Knowledge Engineering and Data Science (KEDS)
Vol 3, No 2, December 2020, pp. 106–111

pISSN 2597-4602
eISSN 2597-4637

Generating Javanese Stopwords List using K-means Clustering Algorithm

Aji Prasetya Wibawa [a, 1, *], Hidayah Kariima Fithri [a, 2], Ilham Ari Elbaith Zaeni [a, 3], Andrew Nafalski [b, 4]

[a] *Electrical Engineering Department, Universitas Negeri Malang*
Jl Semarang 5, Malang, East Java 65145, Indonesia

[b] *UniSA Education Futures, School of Engineering, University of South Australia*
SCT2-39 Mawson Lakes Campus, Adelaide, South Australia 5095, Australia

[1] *aji.prasetya.ft@um.ac.id *;* [2] *hidayah9a20@gmail.com;* [3] *Ilham.ari.ft@um.ac.id;* [4] *andrew.nafalski@unisa.edu.au*
** corresponding author*

ARTICLE INFO

Article history:
Received 1 December 2020
Revised 15 December 2020
Accepted 29 December 2020
Published online 30 December 2020

Keywords:
Stopwords
Javanese language
Clustering
K-means

ABSTRACT

Stopword removal necessary in Information Retrieval. It can remove frequently appeared and general words to reduce memory storage. The algorithm eliminates each word that is precisely the same as the word in the stopword list. However, generating the list could be time-consuming. The words in a specific language and domain must be collected and validated by specialists. This research aims to develop a new way to generate a stop word list using the K-means Clustering method. The proposed approach groups words based on their frequency. The confusion matrix calculates the difference between the findings with a valid stopword list created by a Javanese linguist. The accuracy of the proposed method is 78.28% (K=7). The result shows that the generation of Javanese stopword lists using a clustering method is reliable.

I. Introduction

Text processing in Information Retrieval (IR) requires text documents as primary data sources. However, not all words in the text document are used. Some words often appear in text documents and do not have meaning called stopword [1], stored in a stopword list called a stopword database (corpus) [2][3]. The stopword removal approach depends on this Corpus to remove unnecessary words on the text [4]. The formed word list must be in the same language [1][5]. Various stopword list has been developed for popular languages such as English, Chinese [6], Sanskrit [7], Arab [8], Gujarati [9], and Indonesia [10]. However, a stopword list for low resources language such as Javanese is not available yet.

Javanese is one of the traditional languages in Indonesia [11]. Javanese language has a level of politeness or known as unggah-ungguh, namely Ngoko, Madya and Krama [12][13]. Many historical documents, news, and stories are written in Javanese. Since the use of Javanese tends to become unpopular, retrieving information from such language could be difficult. The use of stopword removal may ease the IR process on Javanese text. Despite its benefit, list generation is quite complicated. In general, linguists manually label the substantial Corpus and store and send the result to separate storages. Therefore, an alternative to stopword list generation is badly needed.

This paper aims to explore the use of the clustering approach for creating a stopword list in Javanese. The words are excluded from the bag of words to speed up the text classification process [14]. The clustering method used is K-means, one of the fast algorithms in the big data processing. The method classifies a given set of data through a certain number of K clusters [15]. Determination of words included in the Stopword list is done by grouping words based on each word frequency. Clustering eases the way to determine the threshold of words that include stopwords.

https://doi.org/10.17977/um018v3i22020p106-111

II. Materials and Methods

The goal of this study is to generate a stopword list from the Javanese stopword corpus. The selected Javanese level of politeness is Ngoko, due to its usage and vocabularies [11][12]. Figure 1 shows the four stages in conducting this research.

The first stage is data collection. The dataset used was taken from the website Ki-demang.com in the Javanese Short Stories category. The data consists of 106 stories without considering page numbers and titles. The collection of stories is combined into a text document, used as the stopword generation dataset.

The second stage is data preprocessing: case folding, punctuation removal, tokenizing, and filtering. The first preprocessing, case folding, changes uppercase letters into lowercase letters. The punctuation removal deletes the punctuation characters and numbers from the dataset. Furthermore, the tokenizing step spits the dataset into a single word. This step produces 17,763 types of words and their frequency. The result of tokenizing is words, cleared from typographical errors, words without meaning, names, and non-Ngoko words, resulting in 14,384 types. This deletion is based on a Javanese-Indonesian and Indonesian Javanese translation dictionary. Table 1 shows examples of deleted words.

The dataset of 14,384 different words is submitted to Javanese linguists. The linguists group the dataset into two classes, namely stopwords and non-stopwords. Furthermore, general words (conjunction) considered as stopwords are 3,224 words. The non-stop words consist of 11,160 specific words: noun, verb, and adjectives. Table 2 shows the example of two categories.

The third stage is clustering the 14,384 unique words and their frequency. Figure 2 shows the pseudocode of the k-means clustering method [16].

The first k-means clustering stage determines the k value or the number of clusters. In the study, the k value is k=3, k=5, k=7, k=9, k=11, k=13, and k=15 [17]. The next step calculates the distance between data and centroid using Euclidean Distance [18].

Here, the results of each case are recognized in two classes: stopwords and non-stopwords. All words in cluster 0 are labeled as non-stopwords, while stopword is all words in other clusters. For example, if k=7, each word in the cluster 1 to 6 are stopwords, while the rest (in cluster 0) is non-stop words. This first assumption is based on the observation that words with high frequency [19] are outside cluster 0. Table 3 illustrates one example of the frequency distribution of stopwords when k=7. In this case, 680 words is labeled as stop words, where 13704 words are non-stopwords.

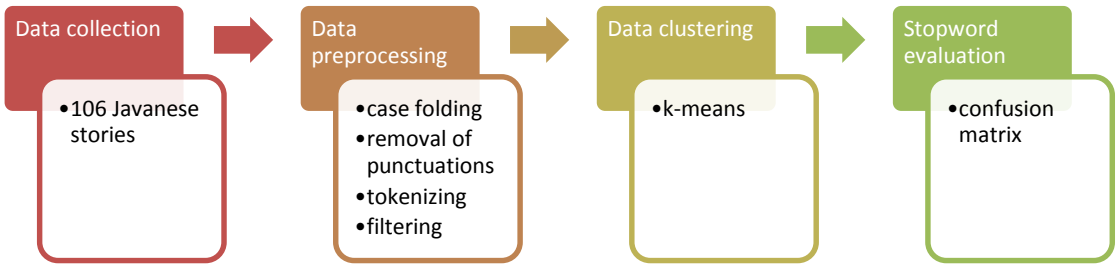

Fig. 1. Research stages

Table 1. Examples of deleted words

Typographical errors	Words without meaning	Names	Non-Ngoko words
lungaa	*lha*	*Ezza*	*wontening*
rilaaaaa	*we*	*Sukartiah*	*inbox*
ã³mongan	*lur*	*Yono*	*meresahkan*
ewosemono	*aaaaaaaa*	*Inah*	*mengganggu*
senaosa	*loh*	*Sumantri*	*pusaraning*
banjarpetambakan	*dhuk*	*Laras*	*out*
sesambhungane	*ugh*	*Yani*	*awalnya*
ampuunn	*sttt*	*Irvan*	*berbincang*

Table 2. Example of linguists' classified words

Stopwords	Non-stopwords
aku	*artane*
ana	*birahine*
apa	*cungkup*
dadi	*dhialog*
iki	*endhog*
ing	*garwamu*
kang	*jaitan*
sing	*karak*
wae	*langgananku*
yen	*macak*

Table 3. Stopwords and non –stopwords when K=7

k	Frequency distribution	Number of stopwords	Number of Non-stopwords
0	1-25	0	13704
1	2000-3000	3	
2	650-1050	13	
3	290-600	28	
4	1100-1600	5	
5	26-100	531	
6	105-290	100	
	Total	680	13704

Input:
D = {d1, d2 … dn} Data used.
k = {2, 3, 4 … n) Desired number of clusters

Output:
One set k cluster.

Steps 1:
Randomly select k centroid from D as the initial centroid (center of the initial cluster)

Step 2:
Determine each item in the cluster that has the closest cluster center; Calculate new averages for each cluster;

Step 3:
Repeat step 2 until the centroid cluster value does not change or until the maximum number of iterations is reached

Fig. 2. K-means clustering algorithm

The fourth stage is evaluation, which aims to test the performance of the proposed method. The opinion of experts is used as a reference. A confusion matrix is applied to calculate accuracy and precision [20]. At this stage, all cases are tested to decide the best stopwords set based on the k-means clustering technique.

The accuracy is obtained by dividing the number of only correct documents by all documents [21]. The true value means that the clustering results have the same class as the reference. On the other hand, precision is the comparison of true positive (TP) with the total of true positive and false positive (FP) [21]. TP means that when the result of clustering is a stopword and it is the same as the reference. FP means that the predicted result is stopwords while the reference is non-stopwords.

III. Results and Discussions

Table 4 shows the performance of the stopword list using k-means algorithm. The accuracy and precision represent the method performance by comparing the result with Javanese linguists' manual classification.

In Table 4, the highest accuracy is 78.2%, with 57.3% precision. The cluster supports this result with a value of k = 7. The result consists of 680 stopwords and 13704 non-stopwords, while the experts identify 3,224 and 11,160 of the same categories. The cluster can correctly indicate 11,030 of 14,384 words, which is dominated by non-stopwords category. Figure 3 shows the distribution of the word based on the first assumption that the first cluster is the non-stopwords.

As seen in Figure 3, experts recognize most words as non-stop words. The k-means wrongly categorized the non-stopwords into stopwords category (area within the grey line). On the other hand, the precision is 57.3% of the orange and gray areas, which means that most stopwords are categorized as non-stopwords. The lowest performance is when k=5. The accuracy is 25 %, and the precision 21.7%. Only 3089 is true stopwords, and 65 words are true non-stopwords.

The second assumption is then applied for comparison. Table 5 shows the result, assuming the first cluster is the stopwords, while the rest is non-stopwords.

Table 4. Stopword generator performance with the first asumption

k	Stopwords	Non-stopwords	Accuracy	Precision
3	49	14335	77.9%	100.0%
5	14184	200	21.9%	21.7%
7	680	13704	78.2%	57.3%
9	13281	1103	25.0%	21.5%
11	1500	12884	75.6%	40.6%
13	2145	12239	73.4%	36.1%
15	1750	12634	74.9%	39.2%

Table 5. Stopword generator performance with the second assumption

k	Stopwords	Non Stopwords	Accuracy	Precision
3	14335	49	22.07%	22.1%
5	200	14184	78.07%	67.5%
7	13704	680	21.7%	20.68%
9	1103	13281	74.9%	32.6%
11	12884	1500	24.3%	20.2%
13	12239	2145	26.5%	20.0%
15	12634	1750	25.02%	20.08%

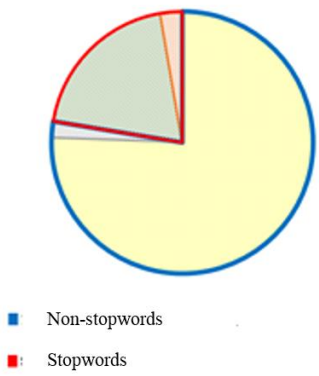

Fig. 3. Words distribution based on the first assumption

The best performance in Table 5 is when k = 5, where the accuracy value is 78.07% and the precision value is 67.5%. This case indicates 135 true stopwords and 11095 true non-stopwords. The obtained precision is 67.5%, which is equal to 135 of 200 stopwords.

The accuracy of both scenarios (Table 4 and Table 5) is similar. However, the precision of the best scenario (k=5) in Table 5 is higher than the best of Table 4 (k=3). It means that the performance second assumption is more promising than the first in recognizing the stopwords. Therefore, k-means locates stopwords in the first cluster while the-nonstopwords are in other clusters.

IV. Conlusion

K-means is applicable for Javanese stopwords list generation. The algorithm indicates the stopword location is in the first cluster of the words list. However, the current promising result is still possible to be improved. Further research should consider the balance of frequency distribution and the implementation of word stemming in the preprocessing. The use of more training data may balance the frequency, while the stemming may combine the unique words and unites the occurances of combined words.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] R. T. Lo, B. He, and I Ounis "Automatically Building a Stopword List for an Information Retrieval System," *J. Digit. Inf. Manag.* vol. 3, no. 1, pp. 3–8, 2005.

[2] J. Kaur, "A Systematic Review on Stopword Removal Algorithms," *Int. J. Future Revolut. Comp. Sci. Comm. Eng.* vol. 4, no. 4, pp. 207–210, 2018.

[3] J. Kaur and P.K. Buttar, 2018. "Stopwords Removal and its Algorithms Based on Different Methods". *International Journal of Advanced Research in Computer Science*, vol. 9, no. 5, pp. 81–88, Oct. 2018.

[4] S. Vijayarani, M. J. Ilamathi, and M. Nithya, "Preprocessing Techniques for Text Mining - An Overview," *Int. J. of Comp. Sci. Comm. Net.* vol. 5, no. 1, pp. 7–16, 2015.

[5] L. Dolamic and J. Savoy, "When Stopword Lists Make the Difference," *J. Am. Soc. Inf. Sci. Technol.*, vol. 61, no. 1, pp. 200–203, 2010.

[6] F. Zou, F. L. Wang, X. Deng, S. Han, and L. S. Wang, "Automatic Construction of Chinese Stop Word List," *Proc. 5th WSEAS Int. Conf. Appl. Comp. Sci.* 2006, pp. 1010–1015, 2006.

[7] J. K. Raulji, "Stop-Word Removal Algorithm and its Implementation for Sanskrit Language," *Int. J. Comp. Applica.* vol. 150, no. 2, pp. 15–17, 2016.

[8] R. M. Duwairi, "Arabic Sentiment Analysis using Supervised Classification," *Int. Conf. Futur. Internet Things Cloud*, pp. 579–583, 2014.

[9] R. M. Rakholia and J. R. Saini, "A Rule-Based Approach to Identify Stop Words for Gujarati Language," *Proc. 5th Int. Conf. Front. Intell. Comput. Theory Appl. Adv. Intell. Syst. Comput.*, p. 515, 2017.

[10] M. C. Kirana, N. P. Perkasa, M. Z. Lubis, and M. Fani, "Visualisasi Kualitas Penyebaran Informasi Gempa Bumi di Indonesia Menggunakan Twitter," *Journal of Applied Informatics and Computing*, vol. 3, no. 1, pp. 23–32, 2019.

[11] A. P. Wibawa, A. Nafalski, J. Tweedale, N. Murray, and A. E. Kadarisman, "Hybrid Machine Translation for Javanese Speech Levels," *Proc. 5th Int. Conf. Knowl. Smart Technol.*, pp. 64–69, 2013.

[12] S. Poedjosoedarmo, "Javanese Speech Levels," *Indonesia*, vol. 6, no. 6, pp. 54–81, 1968.

[13] A. P. Wibawa, A. Nafalski, A. E. Kadarisman, and W. F. Mahmudy, "Indonesian-to-Javanese Machine Translation," *Int. J. Innov. Manag. Tech.*, vol. 4, no. 4, pp. 451–454, 2013.

[14] S. V. S. Gunasekara and P. S. Haddela, "Context aware stopwords for Sinhala Text classification," *2018 Natl. Inf. Technol. Conf.*, pp. 1–6, 2018.

[15] T. M. Kodinariya, "Review on determining number of Cluster in K-Means Clustering," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, no. 6, pp. 90–95, 2013.

[16] K. A. A. Nazeer and M. P. Sebastian, "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm," *Proceedings of the World Congress on Engineering 2009*, vol. I, pp. 1–5, 2009.

[17] D. T. Pham, S. S. Dimov, and C. D. Nguyen, "Selection of K in K -means clustering," *Proc. Inst. Mech. Eng. Part C: J. Mech. Eng. Sci.*, vol. 219, no 1, May 2004, pp. 103–119, 2005.

[18] F. Leisch, "A toolbox for K -centroids cluster analysis," *Computational Statistics & Data Analysis*, vol. 51, no 2, pp. 526–544, 2006.

[19] N. Grozavu, Y. Bennani, and M. Lebbah, "From variable weighting to cluster characterization in topographic unsupervised learning," *Proc. Int. Jt. Conf. Neural Networks*, pp. 1005–1010, 2009.

[20] V. M. Patro and M. R. Patra, "Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy," *Transactions on Machine Learning and Artificial Intelligence*, vol. 2, no. 4, pp. 77–91, 2014.

[21] A. Mishra and S. Vishwakarma, "Analysis of TF-IDF Model and its Variant for Document Retrieval," *Int. Conf. Comput. Intell. Commun. Networks Anal.*, pp. 772–776, 2015.