Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 4, No. 4, November 2019, Pp. 375-380

The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on indonesian-language texts

Aditya Wiha Pradana*[1], Mardhiya Hayaty[2]
Universitas AMIKOM Yogyakarta, Indonesia[1,2]

Article Info
Keywords:
Stemming, Stopword Removal, Preprocessing, Text Mining, Classification

Article history:
Received 14 August 2019
Revised 5 October 2019
Accepted 24 October 2019
Published 30 October 2019

Cite:
Pradana, A., & Hayaty, M. (2019). The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 4(4).
doi:http://dx.doi.org/10.22219/kinetik.v4i4.912

*Corresponding author.
Aditya Wiha Pradana
E-mail address:
Aditya.9295@students.amikom.ac.id

Abstract
Preprocessing is an essential task for sentiment analysis since textual information carries a lot of noisy and unstructured data. Both stemming and stopword removal are pretty popular preprocessing techniques for text classification. However, the prior research gives different results concerning the influence of both methods toward accuracy on sentiment classification. Therefore, this paper conducts further investigations about the effect of stemming and stopword removal on Indonesian language sentiment analysis. Furthermore, we propose four preprocessing conditions which are with using both stemming and stopword removal, without using stemming, without using stopword removal, and without using both. Support Vector Machine was used for the classification algorithm and TF-IDF as a weighting scheme. The result was evaluated using confusion matrix and k-fold cross-validation methods. The experiments result show that all accuracy did not improve and tends to decrease when performing stemming or stopword removal scenarios. This work concludes that the application of stemming and stopword removal technique does not significantly affect the accuracy of sentiment analysis in Indonesian text documents.

1. Introduction

Recently, the research of sentiment analysis on social media has attracted many researchers in the world. Such as is Twitter [1] as a microblogging tool allowing users free expression. Sentiment Analysis is used to find out the opinion about a topic which is as positive, negative, and neutral sentiments [2].

The word structure of comments on social media is irregular and contains much noise, and it is a challenge in conducting sentiment analysis [3][4], therefore the role of data preprocessing is essential because it can affect the accuracy and cannot be ignored when conducting sentiment analysis [4]. Preprocessing data is the process of cleaning and preparing data for review [5]. Preprocessing techniques for text classification are stemming and stopwords removal [6][7]. The "stemming" is turning a word into a root word by removing the phrase prefix [8], While the "stopwords removal" is removed words that often appear and do not have any meaning [9].

Previous research used the TF and TF-IDF scenarios with the Naïve Bayes classification algorithm. Stemming had no significant effect on the classification accuracy of both the TF and TF-IDF scenarios [8]. Preprocessing in Arabic text using SVM algorithm has stated that normalization can increase efficiency from 96.66% to 97.50%, while "stemming" actually reduces accuracy using both ISRI Stemmer and Tashaphyne with an accuracy value of 93.06% and 95.83% [10]. Preprocessing text in English documents has been done and was concluded that using stemming and stopwords removal improves the accuracy of sentiment analysis in all situations [11].

However, whether stemming and removal of stopwords can also improve the accuracy of sentiment analysis in Indonesian documents given that a word has a different meaning in the language used. The purpose of this paper is to examine the effect of stemming and deletion of stopwords on the accuracy of sentiment analysis in Indonesian text documents-sentiment analysis using the SVM algorithm.

2. Research Method

In this section Figure 1, we briefly describe the experimental design. In the first step, we collect the data from twitter used GetOldTweets. In the second step, pre-processing; consist of stemming dan stopwords removal to clean the data from noise. Afterward, weighting TF-IDF. The classification process used Support Vector Machine (SVM), and the end proses evaluate the performance with confusion matrix and cross-validation.

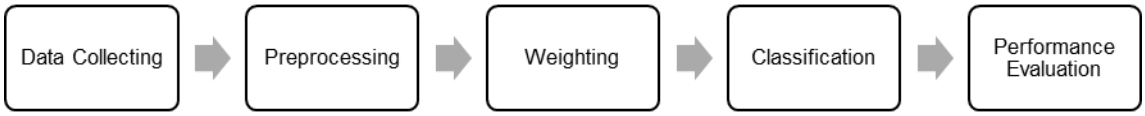

Figure 1. Experimental Design

2.1 Data Collecting

This study uses data from Twitter user comments collected using python program GetOldTweets. There are 2000 data tweets and labeled manually into two sentiment polarities, a positive and negative sentiment. With the number of positive tweets is 675, and 1325 negative tweets, shown the Table 1.

Table 1. Tweet Polarity

Polarity	Quantity
Positive	675
Negative	1325

2.2 Preprocessing

Preprocessing data is the process of cleaning and preparing data for analysis [5]. Preprocessing can also be used to reduce computational processes and feature space which can improve performance accuracy and classification. In the case of text classification, many preprocessing techniques can be used [12][13]. Preprocessing techniques used in this study are as follows.

a. Case folding aims to change all letters in a text document into lowercase letters [14].
b. Removing the URL, Many researchers argue that the URL does not carry information about the sentiment on Twitter [13].
c. Removing numbers, numbers on tweets has no effect on sentiment analysis, and deleting them can reduce noise and increase efficiency [15].
d. Removing punctuation, Punctuation is a unique character like an exclamation mark, comma, question mark, and others. It not required in sentiment classification [16].
e. Removing special characters from Twitter: cleanups such as deleting Twitter user usernames, hashtags, and non-ASCII characters [12].
f. Removing word less than three characters.
g. Normalization is the process of changing a word to standard form [17]. Normalization in this study also changed the slang word to ordinary word.
h. Stemming aims to turn a word into a root word by removing the phrase prefix and prefix [8]. This study uses Sastrawi Stemmer adapted from the Nazief-Andriani [18] algorithm with a modified confix-stripping [19].
i. Stopwords Removal is a word that often appears and does not have any meaning [9]. Stopwords in Indonesian such as "*yang*", "*di*", "*untuk*", and "*dari*". In this study, the stopwords list used is Sastrawi stoplist.
j. Tokenization is a task to separate the full text string into list of separate words [4].

In this study, we conducted four models pre-processing, shown the Table 2.

Table 2. Preprocessing Model

Preprocessing Model	Preprocessing
Stem - Stop	Apply all pre-processing
No Stem	Without stemming
No Stop	Without stopwords removal
No Stem – No Stop	Without stemming and without stopwords removal

2.3 Weighting

The preprocessing stage finished, the next step is weighting uses Term Frequency - Inverse Document Frequency. TF-IDF reflects the importance of a word in a text document [20]. The Level of importance increases when a word appears several times in a document, but the frequency of words appearing a document keep balanced.

Term Frequency (TF) is the frequency with which words appear in a document. For term t_i in a document, can be formulated follows Equation 1 [21].

$$tf_{i,j} = n_{i,j} \quad (1)$$

$n_{i,j}$ is the number of occurrences of each word t_i on d_j document. Inverse Document Frequency (IDF) measures the general importance of a word in a document. I have formulated follows Equation 2.

$$idf_{i,j} = log \frac{D}{df_{i,j}} \quad (2)$$

D is the total number of text documents $df_{i,j}$ is a number of document d_j which contains the term t_i. TF-IDF is a combination of TF and IDF, the formula follows Equation 3.

$$tf - idf_{i,j} = tf_{i,j} * idf_{i,j} \quad (3)$$

2.4 Classification

Classification algorithm using the Support Vector Machine algorithm. SVM is a supervised machine learning algorithm, and this approach works if there are trained data and targeted data. The Support Vector Machine algorithm is a statistical classification approach based on maximizing the margin between instances and hyperplane separation [22]. SVM separate class of data by using three different lines, one for the main separating line and two other lines are supported line [23].

For example there is data training x and label y ∈{-1,1} to show the label class. Whereas -1 negative class and 1 is positive class. With the result that hyperplane formulas as Equation 4 [22].

$$w.x + b = 0 \quad (4)$$

In the equation above, w is the vector weight, and b is the bias factor. In Figure 2, two hyperplanes that determines the margin side. For the positive side hyperplane, the formula follows Equation 5.

$$w.x + b \geq +1 \quad (5)$$

and negative side hyperplane, the formula follows Equation 6.

$$w.x + b \leq -1 \quad (6)$$

while the distance between the two hyperplanes is shown on Equation 7.

$$\frac{2}{\|w\|} \quad (7)$$

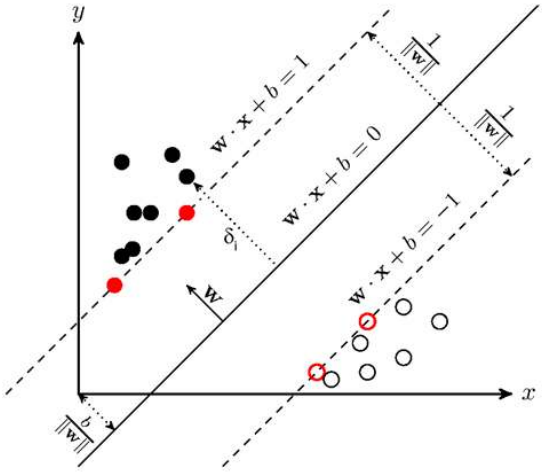

Figure 2. Support Vector Machine Hyperplanes

2.5 Performance Evaluation

The Evaluation to measure how appropriate the proposed method for classifying text. Evaluation using confusion matrix and k-fold cross-validation. Confusion matrix table for prediction of two classes as follows Table 3 [12].

Table 3. Confusion Matrix

		Actual Class	
		Class-1	Class-2
Predicted Class	Class-1	True Positive (TP)	False Negative (FN)
	Class-2	False Positive (FP)	True Negative (TN)

The confusion matrix table above is used to calculate the accuracy of the proposed method. The formula calculates the accuracy follows Equation 8.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Afterward, further evaluation uses k-fold cross-validation. The method works are to divide the dataset randomly as many as "*k*" separate parts of the same size, and each piece is used to test the model with a classification algorithm [24]. Calculation of k-fold cross-validation produces average accuracy. In this study, the value of "*k*" is 10.

3. Results and Discussion

This study uses python programming and machine learning tools called scikit-learn to conduct all the experiments. Moreover, this research tested classification performance 35 times for each preprocessing approach that we proposed for this study and using the average accuracy to compare each other's performance.

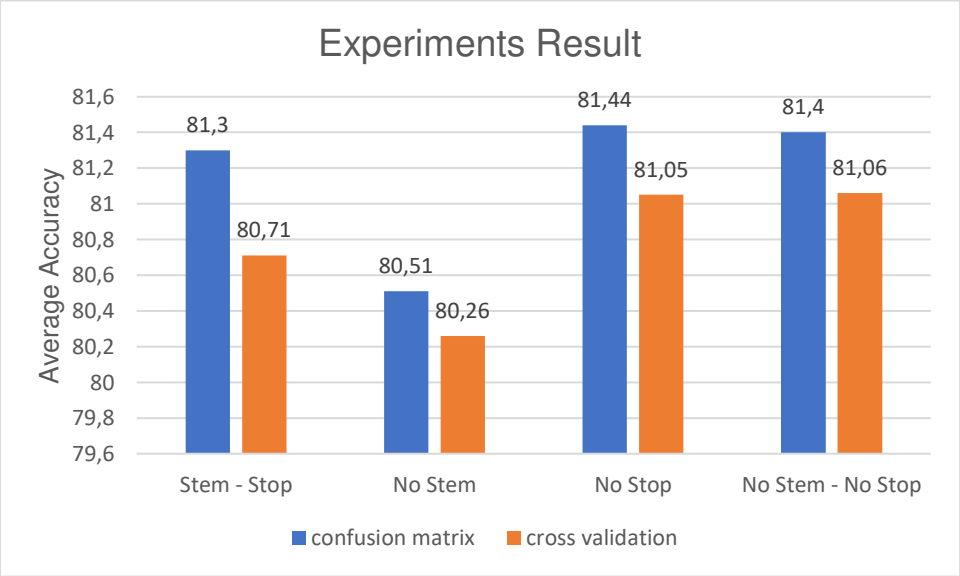

Figure 3. Experiments Result

The experiments result using both of the confusion matrices, and k-fold cross-validation is shown in Figure 3. The best accuracy of the confusion matrix test used the stemming with an accuracy score of 81.44%. While the stopword removal implementation got the worst accuracy score of 80.51%, the other's are both stemming and stopword elimination is 81.3% and without implementing both 81.4%.

On the other hand, the k-fold cross-validation test shown the different results in terms of the best accuracy with accuracy score was 81.06% on without both stemming and stopwords removal. While the lowest accuracy score is 80.26% with stopword elimination and the others are 80.71% and 81.05%. The result of both the confusion matrix and k-fold cross validation has shown there are no significant differences between all preprocessing model. The difference between the highest and the lowest accuracy are only 0.93% and 0.8%.

Compare to the prior research [8], and There is very lightly enhancement regarding the accuracy differences with the same situation. The effect of stemming technique implementation, which in this experiment got 0.93% better while the prior research reduces the accuracy by 1.34% when stemming technique is applied.

Accuracy does not increase significantly because the way stemming works only cuts a word into a basic word so that sometimes it has a misunderstanding, whereas, for sentiment analysis, the meaning of the word has a critical role in judging an opinion person.

4. Conclusion

This paper examines the effect of stemming and stopword removal implementation on the preprocessing step toward the accuracy of sentiment analysis in Indonesian text documents. The result, the application of the stemming and stopword removal technique on the preprocessing stage does not significantly affect the accuracy of sentiment analysis with the accuracy differences of the highest and the worst are only 0.93% and 0.8% on both confusion matrix and k-fold cross-validation test.

In future work, we suggest trying the lemmatization technique for conducting sentiment analysis in which the lemmatization does full morphological analysis to identify the root word for each word accurately.

References

[1] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4: Sentiment Analysis in Twitter," in *Proceedings ofthe 11th International Workshop on Semantic Evaluations (SemEval-2017)*, 2017, Pp. 502–518. http://dx.doi.org/10.18653/v1/S16-1001

[2] Y. Wang, K. Kim, B. Lee, and H. Y. Youn, "Word clustering based on POS feature for efficient twitter sentiment analysis," *Human-centric Comput. Inf. Sci.*, Vol. 8, No. 17, Pp. 1–25, 2019. https://doi.org/10.1186/s13673-018-0140-y

[3] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," in *IISA 2016 - 7th International Conference on Information, Intelligence, Systems and Applications*, 2016. https://doi.org/10.1109/IISA.2016.7785373

[4] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis," *Expert Syst. Appl.*, Vol. 110, Pp. 298–310, 2018. https://doi.org/10.1016/j.eswa.2018.06.022

[5] M. Mhatre, D. Phondekar, P. Kadam, A. Chawathe, and K. Ghag, "Dimensionality Reduction for Sentiment Analysis using Pre-processing Techniques," in *Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC)*, Pp. 16–21, 2017. https://doi.org/10.1109/ICCMC.2017.8282676

[6] H. M. Zin, N. Mustapha, M. A. A. Murad, and N. M. Sharef, "The Effects of Pre-Processing Strategies in Sentiment Analysis of Online Movie Reviews," *AIP Conf. Proc.*, Vol. 1891, No. 1, Pp. 020089–1–020089–7, 2017. https://doi.org/10.1063/1.5005422

[7] S. Gharatkar, A. Ingle, T. Naik, and A. Save, "Review Preprocessing Using Data Cleaning And Stemming Technique," in *International Conference on Innovations in information Embedded and Communication Systems (ICIIECS)*, 2017. https://doi.org/10.1109/ICIIECS.2017.8276011

[8] A. F. Hidayatullah, "The Influence of Stemming on Indonesian Tweet Sentiment Analysis," in *Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2015)*, Pp. 127–132, 2015.

[9] K. V. Ghag and K. Shah, "Comparative analysis of effect of stopwords removal on sentiment classification," in *IEEE International Conference on Computer Communication and Control (IC4-2015)*, 2015. https://doi.org/10.1109/IC4.2015.7375527

[10] R. M. Sallam and M. Hussein, "Improving Arabic Text Categorization using Normalization and Stemming Techniques," *Int. J. Comput. Appl.*, Vol. 135, No. 2, Pp. 38–43, 2016.

[11] E. Haddi, X. Liu, and Y. Shi, "The Role of Text Pre-processing in Sentiment Analysis," *Procedia Comput. Sci.*, Vol. 17, Pp. 26–32, 2013. https://doi.org/10.1016/j.procs.2013.05.005

[12] A. Fathan Hidayatullah, C. I. Ratnasari, and S. Wisnugroho, "Analysis of Stemming Influence on Indonesian Tweet Classification," *TELKOMNIKA*, Vol. 14, No. 2, Pp. 665–673, 2016. http://dx.doi.org/10.12928/TELKOMNIKA.v14i1.3113

[13] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis," *IEEE Access*, Vol. 5, Pp. 2870–2879, 2017. https://doi.org/10.1109/ACCESS.2017.2672677

[14] C. Slamet, A. R. Atmadja, D. S. Maylawati, R. S. Lestari, W. Dharmalaksana, and M. A. Ramdhani, "Automated Text Summarization for Indonesian Article Using Vector Space Model Model," *IOP Conf. Ser. Mater. Sci. Eng.*, Vol. 288, No. 1, 2018. https://doi.org/10.1088/1757-899X/288/1/012037

[15] M. Khader, A. Awajan, and G. Al-Naymat, "The Effects of Natural Language Processing on Big Data Analysis: Sentiment Analysis Case Study," in *ACIT 2018 - 19th International Arab Conference on Information Technology*, 2018. https://doi.org/10.1109/ACIT.2018.8672697

[16] A. Filcha and M. Hayaty, "Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa," *JUITA J. Inform.*, Vol. 7, No. 1, Pp. 25, 2019. https://dx.doi.org/10.30595/juita.v7i1.4063

[17] S. M. Arif and M. Mustapha, "The Effect of Noise Elimination and Stemming in Sentiment Analysis for Malay Documents," *Proc. Int. Conf. Comput. Math. Stat. (iCMS 2015)*, Pp. 93–102, 2015. https://doi.org/10.1007/978-981-10-2772-7_10

[18] J. Asian, H. E. Williams, and S. M. M. Tahaghoghi, "Stemming Indonesian," in *ACSC '05 Proceedings of the Twenty-eighth Australasian conference on Computer Science*, Vol. 38, Pp. 307–314, 2005. https://dx.doi.org/10.1145/1316457.1316459

[19] J. Asian, B. Nazief, and H. Williams, "Stemming Indonesian : A confix-stripping approach," *ACM Trans. Asian Lang. Inf. Process.*, Vol. 6, No. 13, 2007. https://doi.org/10.1145/1316457.1316459

[20] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, Vol. 57, Pp. 117–126, 2016. https://doi.org/10.1016/j.eswa.2016.03.028

[21] G. Li and J. Li, "Research on Sentiment Classification for Tang Poetry based on TF-IDF and FP-Growth," in *Proceedings of 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2018*, Pp. 630–634,2018. https://doi.org/10.1109/IAEAC.2018.8577715

[22] Y. A. L. Amrani, M. Lazaar, K. Eddine, and E. L. Kadiri, "Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis," *Procedia Comput. Sci.*, Vol. 127, Pp. 511–520, 2018. https://doi.org/10.1016/j.procs.2018.01.150

[23] M. Athoillah and R. K. Putri, "Handwritten Arabic Numeral Character Recognition Using Multi Kernel Support Vector Machine," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, Vol. 4, No. 2, Pp. 99, 2019. http://dx.doi.org/10.22219/kinetik.v4i2.724

[24] T. T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognit.*, Vol. 48, No. 9, Pp. 2839–2846, 2015. https://doi.org/10.1016/j.patcog.2015.03.009