus listed of positive and negative words used synonyms and antonyms by wordnet. The total listed of positive words which previously was 75 positive words, after extending their meaning with wordnet, produced 117 synonyms of positive words and 133 antonyms negative words. The total listed of negative words which previously was 75 words, after extending with wordnet, produced 134 synonyms of negative words and 248 antonyms of negative words. So the total new positive and negative word listed from the results of the expansion of wordnet which previously amounted to 150 words to 499 words.

4.3. Feature extraction

Valued from Table 3 obtained from calculating the weight of TF and IDF between listed of positive and negative words from Table 3 with listed of tweets, so that positive and negative labeling could be used used the TF-IDF algorithm for each tweet. this labeling had been evaluated used the sentiment analysis classification methods. The number of positive labels in training data before used spelling checker was 233 tweets and after used the spelling checker became 188 tweets, testing data from 231 tweets to 177 tweets. The number of negative labels in training data before used spelling checker was 220 tweets and after used the spelling checker became 262 tweets, testing data from 110 tweets to 132 tweets. The number of neutral labels in training data before used spelling checker was 147 tweets and after used the spelling checker became 150 tweets, testing data from 85 tweets to 117 tweets. The results of the labeling testing data in Table 4 had been used as a comparison in determining the accuracy of each classifier methods used in this studied.

Table 3. Dataset feature extraction with TF-IDF algorithm

	Without spelling checker and wordnet		With spelling checker and wordnet	
	Training data	Testing data	Training data	Testing data
Positive Label	233	231	188	177
Negative Label	220	110	262	132
Neutral Label	147	85	150	117
Total	600	426	600	426

4.4. Expansion words with wordnet

The results total values of true positive, true negative, false positive, and false negative for each of classification methods in sentiment analysis in this researched can be seen in Table 4. Table 4 also produces lower false negative, false positive and true negative values, because the resulting dataset contains standard words through the spelling checker process before preprocessing and when performing the feature extraction process combined with wordnet. After getting the true positive, true negative, false positive and false negative values, then we can calculate the accuracy, precision, recall, and F1 Score values. Calculation details can be seen in the Table 5.

Table 4. Confusion matrix model

Variable	True Positive	True Negative	False Positive	False Negative
K-NN, K=3	149	109	235	107
Naive Bayes	132	114	233	121
Decision tree	174	108	258	60
K-NN with wordnet, K=3	223	101	129	147
Naive Bayes with checker	201	105	122	172
Decision tree with checker	241	107	117	135
K-NN with spelling checker, K=3	212	130	181	77
Naive Bayes with wordnet	191	109	177	123
Decision tree with wordnet	233	109	151	107
K-NN using with checker and wordnet, K=3	322	110	98	70
Naive Bayes with Spelling Checker and wordnet	319	105	105	71
Decision tree with spelling checker and wordnet	269	181	77	73

Table 5. Comparison of result using cross validation

Method	Precision	Recall	Accuracy	F1 Score
K-NN, K=3	39%	58%	43%	47%
Naive Bayes	36%	52%	41%	43%
Decision tree	40%	74%	47%	52%
K-NN with Wordnet, K=3	63%	60%	54%	62%
Naive Bayes with checker	62%	54%	51%	58%
Decision tree with checker	67%	64%	58%	66%
K-NN with spelling checker, K=3	54%	73%	57%	62%
Naive Bayes with wordnet	52%	61%	50%	56%
Decision tree with wordnet	61%	69%	57%	64%
K-NN using with checker and wordnet, K=3	77%	82%	72%	79%
Naive Bayes with spelling checker and wordnet	75%	82%	71%	78%
Decision tree with spelling checker and wordnet	78%	79%	75%	78%

From Table 5, it could be seen that the comparison of the results valued of the calculations of Precision, Recall, Accuracy and F1 Score in twelve classification method. The values of precision, recall, accuracy and f1 score could be obtained from the confusion matrix values in Table 4. From the figure, it is explained in outline that the values of precision, recall, accuracy and F1 score have increased after using spelling checker and Wordnet. This happens because the results of the confusion matrix in Table 4 produce a larger true positive value and a lower false positive, false negative and true negative value.

Figure 3 depicts a diagram of the calculation results from the confusion matrix. It can be seen that there has been an increase in terms of accuracy for the sentiment analysis method whose training data and testing data have gone through word expansion using Wordnet and the data set has been cleaned of slang words using a spelling checker. The accuracy value can be obtained from the calculation in Table 5 which is represented in the form of a diagram in Figure 3.

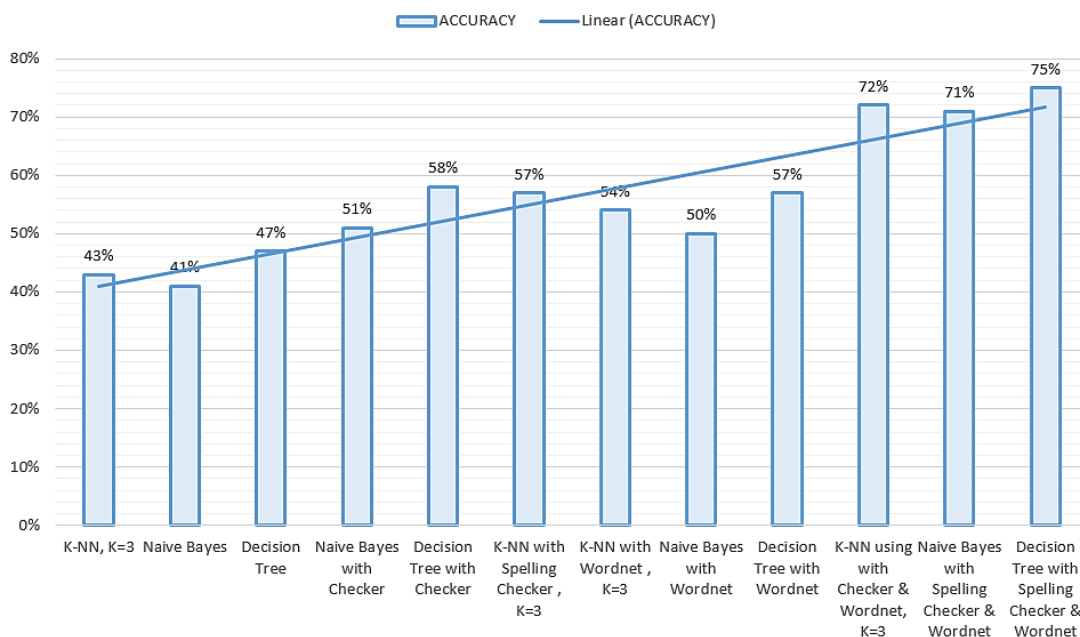


Figure 3. Confusion matrix of all classification methods

5. CONCLUSION

The conclusion of this paper is after using a spelling checker to convert slang words into standard words and using wordnet as an NLP to find similar words, spelling checker and wordnet in sentiment analysis can improve the value of accuracy score in sentiment analysis. This happens because the values of false negative, false positive and true negative have decreased in the confusion matrix calculation process after using spelling checkers and after using wordnet the true positive value in the confusion matrix increases, because wordnet allows the feature extraction process to recognize words and their similarities. Based on (6)-(9) said that if the value of false positive, false negative and true negative decreases then the accuracy value will increase. It is explained that the calculation of this accuracy value is very dependent on the false positive, false negative, and true negative values.

ACKNOWLEDGEMENTS

This work was partly supported by Lecturers from Bina Nusantara University. The author also does not forget to thank you for the feedback, comments and suggestions that are useful for the journal that the author has created, resulting in a much better journal presentation.

REFERENCES





- [1] R. A. Stine, "Sentiment analysis," *Annual Review of Statistics and Its Application*, vol. 6, no. 1, pp. 287–308, Mar. 2019, doi: 10.1146/annurev-statistics-030718-105242.
- [2] F. M. Kundi, S. Ahmad, A. Khan, and M. Z. Asghar, "Detection and scoring of Internet Slangs for sentiment analysis using SentiWordNet," *Life Science Journal*, vol. 11, no. 9, pp. 66–72, 2014, doi: 10.6084/M9.FIGSHARE.1609621.
- [3] D. Hládek, J. Staš, and M. Pleva, "Survey of automatic spelling correction," *Electronics (Switzerland)*, vol. 9, no. 10, pp. 1–29, Oct. 2020, doi: 10.3390/electronics9101670.
- [4] D. Moeljadi and F. Bond, "Identifying and exploiting definitions in Wordnet Bahasa," in *Proceedings of the 8th Global WordNet Conference, GWC 2016*, 2016, pp. 226–232, [Online]. Available: <https://aclanthology.org/2016.gwc-1.33>.
- [5] N. Mohammed and Y. Abdellah, "The vocabulary and the morphology in spell checker," *Procedia Computer Science*, vol. 127, pp. 76–81, 2018, doi: 10.1016/j.procs.2018.01.100.
- [6] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International Journal of Lexicography*, vol. 3, no. 4, pp. 235–244, 1990, doi: 10.1093/ijl/3.4.235.
- [7] D. Gunawan and A. Amalia, "The design of lexical database for Indonesian language," *IOP Conference Series: Materials Science and Engineering*, vol. 180, no. 1, p. 012052, Mar. 2017, doi: 10.1088/1757-899X/180/1/012052.
- [8] H. K. Azad and A. Deepak, "A new approach for query expansion using Wikipedia and WordNet," *Information Sciences*, vol. 492, pp. 147–163, Aug. 2019, doi: 10.1016/j.ins.2019.04.019.
- [9] A. Yunus and M. Masum, "A Context Free Spell Correction Method using Supervised Machine Learning Algorithms," *International Journal of Computer Applications*, vol. 176, no. 27, pp. 36–41, Jun. 2020, doi: 10.5120/ijca2020920288.
- [10] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment in short strength detection informal text," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, Dec. 2010, doi: 10.1002/asi.21416.

- [11] J. Ababneh, "Application of Naïve Bayes, Decision Tree, and K-Nearest Neighbors for Automated Text Classification," *Modern Applied Science*, vol. 13, no. 11, p. 31, Oct. 2019, doi: 10.5539/mas.v13n11p31.
- [12] M. Y. Soleh and A. Purwarianti, "A non word error spell checker for Indonesian using morphologically analyzer and HMM," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, ICEEI 2011*, Jul. 2011, pp. 1–6, doi: 10.1109/ICEEI.2011.6021514.
- [13] L. Dey, S. Chakraborty, A. Biswas, B. Bose, and S. Tiwari, "Sentiment Analysis of Review Datasets Using Naïve Bayes' and K-NN Classifier," *International Journal of Information Engineering and Electronic Business*, vol. 8, no. 4, pp. 54–62, Jul. 2016, doi: 10.5815/ijieeb.2016.04.07.
- [14] Suhariyanto, A. Firmanto, and R. Sarno, "Prediction of Movie Sentiment based on Reviews and Score on Rotten Tomatoes using SentiWordnet," in *Proceedings - 2018 International Seminar on Application for Technology of Information and Communication: Creative Technology for Human Life, iSemantic 2018*, Sep. 2018, pp. 202–206, doi: 10.1109/ISEMANTIC.2018.8549704.
- [15] S. D. Jadhav and H. P. Channe, "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques," *International Journal of Science and Research (IJSR)*, vol. 5, no. 1, pp. 1842–1845, 2016, doi: 10.21275/v5i1.nov153131.
- [16] A. Verasius and D. Sano, "Comparison of Prediction Accuracy Between Decision Tree, Naïve Bayes and K-Nn on Web Phising," in *PROCEEDING International Conference Technopreneur and Education 2018*, 2018, pp. 380–384.
- [17] Z. E. Rasjid and R. Setiawan, "Performance Comparison and Optimization of Text Document Classification using k-NN and Naïve Bayes Classification Techniques," *Procedia Computer Science*, vol. 116, pp. 107–112, 2017, doi: 10.1016/j.procs.2017.10.017.
- [18] D. A. Adeniyi, Z. Wei, and Y. Yongquan, "Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method," *Applied Computing and Informatics*, vol. 12, no. 1, pp. 90–108, Jan. 2016, doi: 10.1016/j.aci.2014.10.001.
- [19] M. Unnisa, A. Ameen, and S. Raziuddin, "Opinion Mining on Twitter Data using Unsupervised Learning Technique," *International Journal of Computer Applications*, vol. 148, no. 12, pp. 12–19, Aug. 2016, doi: 10.5120/ijca2016911317.
- [20] T. Pranckevičius and V. Marcinkevičius, "Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification," *Baltic Journal of Modern Computing*, vol. 5, no. 2, 2017, doi: 10.22364/bjmc.2017.5.2.05.
- [21] D. T. Alamanda, A. Ramdhani, I. Kania, W. Susilawati, and E. S. Hadi, "Sentiment Analysis Using Text Mining of Indonesia Tourism Reviews via Social Media," *International Journal of Humanities, Arts and Social Sciences*, vol. 5, no. 2, pp. 72–82, Apr. 2019, doi: 10.20469/ijhss.5.10004-2.
- [22] E. I. Elmurghi and A. Gherbi, "Unfair reviews detection on Amazon reviews using sentiment analysis with supervised learning techniques," *Journal of Computer Science*, vol. 14, no. 5, pp. 714–726, May 2018, doi: 10.3844/jcsp.2018.714.726.
- [23] N. Zukarnain, B. S. Abbas, S. Wayan, A. Trisetyarso, and C. H. Kang, "Spelling Checker Algorithm Methods for Many Languages," in *Proceedings of 2019 International Conference on Information Management and Technology, ICIMTech 2019*, Aug. 2019, pp. 198–201, doi: 10.1109/ICIMTech.2019.8843801.
- [24] M. P. Patel and M. K. Mistry, "A Review: Text Classification on Social Media Data," *IOSR Journal of Computer Engineering Ver. IV*, vol. 17, no. 1, pp. 2278–661, 2015, doi: 10.9790/0661-17148084.
- [25] M. A. Fauzi, R. F. N. Firmansyah, and T. Afrianto, "Improving sentiment analysis of short informal Indonesian product reviews using synonym based feature expansion," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 3, pp. 1345–1350, Jun. 2018, doi: 10.12928/TELKOMNIKA.v16i3.7751.
- [26] D. Kumar, A. Kumar, M. Singh, A. Patel, and S. Jain, "Modern WordNet: An Affective Extension of WordNet," in *New Trends in Computational Vision and Bio-inspired Computing*, Cham: Springer International Publishing, 2020, pp. 527–536.
- [27] Indoindian, "Popular Slang Words in Bahasa Indonesia," *Indoindians*, 2011. <https://www.indoindians.com/popular-slang-words-in-bahasa-indonesia/> (accessed Apr. 01, 2020).
- [28] E. Asgarian, M. Kahani, and S. Sharifi, "HesNegar: Persian Sentiment WordNet," *Signal and Data Processing*, vol. 15, no. 1, pp. 71–86, Jun. 2018, doi: 10.29252/jsdp.15.1.71.
- [29] L. Meng, R. Huang, and J. Gu, "A Review of Semantic Similarity Measures in WordNet," *International Journal of Hybrid Information Technology*, vol. 6, no. 1, pp. 1–12, 2013.
- [30] K. Orkphol and W. Yang, "Word sense disambiguation using cosine similarity collaborates with word2vec and wordNet," *Future Internet*, vol. 11, no. 5, p. 114, May 2019, doi: 10.3390/fi11050114.
- [31] A. Wahdan, S. Hantoobi, S. A. Salloum, and K. Shaalan, "A systematic review of text classification research based on deep learning models in Arabic language," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 6629–6643, Dec. 2020, doi: 10.11591/IJECE.V10I6.PP6629-6643.
- [32] K. Matsumoto, F. Ren, M. Matsuoka, M. Yoshida, and K. Kita, "Slang feature extraction by analysing topic change on social media," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 1, pp. 64–71, Mar. 2019, doi: 10.1049/trit.2018.1060.
- [33] P. Zhang and Y. Fang, "Research on Text Classification Algorithm Based on Machine Learning," *Journal of Physics: Conference Series*, vol. 1624, no. 4, p. 042010, Oct. 2020, doi: 10.1088/1742-6596/1624/4/042010.
- [34] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.
- [35] Y. Wang *et al.*, "Unsupervised anomaly detection with compact deep features for wind turbine blade images taken by a drone," *IPSI Transactions on Computer Vision and Applications*, vol. 11, no. 1, p. 3, Dec. 2019, doi: 10.1186/s41074-019-0056-0.
- [36] E. Haddi, X. Liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," *Procedia Computer Science*, vol. 17, pp. 26–32, 2013, doi: 10.1016/j.procs.2013.05.005.
- [37] K. Chen, Z. Zhang, J. Long, and H. Zhang, "Turning from TF-IDF to TF-IGM for term weighting in text classification," *Expert Systems with Applications*, vol. 66, pp. 1339–1351, Dec. 2016, doi: 10.1016/j.eswa.2016.09.009.
- [38] M. Rezwanul, A. Ali, and A. Rahman, "Sentiment Analysis on Twitter Data using KNN and SVM," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, 2017, doi: 10.14569/ijacsa.2017.080603.
- [39] A. H. Mohammad, O. Al-momani, and T. Alwada'n, "Arabic Text Categorization using k-nearest neighbour, Decision Trees (C4.5) and Rocchio Classifier: A Comparative Study," *International Journal of Current Engineering and Technology*, vol. 6, no. 2, pp. 477–482, 2016, [Online]. Available: <http://inpressco.com/wp-content/uploads/2016/03/Paper16477-482.pdf>.
- [40] W. M. Hadi, M. Salam, and J. A. Al-Widian, "Performance of NB and SVM classifiers in Islamic Arabic data," in *ACM International Conference Proceeding Series*, 2010, pp. 1–6, doi: 10.1145/1874590.1874604.
- [41] D. Nurcahyono, W. P. Putra, A. Najib, and T. R. Tulili, "Analysis sentiment in social media against election using the method naive Bayes," *Journal of Physics: Conference Series*, vol. 1511, no. 1, p. 012003, Mar. 2020, doi: 10.1088/1742-6596/1511/1/012003.





- [42] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Information Sciences*, vol. 507, pp. 772–794, Jan. 2020, doi: 10.1016/j.ins.2019.06.064.
- [43] R. Andika, "Sentiment Analysis Project for thesis bina nusantara university," *Binus University*, 2020. <https://github.com/randika95/sentimentAnalysis> (accessed Apr. 03, 2021).

BIOGRAPHIES OF AUTHORS



Rizky Andika     Born February 09, 1995, a postgraduate student at Bina Nusantara University. Currently working as Supervisor Software Developer at PT Metropolitanland Tbk. Previously, Rizky Andika had also worked at PT Pamapersada Nusantara as an IT development staff for almost 2 years. His expertise consists of in application development, business intelligence, artificial intelligence and big data. He can be contacted at email: rizky.andika@binus.ac.id.



Dr. Suharjito     is Lecturer of Magister Teknologi Informasi, currently involve on research of sign language recognition using deep learning especially for Indonesian language (SIBI). He uses CNN and RNN to develop model for classifying input videos of Indonesian sign language based on words and sentences. His main expertise are E-Learning, Optimization, Software Engineering, Unified Modeling Language, Artificial Neural Networks, Scheduling Neural Networks, Evolutionary Computation, Supply Chain Management, and Fuzzy Logic. To access his current research interest, please visit his Sinta, Google Scholar and Scopus profile. He can be contacted at email: suharjito@binus.edu.