J. Basic. Appl. Sci. Res., 7(9)1-9, 2017

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Improving the Performance of Repeated Character Preprocessing in Recognizing Words in the Indonesian Sentiment Classification

Fachrian Anugerah* and Arif Djunaidy

Faculty of Information Systems, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia

Received: May 11, 2017
Accepted: July 29, 2017

ABSTRACT

Relevant data is obtained through the pre-process by removing the noise so that the data to be processed in accordance with the needs. Noise removal is done by deleting repetitive characters, as the characters are often encountered in twitter data due to errors. This study aims to analyze the relevant results of the pre-process removal of repeated characters in the Indonesian sentiment classification. This is obtained by modifying the removal of characters repeatedly to calculate the similarity to determine the level of similarity with the dictionary. There are four types of characters repetitions were analyzed using repetitive character removal modifications to improve the quality of sentiment results using Support Vector Machines (SVM). Three ways of testing are done to analyze the deletion of repetitive characters by comparing: without, with, and modification of repetitive character removal. The test results show that the modifications performed show the best classification performance with an accuracy of 72.71%, whereas with the removal of repetitive characters produces a value of 71.25%, and without deletion of repeating characters produces a value of 70.67%. The modification performed has a significant role in the aspect of the meaning of the word, the best result of the character removal modification with a recognizable word of 59%. In addition, modifications made to improve performance at stemming and stop words. Improved stemming performance is evidenced by the number of words that can be recognized for 682 words. On the other hand improvement in performance of stop words is evidenced by 86 words that can be reduced so as to decrease the level of diversity of words that have the same meaning.

KEYWORDS: repeated characters, sentiment, classification, support vector machines

1. INTRODUCTION

Opinion mining is the process used to analyze conversations on a topic. Opinion mining has the purpose to classify comments into positive or negative opinions [1] and determine the emotions of a document [2]. The stages in opinion mining begin with data collection, then pre-processing on the data, and terminated by the classification process [1]. Each stage of the pre-process depends on the previous stage. The output from the previous stage will act as the next stepwise input. If the process at a certain stage does not work properly, it will affect the outcome of the next stage process [3].

This study discusses how to analyze the relevant results of the pre-process removal of repeated characters in the Indonesian sentiment classification. Challenges in using twitter data have a limited number of words on each twette of 140 characters, informal language use [4], and repetitive character writing [2]. The use of informal language is often incompatible with EYD (Improved Spelling), poor grammar so requires more processing [5]. Besides the documents in Indonesian language has its own uniqueness, because the words in the Indonesian language can change shape when getting affixes [6]. For that we need to do some stages to process opinion sentences that have a structure that is not standard in order to be processed properly. One of those stages is the elimination phase of repetitive characters. Removal of repetitive characters needs to be done because on twitter data often encountered the use of repetitive characters caused by errors of writing and user deliberate like "iyaaaa", "kapaan".

In a study done by [7], he did the removal of repetitive characters by removal the repetitive characters such as "apaaa" to "apa", "heloooooowwwww" to "helow". Similarly, studies conducted by [2], [3], [8] they also perform the stages by eliminating repetitive characters. But the problem occurs when processing a word that does have a loop on the default word like "sehingga", "tunggu", and "saat". The repetition character removal phase will remove the repetition "sehingga" becomes "sehinga", "tunggu" becomes "tungu", and "saat" becomes "sat" so the word will lose its meaning and can not be processed properly in the next stage. In addition, the problem occurs when the word gets imbuhan like the word "*pelanggannya*", which consists of the word "*pelanggan*" and then get affixed "*nya*" which resulted in the word experiencing repetition of characters "*g*" and "*n*" so that can not be recognized by the Indonesian dictionary . If the word is processed using repeated catastrophic removal, the characters "*g*" and "*n*" will be reduced to "*pelanganya*" which resulted in the word can not be recognized even though the "*nya*" has been removed.

Corresponding Author: Fachrian Anugerah, Faculty of Information Systems, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia.
e-mail: fachriananugerah.si@gmail.com

2. LITELATURE REVIEW

Data twitter has a challenge so it requires different processing. That's because there is a limit of the number of characters of 140 to send tweets. The restrictions leave users unable to express themselves so as to use informal language and remove some vowels, such as "story" to "stry" [9]. Other studies have shown lower results using twitter data than longer texts [4]. This could potentially occur spelling errors and unstructured sentences well. Accuracy depends on choosing relevant features [9].

To look for twitter data sentiment, it needs to be pre-processed. Preprocess to remove data that is not relevant to the research, because the data degrade the performance of classification [10]. Prior to the pre-processing, the collected data is grouped according to sentiment. [11] grouping by scoring each word with positive and negative sentiments. The existence of the score, then each document can be known dominance of sentiment.

The pre-processing stage is done emoticon conversion, uppercase identification, lower casing, URL extraction, username and hashtag conversion, punctuation deletion, stop words, keyword deletion, deletion of characters. Then tested using a variety of classification algorithms such as Naive Bayes, Naïve Bayes Multinominal, Complement Naive Bayes, DM NBteks, Bayesian Logistic Regression, SMO, SVM, J48, Random Forest, Lazy IBK. The best result is obtained by SMO algorithm with an accuracy value of 81.86% [11].

In other studies [2] modify when performing noise cleaning by removing signs of hashtags, and URLs, then case folding which converts characters into lowercase and characters other than the letter "a" - "z" are deleted. In addition there are exceptional characters that are not deleted because the characters are used in writing emoticons. Then perform the non-standard word conversion step by changing the non-standard word-shaped tokens into their default word form. Then do the classification by using SVM algorithm. The results showed that using all feature extraction phases obtained accuracy of 94.67% and 91.65% when not using feature extraction [2]. In a study conducted by [12], he modified the Iterative Computatuon Framework (ICF) by taking into account ratings and text and applying Genetic Algorithm (GA) to delve into the sentiment. From the results obtained show the addition of text variables can increase the value of accuracy obtained by 63% whereas if combined with GA get an accuracy of 74%.

In research [2], [3], [8], [13] performs repetitive character removal steps by eliminating repetitive characters. But deletion of characters used can not process words that do have repetitions on their default words. The repeating character removal phase will remove the repetition on the word, so the word will lose its meaning and can not be processed further. This study was conducted to improve the repetitive character removal phases, so that no word lost meaning after passing the phase of repetitive character removal.

2.1 Jaro Winkler

Jaro Winkler distance is a variant of Jaro distance metric which is an algorithm to measure the similarity between two strings. According to Cohen, that Jaro Winkler is intended to measure the similarity of short strings [14]. The result of this calculation yields a value of 0 - 1 where 0 denotes there is no similarity to the document and 1 denotes any similarity to document [15]. The Jaro Winkler algorithm is written in equations 1 and 2. In this equation, the parameters dj, m, | s1 |, | s2 |, t, dw, l, and p denote the Jaro distance, the same number of characters, the length of the string 1, 2, and half the number of transposition characters, Jaro Winkler distance value, same character length before found inequality and constant prefix weight (default = 0.1).

$$d_j = \frac{1}{3} \times \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) \quad (1)$$

$$d_w = d_j + \left(l \times p \left(1 - d_j \right) \right) \quad (2)$$

2.2 Support Vector Machines (SVM)

SVM is an algorithm used for classification using linear and nonlinear data. The algorithm by using a nonlinear mapping to convert training data to a higher dimension. SVM is a very accurate algorithm, due to their ability to model complex nonlinear [16]. The basic idea of the SVM algorithm is to find the optimal line with the maximum margin value. Begin by defining the equation of a dividing line written in equation 3. Where W is the weight of the vector (W1, W2, ..., Wn), n is the number of attributes, and b is the scalar.

$$W \times X + b = 0 \quad (3)$$

2.3 Removal of Repeated Characters

A repeating character removal stage is performed to repair a document from a word that has repeated characters caused by errors in writing. In studies [2], [3], [7], [8] they perform repetitive character removal steps by eliminating repetitive characters because in the twitter data, many found the writing of words that are not raw [5]. The input of this process is a word in which it undergoes repetition and does not experience repetition. If detected there is repetition, then the process of reduction so that the characters undergo repetition will be reduced to a single character like "*apaaaaa*" to "*apa*", "*cepaat*" to "*cepat*" as described in figure 1. The result of this repetitive character removal process is made Output from the repetitive character deletion stage.

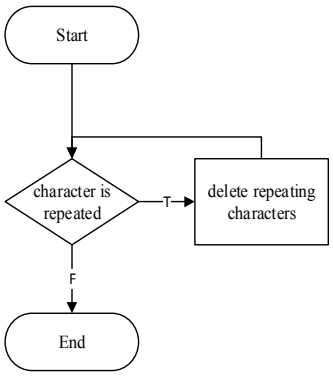

Figure 1: Repeated Character Removal Flow

2.4 Character Repeated Type

Based on the type of repetition, there are four types in the Indonesian text as described in Table 1. Each repetition has different characteristics requiring different treatments to be processed into recognizable words.

Table 1: Character Repeated Type.

Character Repeated Type	Examples of words
The standard word contains a repetition that encountered a characteristic repetition of more than one type	*"pelanggannya", "mengganggu", "penggunaan", "pembukaannya", "berlangganaan"*
The standard word contains a loop that does not experience a character repetition	*"maaf", "manfaat", "berlangganan", "hingga", "panggil", "saat"*
The standard word does not contain repetitions that have a character error	*"kecewaaa", "lagiii", "payaaaaaaaaaah", "terusss", "jauhhhh", "cobaan"*
The standard word does not contain a repetition that encounters a character repetition of more than one type	*"buseeettttttt", "hhilaaanngg", "masiiihhh", "kenyataannya", "pertanyaannya"*

2.5 Sentiment Grouping

The clustering stage of sentiment is used to classify the documents that have accumulated into a positive sentiment class or negative sentiment in accordance with the sentiments contained in the document. Bahrainian and Dengel use lexicon's centi strength to group by adding a "+1" sentiment value to the document if there is positive sentiment and reducing a sentence value of "-1" if nigative sentiment is found. If the document has a sentiment value above "0", then the document goes into the positive group and vice versa if the document has a sentiment value below "0", then the document goes into negative group [5].

3. METHODS

3.1. Data Preparation

Tweet data retrieval used a customer opinion about cellular telecommunication service provider in Indonesia, such as telkomsel, indosat, XL, smartfren. Tweet data containing sentiment collected 2400 tweets, number of positive tweet 800, negative 800 tweet, and neutral 800 tweets. From the data that has been collected there are 1163 contains word repetition with 57% contains recurrent words that are not recognized by standard words (682 words) and 43% contain recurrent words recognized by standard words (481 words).

3.2. Preprocess Text

The first stage of preprocessing is tokenizing. Documents broken down into multiple tokens, then noise cleaning to remove irrelevant documents such as URLs, symbols "@", "#". After that proceeded case folding (turning the token into a lowercase letter). In addition, removal of characters other than the letters a-z. Fourth stage, deletion of repeating characters who experience writing errors. The fifth stage, stop word removal to remove the words that often appear. Stage six, stemming is to convert a word into its basic word to get a token that is relevant to the research. The last stage, the word conversion is not standard by comparing the list of non-standard words. The result of this pre-process for input in the classification process.

3.3. Modification of repeated character removal

In this study, the development of the recurrent character removal process from Figure 1. The development of the process requires a standard Indonesian dictionary and the use of the Jaro Winkler algorithm in equation (2) which is used to measure the degree of similarity between two words. The standard word dictionary is used as system knowledge to know the various Indonesian raw words. If repetition of words is found, character deletion is not done by reducing to a single character that causes the word to lose meaning because it redundantly repeats characters. This is done to solve the first and second problem types in Table 1. The development flow of the character removal process is described in Figure 2. The process of repeating characters repeats begins by checking whether or not the word is repeated. If it is found the loop is continued to the second stage, it is a search of word similarity with the dictionary to determine the method of removal that will be done. If at this stage the word is successfully identified then the loop will be deleted, but if the word is not recognized by the dictionary it will proceed to the next stage of the repetition reduction stage. The word that passes through this stage is a word that has a repetition of more than one repetitive character such as "*sehinggga*" being "*sehingga*" and "*berlangganaan*" to be "*berlangganan*". Reducing a word that has only repetitions of similar characters over a repetition is done to avoid excessive repetition reduction which will create some raw words containing repetitive loss of meaning like the first and second problems in Table 1. If this stage is not done then words like "*sehingga*", The repetition of the character "*g*" will be reduced to a single character that causes the word "*sehinga*" to not be properly processed because of the loss of meaning. In stage four, the result of step three is used to find the key on the standard word dictionary by calculating the degree of similarity. This keyword search is used to find the similarity of words from the output of stage three with the word in the dictionary. This process uses the Jaro Winkler algorithm to get the weight value of each standard word in the dictionary. The next stage is done after each word in the dictionary has a weight, then selected a word that has a weight close to the value 1 that is used as a key in the next stage. The sixth stage is to calculate the weight of the possibility of character deletion into its single letter. As the word "*masiihh*" the removal of possible characters is "*masiih*" and "*masihh*" which are each counted in common with the keywords that have been set at stage five. From the similarity calculation process which yields the weight value of each of the possible character deletions, then weighs a value close to the value 1 used for input in stage five. This repetition will be completed until no weights are found that are greater than the previous maximum weight. After the loop completes, if the maximum weight equals 1 denotes that the foundation of a default word that is 100% similar to the word input in this process so that the output in this process is a word with weight 1. However, if the maximum weight is less than 1 indicates that it is not found with A standard word dictionary whose level of resemblance is 100%, so the output of this process is the word with maximum weight to reduce diversity in the same word.

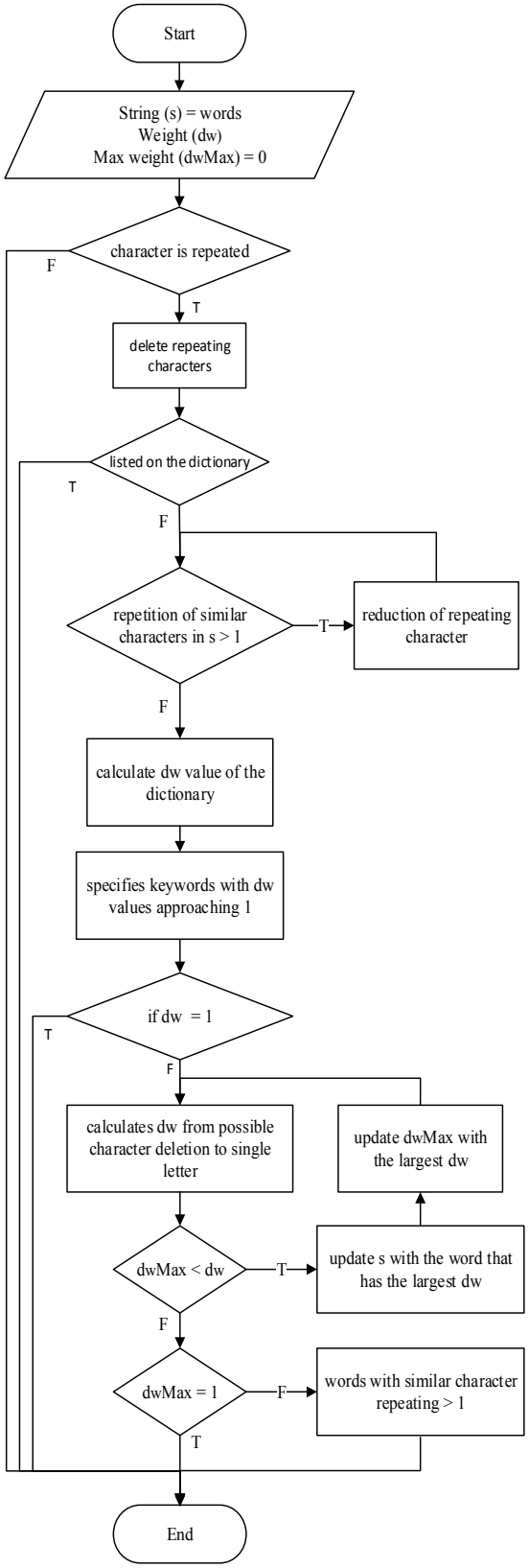

Figure 2: Modification of Repeated Character Removal Process

An example of applying development of repetitive character removal process to various types of loops using standard word dictionaries as in table 2 with "*berbangganaan*" input.

Table 2: Sample Word Dictionary

No	Word
1	*terus*
2	*masih*
3	*sayat*
4	*hingga*
5	*hinggap*
6	*sehingga*
7	*bangga*
8	*tangga*
9	*enggan*
10	*anyam*
11	*langgan*

The first stage, examined whether in the word experiencing repetition or not. The word "*berlangganaan*" experiences a repetition of the characters "g" and "a" so it qualifies and resumes at stage two.

The second stage, the repetition of the word will be removed to "*berlangganan*" and then match it with the Indonesian dictionary. Because the word "*berlangganan*" is not found in the dictionary so proceed to stage three.

The third stage, examined the number of repetitions contained in the word. The word "*berlangganaan*" experiences repetition of the character "g" as much as one repetition and the character "a" undergoes one repetition. Because no repetition of similar characters > 1, then resumed in stage four. The fourth stage, searching for the equivalence of the word "*berlangganaan*" with Indonesian dictionary. Each standard word in the dictionary is compared to the word "*berlangganaan*" and searched for similarity level using equation (2). The results of similarity calculations are presented in Table 3.

Table 3: Similarity Calculation of Words With Jaro Winkler's Algorithm

No	Word	Weight
1	*terus*	0.517
2	*masih*	0
3	*sayat*	0.425
4	*hingga*	0.533
5	*hinggap*	0.501
6	*sehingga*	0.569
7	*bangga*	0.838
8	*tangga*	0.739
9	*enggan*	0.533
10	*anyam*	0.425
11	*langgan*	0.846

The fifth stage, after every standard word in the dictionary is compared and has a weight value. Then determined the value of weight approaching with 1 as a key in the next stage is the word "*langgan*" to the standard word dictionary with a weight of 0.846. The sixth stage, calculated the weight of the possibility of elimination of characters "*berlangganaan*" into single letters that "*berlangnaan*" of 0.785 and "*berlangganan*" of 0861. Because the weight of the "*berlangganan*" character is 0.861 is greater than the key weight of 0.846, so it is recalculated at stage six with the word "*berlangganan*". After calculating the possibility of removal of characters "berlangganan" to the single letter that is "*berlanganan*" of 0.800. This stage does not experience a repetition because there is no possibility of deletion of characters and the weight of the deletion of the possibility of nonexisting characters exceeding the key weights, so this stage does not undergo repetition and proceed at a later stage. The seventh stage, because the maximum weight obtained < 1, so the output of the repetition phase is repeated is the output from stage 3 is the word "*berlangganan*".

3.4. Modification Test of Repetitive Character Removal

This repetitive character removal modification test uses two kinds of data, the trial data and real data (twitter data). This trial is done by modifying the removal of repetitive characters and comparing using and not using stop word. This modification done in three ways: providing word input that has no repetition, a word containing repetition, and a word that has errors in the loop. The first test gives 40 non-repetitive words like "*balas*", "*jawab*", and "*cepat*". The second test provides 40 words containing repetitions such as "*mengganggu*" and "*hingga*" (Table 1). The third test, giving 40 words input that experienced errors in the repetition such as "*kecewaa*" and "*kenyataannya*" (Table 1).

4. RESULTS AND DISCUSSION

4.1. The result of Trial of Repetitive Character Removal Modification

This test is to know the result after modification. The results of first test show that each process can recognize 40 non-recurring words, so the whole word can be well recognized. The results of second test, indicating that the process without the removal of recurring characters can recognize 40 words. However, this process is incapable of processing words containing repetitions of characters such as "*anggap*" to be "*angap*", "*gangguan*" to "*ganguan*," and "*perubahannya*" to "*perubahanya*" that causes the word to lose meaning and can not be processed by good. Unlike processes that use repetitive character removal modifications, the results obtained are satisfactory by recognizing 40 words. Modifications made to distinguish words that have with that do not have a repetition of characters.

The result of third test, indicating that the process without removal of repetitive characters can not process words that have repeated characters, such as "*toolooooong*", "*cepaaatt*", and "*sanggguup*". This is because the word contains a repetition of characters, so it could not be recognized. There is an increase in the number of words (18 words) that are recognized when using the recurring character removal process and the word has no repetition of characters. If the character's repetition is removed, then the word can be recognized well. Significant improvements occur using repetitive character removal modifications. Repetitions that occur on words can be processed, both words containing and not containing repetitions can be recognized. Number of words that can be recognized as many as 40 words, because before character deletion, the repetition in the base word is evaluated. If the base word has no repetition, then the character repetition will be deleted until no repetition occurs. But if the base word has a repetition, then the deletion of repeated characters will leave one repetition and continue the search process similarities using Jaro Winkler Algorithm. The word will be searched for its resemblance by searching for possible deletion of characters over and over until it finds a weight equal to 1 which means to have a resemblance to a 100% Indonesian dictionary. But if you do not find any resemblance, then the result of removal leaves one repetition to minimize the word diversity.

Table 4 shows that repetitive character removal modifications can process different types of repetitions. Unmoded repeating character repetitions can be used on non-repetitive words, so deleting characters makes the word recognizable.

Table 4: Number of Recognizable Words in a Repeating Character Removal Scenario

	Without repetition (word)	With repetition (word)	With repetition error (word)
Without removal	40	40	0
With removal [7]	40	0	18
With modification	40	40	40

After testing using three ways (Table 4), then tested using real data (twitter data). The test results are shown in Table 5. Table 5 shows that the process without repetitive character deletion can recognize 43% of 1163 words experiencing repetition of characters, while 57% are not recognizable. However there is a decrease in the number of words that are recognized when using repetitive character deletion. A total of 1163 words that have repetition of repetitive character deletion can recognize 19% (224 words). The best results using repetitive character removal modifications can recognize 59% (682 words).

Table 5: Comparison of Repeated Character Performance

	Recognizable stemming (word)	Not recognizable stemming (word)
Without removal	501	662
With removal [7]	224	939
With modification	682	481

Process without deleting recurring characters that recognize 43% because there are words that have repetitive errors like "*yaaa*" should the character repeat be deleted to "*ya*", so the word can not be recognized. The recurring character removal process that recognizes 19% is because there are words that have errors in the loop like "*gangguan*" should the loop should not be deleted so that the word can not be recognized. In the process of modifying the deletion of recurring characters that recognize 59% is because 41% of them are words that have a repetition of characters in English, abbreviations, or inappropriate EYD so it can not be processed properly.

The test results are known to eliminate repetitive characters resulting in the best classification performance with modification (Table 6). This type of test produces an accuracy of 72.71%; Precision 73.2%; Recall 72.7%; And f-measure 72.9%. The lowest yield is obtained by without deletion of repeating characters with an accuracy of 70.67%; 71% precision; 70.7% recall; and f-measure 70.8%.

Table 6: Comparison of Repeating Character Removal

	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
Without removal	70.67	71	70.7	70.8
With removal [7]	71.25	72.1	71.3	71.6
With modification	72.71	73.2	72.7	72.9

The use of repetitive character removal modifications is better than without repetitive deletion, but no for deletion of recurring characters because the modification can not work optimally. Unoptimized modifications due to deletion of words that have been fixed by stop words, so repetitive character fixes do not look prominent. However, the results can be concluded that the modification significantly affects the meaning of the word. The best results are obtained from the character removal modification by recognizing the word 59%. That's because the repetition of the data also contains languages other than Indonesian, abbreviations, and words that do not match the EYD, so the character removal modification can not process all words.

Modifications can process that the words have the same suffix as the beginning of the affix as the word "*pelanggannya*", consisting of the word "*pelanggan*" and then affixed "*nya*". If no modification is done on the character deletion the word "*pelanggannya*" will change to "*pelanganya*" causes the word can not be processed by stemmer. If repetitive character deletion is not performed. In addition, modifications also improve performance in stemming and stop words.

Improved stemming performance is presented in Table 3. Modified character deletion removal recognizes 682 words, whereas repetitive character deletion recognizes 224 words. Performance improvement of stop words there were 86 words, that can be reduced when using repetitive character removal modifications. So decreasing the words diversity that have the same meaning.

4.2. Comparative Trial Stop Words

The comparison test of stop words indicates that when using the word stop, the results obtained are lower when compared to the non stop words (Table 7). This is because there are 86 words that have been fixed but removed with word stop.

Table 7: Comparison accuracy of Stop Word

	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
With stop words	72.71	73.2	72.7	72.9
Without stop words	74.46	74.9	74.5	74.6

5. CONCLUSION

Repetitive character removal mods can handle repeated words without changing their meaning; The results of testing the twitter data indicate that the modifications made to produce the best classification performance (accuracy value 74.46%); The best results using repetitive character removal modifications can recognize 59% recurring words; And repetitive character removal modification can improve performance at stemming stage by recognize 682 words, and 86 words can be reduced by stop words.

6. REFRENCES

[1] K. Amarouche, H. Benbrahim, and I. Kassou, “Product Opinion Mining for Competitive Intelligence,” Procedia Computer Science, vol. 73, Mohammed 5 University, Rabat, Morocco, 2015, 358–365.

[2] A. A. Arifiyanti, “Ekstrasi Fitur Pada Konten Jejaring Sosial Twitter Berbahasa Indonesia Dalam Peningkatan Kinerja Klasifikasi Sentimen,” Intitut Teknologi Sepuluh Nopember Surabaya, 2015.

[3] A. G. Shirbhate and S. N. Deshmukh, “Feature Extraction for Sentiment Classification on Twitter Data”, International Journal of Science and Research, University Aurangabad, Maharashtra, India, 2016.

[4] M. Bouazizi and T. Ohtsuki, “Opinion mining in Twitter: How to make use of sarcasm to enhance sentiment analysis,” International Conference on Advances in Social Networks Analysis and Mining, Keio University, Yokohama, Japan, 2015, pp. 1594–1597.

[5] S. A. Bahrainian and A. Dengel, “Sentiment Analysis and Summarization of Twitter Data,” International Conference on

Computational Science and Engineering, Univ. Of Kaiserslautern, Kaiserslautern, Germany, 2013.

[6] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," Universiteit van Amsterdam The Netherlands, Amsterdam, 2003.

[7] M. Illecker, "Real-time Twitter Sentiment Classification based on Apache Storm", Innsbruck University, 2015.

[8] A. Amolik, N. Jivane, M. Bhandari, and M. Venkatesan, "Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques", International Journal of Engineering and Technology, VIT University , India, 2016.

[9] S. Keretna, A. Hossny, and D. Creighton, "Recognize User Identity in Twitter Social Networks via Text Mining", 2013 IEEE International Conference on Systems, Deakin University, Australia, 2013.

[10] K. Aurangzeb, B. Baharum, H. Lam, and khan Khairullah, "A Review of Machine Learning Algorithms for Text Documents Classification", Journal of Advances In Information Technology, Universiti Teknologi Petronas, 2010.

[11] B. Gokulakrishnan, P. Priyanthan, T. Ragavan, N. Prasath, and A. Perera, "Opinion mining and sentiment analysis on a Twitter data stream", International Conference on Advances in ICT for Emerging Regions (ICTer2012), University of Moratuwa, Sri Lanka, 2012, pp. 182–188.

[12] E. Dyar Wahyuni and A. Djunaidy, "Fake Review Detection from a Product Review Using Modifed Method of Iterative Computation Framework," MATEC Web of Conferences, ITS, East Java, Indonesia, 2015.

[13] Y. Garg, "yogeshg/Twitter-Sentiment," GitHub, 2014. [Online]. Available: https://github.com/yogeshg/Twitter-Sentiment. [Accessed: 21-Jan-2017].

[14] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A Comparison of String Distance Metrics for Name-Matching Tasks", Carnegie Mellon University, Pittsburgh, 2003.

[15] K. Dreßler and A.-C. N. Ngomo, "Time-Efficient Execution of Bounded Jaro-Winkler Distances", University of Leipzig, Germany, 2014.

[16] J. Han and M. Kamber, Data Mining Concept and Techniques, Second edition. 2000.