IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.13, No.1, January 2019, pp. 53~62
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.41136

Sarcasm Detection For Sentiment Analysis in Indonesian Tweets

Yessi Yunitasari[1], Aina Musdholifah[2], Anny Kartika Sari*[3]
[1]Master Program of Computer Science; FMIPA UGM, Yogyakarta
[2,3]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta
e-mail: *[1]**yessi.yunitasari@mail.ugm.ac.id,** [2]aina_m@ugm.ac.id, ***[3]a_kartikasari@ugm.ac.id**

Abstrak

Salah satu jenis media sosial yang banyak digunakan saat ini adalah Twitter. Percakapan di tweet dapat diklasifikasikan berdasarkan sentimennya. Adanya sarkasme yang terkandung di dalam suatu tweet kadang-kadang mengakibatkan penentuan sentimen suatu tweet menjadi tidak tepat karena sarkasme sulit dianalisis secara otomatis, bahkan oleh manusia sekalipun. Oleh karena itu, deteksi sarkasme perlu dilakukan, yang diharapkan dapat meningkatkan hasil analisis sentimen. Pengaruh deteksi sarkasme pada analisis sentimen dapat dilihat dari sisi akurasi, presisi, dan recall. Pada penelitian ini, deteksi sarkasme dilakukan untuk tweet berbahasa Indonesia. Metode untuk ekstraksi fitur deteksi sarkasme menggunakan unigram dan 4 set fitur Boazizi yang terdiri dari fitur sentiment-relate, fitur punctuation-relate, fitur lexical and syntactic, dan fitur top word. Proses deteksi sarkasme menggunakan algoritme Random Fores. Ekstraksi fitur untuk analisis sentimen menggunakan TF-IDF dan klasifikasinya menggunakan algoritme Naïve Bayes. Hasil pengujian analisis sentimen dengan deteksi sarkasme menunjukkan peningkatan rata-rata akurasinya sebesar 5,49% dengan nilai akurasi sebesar 80,4%, presisi sebesar 83,2%, dan recall sebesar 91,3%.

Kata kunci— *Naïve bayes, sarcasm, tweet, sentiment analysis, random forest*

Abstract

Twitter is one of the social medias that are widely used at the moment. Tweet conversations can be classified according to their sentiments. The existence of sarcasm contained in a tweet sometimes causes incorrect determination of the tweet's sentiment because sarcasm is difficult to analyze automatically, even by humans. Hence, sarcasm detection needs to be conducted, which is expected to improve the results of sentiment analysis. The effect of sarcasm detection on sentiment analysis can be seen in terms of accuracy, precision and recall. In this paper, detection of sarcasm is applied to Indonesian tweets. The feature extraction of sarcasm detection uses unigram and 4 Boazizi feature sets which consist of sentiment-relate features, punctuation-relate features, lexical and syntactic features, and top word features. Detection of sarcasm uses the Random Forest algorithm. The feature extraction of sentiment analysis uses TF-IDF, while the classification uses Naïve Bayes algorithm. The evaluation shows that sentiment analysis with sarcasm detection improves the accuracy of sentiment analysis about 5.49%. The accuracy of the model is 80.4%, while the precision is 83.2%, and the recall is 91.3%.

Keywords— *Naïve Bayes, sarcasm, tweet, sentiment analysis, random forest*

Received November 21th,2018; Revised January 23th, 2019; Accepted January 31th, 2019

1. INTRODUCTION

The use of social media such as Facebook, Twitter, Google Plus, etc. in everyday life changes communication patterns [1]. One type of social media that is widely used today is Twitter. Twitter allows users to write and read messages, commonly called as tweets. Twitter limits the number of characters in one tweet to 140 characters. Every day, millions of tweets are written by more than 285 million active Twitter users [2].

Sentiment analysis is one branch of text mining research that performs classification to text documents, including tweets. Tweets can be classified based on their sentiments, namely positive and negative sentiments. Several methods for sentiment analysis have been used in previous studies, including Naïve Bayes, SVM (Support Vector Machine), and KNN (K-nearest neighbor) algorithms. Among them, Naïve Bayes method is commonly used because it is simple and easy to be implemented to various situations [3].

It is often that the sentiment of a tweet cannot be determined correctly when the tweet contains sarcasm. Sarcasm is a special form of irony that happens when someone conveys implicit information, usually having the opposite meaning of what is said [2]. Sarcasm is difficult to be analyzed automatically, even by humans [4]. Sentiment analysis detects polarity based on the value of each word, while sarcasm detection also considers the intonation or facial movements when the person speaks. Unfortunately, there is no information about intonations or facial movements. As a result, the detection of sarcasm is still considered as a difficult problem in sentiment analysis [5], including sentiment analysis of tweets.

Several previous studies in the field of sarcasm detection have been done, including investigating the effects of sarcasm on sentiment analysis [2, 4]. One of the methods that can be used for sarcasm detection is random forest. Random forest is an ensemble method that consists of several decision trees as a classifier [6]. Random forest is suitable for binary dataset because it uses decision tree as base learner and it is fairly good to classify data with binary types [7].

This study focuses on developing a model for sentiment analysis by considering the possibility of sarcasm content in a tweet. Naïve Bayes is applied as the algorithm to analyze the sentiment of tweets, while random forest is used to detect sarcasm. The inclusion of sarcasm detection in sentiment analysis is expected to improve the results of sentiment analysis.

This paper is organized as follows. Section 2 describes the proposed method, while Section 3 elaborates the results of the evaluation. The conclusion and future work is presented in Section 4.

2. METHODS

In this section, the proposed method is explained in detail. This includes the data that is used in this research, the model to detect sarcasm, and the sentiment analysis model.

2.1 Data Collection

The data used in this research is Indonesian tweets. The tweets were collected from 13rd January to 15th January, 2018, from global stream data of twitter using geolocation filter and some hashtag keywords. The tweets came from almost all locations in Indonesia. Several viral hashtags at that time that highly potential to contain sarcasm, such as “terorismebukanislam” were used as filters. There were 3000 tweets obtained from the data collection process. Manual POS tagging process was conducted, and Indonesian negative-positive words are manually listed.

2.2 Sarcasm Detection Model

Figure 1 shows the flowchart of the sarcasm detection process. There are four activities that must be done in the process, i.e. pre-processing, feature extraction, sarcasm detection and evaluation (testing).

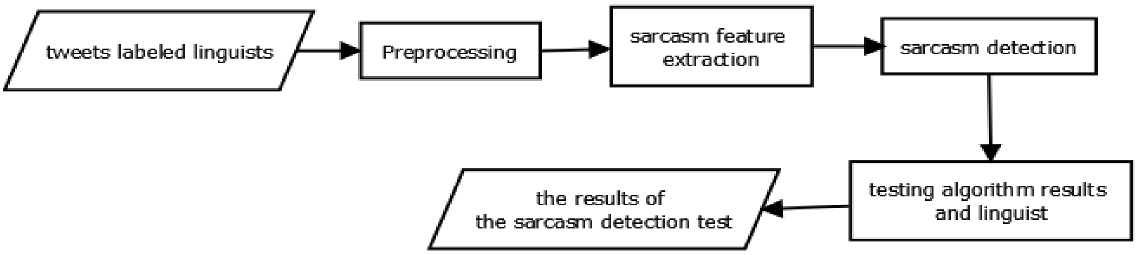
tweets labeled linguists
Preprocessing
sarcasm feature extraction
sarcasm detection
testing algorithm results and linguist
the results of the sarcasm detection test

Figure 1 Flowchart of sarcasm detection process

Detailed explanation of the sarcasm detection is as follows.

1. Manual labelling by domain expert
   An Indonesian linguist labelled each tweet with either ‘sarcasm’ or ‘not sarcasm. The result of this process is collection of tweets that have been labelled.
2. Preprocessing process, consisting of the following sub-processes:
   a. Conversion of *emoticon* to word
   b. Conversion of *number* to word
   c. *Slang word fixing*
   d. *Translation* of non-Indonesian words to Indonesian
3. Feature extraction
   The features used are mostly adapted from [2]. There are several types of features used in this model, as explained below:
   a. *Unigram*
   To extract this feature, each tweet is split into words.
   b. S*entiment-Related Features*
   Sentiment related features consist of special weight *ρ(t)* as shown in Equation 1, the number of positive emoticons, the number of negative emoticons, the number of sarcasm emoticons, the number of positive hashtags, the number of negative hashtags, the number of word and word contrast, the number of hashtag and hashtag contrast, the number of word and hashtag contrast, and the number of word and emoticon contrast [2].

$$\rho(t) = \frac{(\delta.PW+pw)-(\delta.NW+nw)}{(\delta.PW+pw)+(\delta.NW+nw)} \tag{1}$$

   In Equation (1), *t* refers to the tweet, *δ* is 3, i.e. the weight of high emotional word, *PW* is the number of positive words with emotional value, *NW* is the number of negative words with emotional value, *pw* is the number of positive words, and *nw* is the number of negative words.
   c. *Punctuation-related Features*
   The punctuation-related feature is used to detect sarcasm using a form of expression. For each tweet, the number of exclamation marks, question marks, capital letters, and word quotes will be counted. Moreover, the use of the same letter more than twice in a word is also counted.
   d. *Lexical and Syntactic Features*
   Lexical and syntactic features are used to detect sarcasm with ambiguous sentence in order to hide the original intent of the sentence. Three components are counted for this feature, as follows:
      i. The number of laughter
      Examples of laughter often found in Indonesian tweets are hehe, haha, hihi, hoho, wkwk, wkowko, wkewke, and LOL [8].
      ii. The number of exclamation words

Examples of exclamation words are ah, ih, uh, eh, oh, aw, iw, uw, ew, ow, waw, wow, wah, etc.

iii. The number of words that are rarely used
Words that are rarely used are determined from the tweet collection, i.e. words that only appear once.

e. *Top word Feature*, i.e. 100 words that appear most often in certain classes.

4. Sarcasm Detection
The classification process uses random forest classifier. The Flowchart process of how building a decision tree in the Random Forest can be seen in Figure 2.

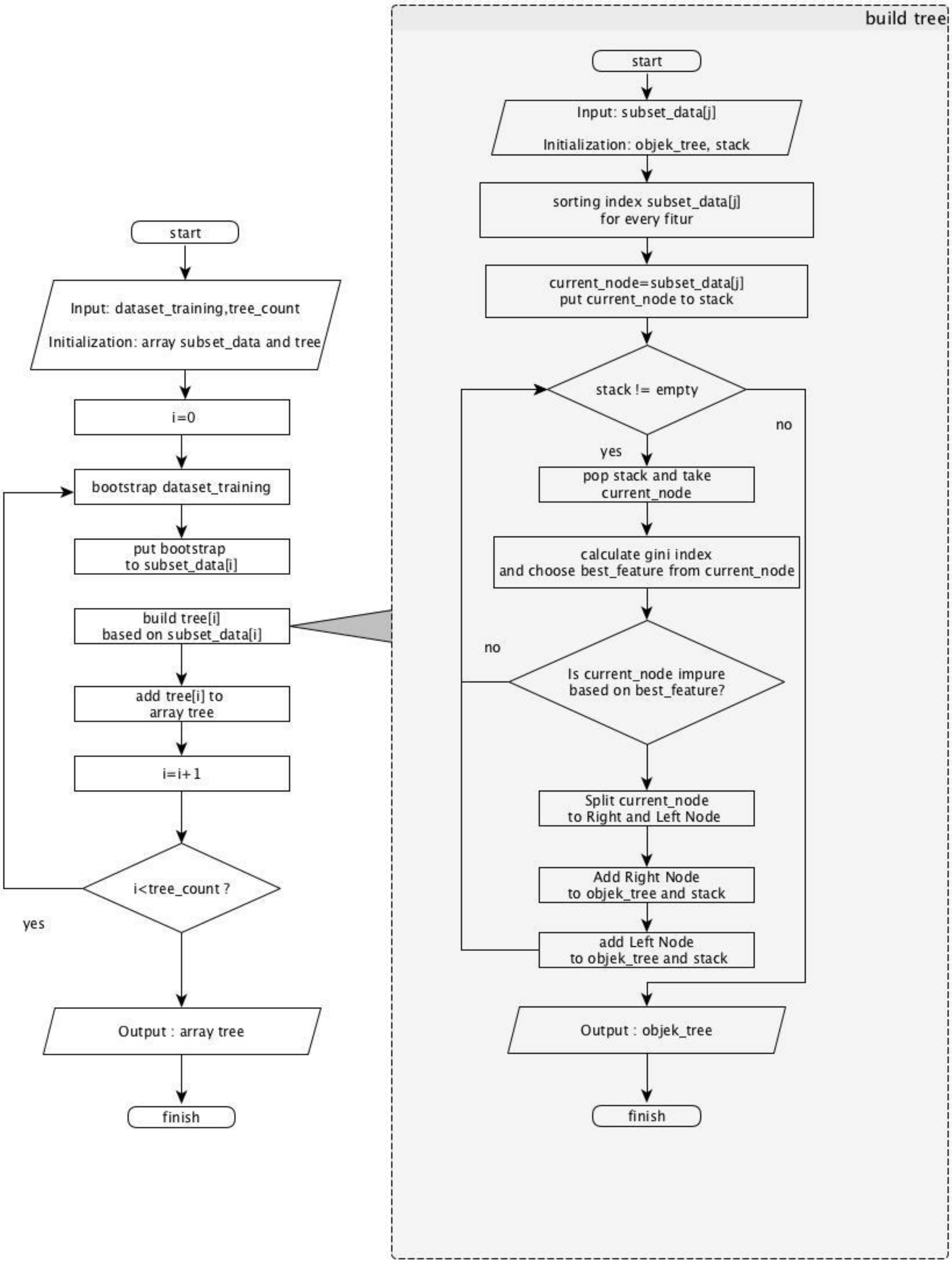
build tree
start
Input: dataset_training,tree_count
Initialization: array subset_data and tree
i=0
bootstrap dataset_training
put bootstrap to subset_data[i]
build tree[i] based on subset_data[i]
add tree[i] to array tree
i=i+1
i<tree_count ?
yes
Output : array tree
finish
start
Input: subset_data[j]
Initialization: objek_tree, stack
sorting index subset_data[j] for every fitur
current_node=subset_data[j] put current_node to stack
stack != empty
yes
no
pop stack and take current_node
calculate gini index and choose best_feature from current_node
no
Is current_node impure based on best_feature?
Split current_node to Right and Left Node
Add Right Node to objek_tree and stack
add Left Node to objek_tree and stack
Output : objek_tree
finish

Figure 2 Flowchart of Random Forest Classifier

The process of Random Forest classifier starts with receiving the training data as input. Initialization of some required variables, such as the number of trees and the array subset_data, are performed. Iteration is conducted based on variable tree_count, in which in each iteration bootstrap is done. Bootstrap is the random selection of a subset of data obtained from the training data, and the data will be used as a data source for building trees.

The process of building the tree starts with receiving the input from bootstrap data that has been created previously. The construction of the tree uses the depth first search method; hence, it takes structure data in the form of _tree and stack objects. Before the algorithm starts the iteration, the bootstrap data is sorted for each feature. Iteration starts with checking the stack, the data in the top stack represents the current node. The gini index is then calculated on the node using Equation (2) and then the best feature is selected. In the equation, T refers to data set, N is the number of classes, p_j is the relative frequency of class j in T.

$$gini(T) = 1 - \sum_{j=1}^{N} {p_j}^2 \tag{2}$$

Based on the best feature, the impurity of the node is then checked. Impurity is a condition where the current node has heterogeneous classes (there are more than 1 types of class). If the current node is in an impure state, then the node is split into the right and the left nodes. The right and the left nodes are inserted into the stack for the next iteration process. The final result of this process is a tree. For the next iteration, the object will be put back into the main iteration to generate a collection of trees stored in the array of trees a.

5. The algorithm is tested using the testing data and then compared with the result of manual labelling by the Indonesian linguist. Testing is done to calculate the accuracy, precision and recall of the proposed sarcasm detection model.

2.3 Improved Sentiment Analysis Model

Sentiment analysis model proposed in this paper utilizes the result of sarcasm detection. It is expected that the sarcasm detection will improve the result of sentiment analysis. The methodology is simple: if a tweet is classified as “positive” in sentiment analysis, but it is detected to contain “sarcasm” in sarcasm detection, the sentiment of the tweet is changed to “negative”. The complete process of the improved sentiment analysis model is shown by Figure 3.

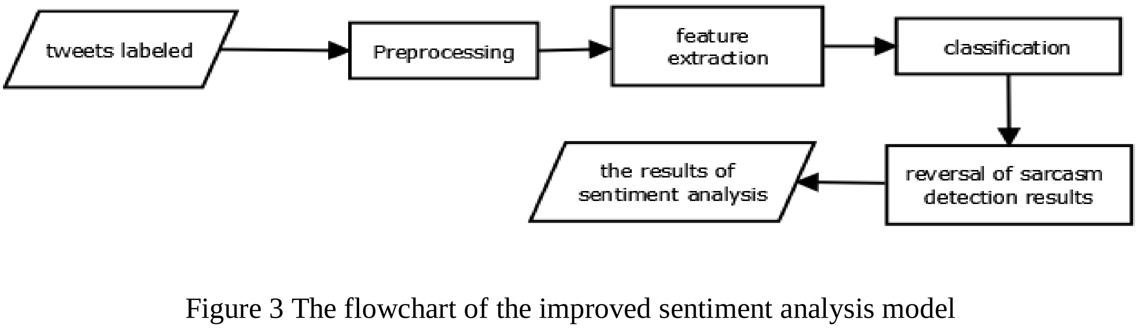
tweets labeled
Preprocessing
feature extraction
classification
reversal of sarcasm detection results
the results of sentiment analysis

Figure 3 The flowchart of the improved sentiment analysis model

The detailed process of sarcasm detection model is as follows.

1. Manual labelling
Each tweet is labelled as positive or negative by the Indonesian linguist.

2. Preprocessing, consisting of the following sub-processes:
   a. Case folding: standardize all letters into lowercase letters.
   b. Stop word removal: removing repetitive (unimportant) words.
   c. Emoticon conversion: converting emoticons into strings (words).
   d. Conversion of number characters to letters in a word, for example the word “p3rgi” is converted into “pergi”.
   e. Slang word conversion: replacing slang words into Indonesian standard words.
   f. Translation of English words into Indonesian words.
   g. Stemming: removing prefix, suffix and infix from word.
3. Feature Extraction, consisting of two features as follows:
   a. *Unigram*, i.e. the same feature as used in the sarcasm detection process.
   b. *TF-IDF* (Term Frequency-Inverse Document Frequency)
      Term Frequency (TF) is the number of occurrences of a specified term in a document, while Document Frequency is the number of documents containing the term. TF-IDF is used in this model for term weighting. To calculate the IDF of a term, Equation (3) is used [9]. In the equation, idf_t is the inverse document frequency of term *t*, *N* is the total number of documents (tweets), and df_t is the number of documents containing term t.

      $$idf_t = log\frac{N}{df_t} \tag{3}$$

      TF-IDF is calculated using Equation (4) by multiplying the term frequency value with the inverse document frequency value of the term [9]. In the equation, $tf.idf_{t,d}$ is the weight of term *t* found in document (tweet) *d*, while $tf_{\ t,d}$ is the number of occurrences of term t in document *d*.

      $$tf.idf_{t,d} = tf_{\ t,d} \times idf_{\ t} \tag{4}$$

4. Classification to determine the sentiment of tweets.
   The classification process using the Multinomial Naïve Bayes Classifier as specified in Equation (4) [10].

   $$P(c|d) \propto P(c) \prod_{\mathbf{1 \le k \le n_d}} \boldsymbol{P} \, (\boldsymbol{t_k} \mid \boldsymbol{c}) \tag{5}$$

   $P(t_k \mid c)$ in Equation (5) is the probability that term t_k is contained in class *c*, *P(c)* is the prior probability of a document classified as class *c*, n_d is the number of tokens in class *d*. The chosen class is obtained by maximizing Posteriori (MAP) of class *c*.
5. Reversal of sentiment based on the result of sarcasm detection
   The sentiment of a tweet is reversed when the tweet is classified as “positive”, but it contains sarcasm. For example, the tweet "Ayo dongg bersahabat 1 minggu ini ajaa!! Plisss 😭😟" is originally classified to have positive sentiment. However, since it is determined to contain sarcasm in the sarcasm detection process, the tweet is then classified as “negative”.
6. Evaluation of the model
   The test is conducted using k-fold cross validation approach. There are 2 models to be compared: sentiment analysis with detection of sarcasm, and sentiment analysis without detection of sarcasm.

3. RESULTS AND DISCUSSION

3.1 Evaluation of Sarcasm Detection Model using k-fold Cross Validation

Experiments have been performed to check if the model is stable to detect the sarcasm. Therefore, k-fold cross validation is used to test the model. The experiments were conducted several times with k = 5, k = 10, and k = 15. Table 1 shows the average of accuracy, precision, and recall for each value of k. The accuracy of the model is always above 70% in every fold;

hence, the model can be considered as stable to be used to improve the performance of sentiment analysis.

Table 1 Testing result of sarcasm detection model using k-fold cross validation

k	Accuracy Avg.	Precision Avg.	Recall Avg.
5	0.721	0.501	0.333
10	0.718	0.489	0.356
15	0.725	0.507	0.362

3.2 Evaluation of Sentiment Analysis with Sarcasm Detection

To evaluate the influence of sarcasm detection in the sentiment analysis of Indonesian tweets, testing is conducted to sentiment analysis with sarcasm detection and sentiment analysis without sarcasm detection. Table 2 shows the average value of accuracy, precision, and recall for every model and for every value of k. It can be seen that sarcasm detection improves the accuracy and precision of sentiment analysis. The improvement of accuracy and precision occurs in every value of k. However, the recall values decrease in each value of k. This means that misclassification of sentiment analysis increases. One of the reasons is that sarcasm does not always cause the sentiment to be negative. In this research, sarcasm is always identified as negative meaning, while in reality, it is possible to have positive meaning.

Table 2 The influence of sarcasm detection for sentiment analysis improvement

Type of Testing	Accuracy Avg.	Precision Avg.	Recall Avg.
k-5 Sentiment analysis model	0.739	0.737	0.988
k-5 Sentimen analysis + Sarcasm model	0.818	0.845	0.915
k-10 Sentiment analysis model	0.761	0.763	0.981
k-10 Sentimen analysis + Sarcasm model	0.797	0.823	0.916
k-15 Sentiment analysis model	0.748	0.756	0.974
k-15 Sentimen analysis + Sarcasm model	0.798	0.828	0.908

3.3 Evaluation of Features to Sentiment Analysis Improvement

In this model, other than the three types of features adapted from [2], i.e. punctuation related featuress, sentiment related featuress, and lexical and syntactic features, top word feature is also proposed. Testing has been conducted to evaluate the performance of sarcasm detection using one of combination of the features. Table 3 shows the average of accuracy, precision, and recall of each model using 5-fold cross validation test. It shows that sentiment related features have the highest accuracy of 72.5%. The reason is that sentiment related features are highly influenced by hashtag and emoticon occurences, in which 8 of the 10 features highly depend on them. Most previous research in sarcasm detection use hashtag and emoticon occurrences as one of the features. This proves that the features are essential to sarcasm detection.

Table 3 Testing result of sarcasm detection model using different types of features

Sarcasm Detection Model	Accuracy Avg.	Precision Avg.	Recall Avg.
All 4 features	0.722	0.499	0.350
Punctuation related features	0.720	0.499	0.201
Sentiment related features	0.725	0.512	0.246
Lexical and syntactic features	0.718	0.270	0.008
Top word + punctuation related features	0.718	0.494	0.237

Sarcasm Detection Model	Accuracy Avg.	Precision Avg.	Recall Avg.
Top word + sentiment related feature	0.723	0.492	0.213
top word + lexical and syntactic features	0.718	0.413	0.023

To evaluate the influence of each sarcasm detection feature for sentiment analysis improvement, testing is conducted for sentiment analysis model with sarcasm detection containing one or more combinations of features. Table 4 shows the result of the test. Even though sentiment related features have the highest influence in sarcasm detection performance, the performance of sentiment analysis model improves the most when using sarcasm detection with all features. It reaches 80.4% of accuracy and outperforms sarcasm detection using other combinations of features. This is caused by several reasons. The first reason is that, as previously shown in Table 1, the sarcasm detection model using all features is stable as the accuracy with of k-fold cross valiation with different value of k always reaches 70%. The second reason is that, as shown in Table 2, the model improves the precision of sentiment analysis model.

Tabel 4 Testing result of sarcasm detection model using different types of features

Sentiment Analysis Model	Accuracy Avg.	Precision Avg.	Recall Avg.
Without sarcasm detection	0.749	0.752	0.981
With sarcasm detection using all features	0.804	0.832	0.913
With sarcasm detection using punctuation related features	0.756	0.776	0.929
With sarcasm detection using sentiment related feature	0.744	0.749	0.969
With sarcasm detection using lexical and syntactic features	0.737	0.739	0.981
With sarcasm detection using top word + punctuation related features	0.769	0.789	0.928
With sarcasm detection using top word + sentiment related feature	0.750	0.767	0.937
With sarcasm detection using top word + lexical and syntactic features	0.741	0.743	0.979

3.4 Evaluation of Slang Word Fixing and English Word Translation Processes

Slang words and English words occur a lot of time in Indonesian tweets; hence, the processes of slang word fixing and word translation are needed. To evaluate the influence of the processes in both sarcasm detection and sentiment analysis models, testing have been conducted. Table 5 shows the influence of the processes to the performance of the models.

From the table, English translation improves the performance of either the sarcasm detection model or the sentiment analysis model. This happens because a lot of Indonesian people use both Indonesian and English words in their tweets, that make it difficult to determine the polarity of the words. Translation also improves the accuracy in sentiment analysis model because it highly depends on bag of words approach. On the other hand, the slang word fixing process decreases the accuracy of the sentiment analysis model. This is because the quantity of bag of words decreases with slang word fixing process. In sentiment analysis model these words are used as features. The occurrence of slang words may increase the quantity of bag of words, thus give datasets more variation and in some cases could increase the accuracy.

Tabel 5 The improvement of models with slang word fixing and English word translation processes

Type of Model	Accuracy Avg.	Precision Avg.	Recall Avg.
Sarcasm detection with English translation	0.722	0.500	0.351
Sarcasm detection without English translation	0.705	0.449	0.293
Sarcasm detection with slang word fixing	0.722	0.500	0.351
Sarcasm detection without slang word fixing	0.718	0.492	0.354
Sentiment analysis with English translation	0.750	0.750	0.750
Sentiment analysis without English translation	0.742	0.744	0.985
Sentiment analysis with slang word fixing	0.750	0.750	0.750
Sentiment analysis without slang word fixing	0.753	0.755	0.982

4. CONCLUSIONS

An improved model of sentiment analysis on Indonesian tweets is proposed in this paper, by integrating the model of sarcasm detection. The evaluation shows that the sarcasm detection model improves the performance of sentiment analysis model about 5.49% in average, from . The highest accuracy of the sarcasm detection model is 80.4%, with 83.2% of precision, and 91.3% of recall. The model can be considered as stable because it reaches the accuracy above 70% accuracy in all cases of test using k-fold cross validation, with the maximum accuracy of 72%. In terms of features, sentiment related features gives the highest impact in sarcasm detection model and reaches 72% of accuracy.

This study in sarcasm detection has many rooms for improvements in the future, one of them is the treatment of slang words. Furthermore, the consideration of context in sarcasm detection is also an interesting area for future work.

REFERENCES

[1] A. C. Pandey, D. S., Rajpoot, M. Saraswat, “Twitter sentiment analysis using hybrid cuckoo search method”, Information Processing and Management, Vol. 53 (4), pp. 764 – 779, July 2017.

[2] M. Bouazizi and T. Ohtsuki, “Sarcasm detection in twitter: : "All Your Products Are Incredibly Amazing!!!" - Are They Really?”, 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6-10 December 2015.

[3] Zhang and Gao., “Performance Analysis and Improvement of Naïve Bayes in Text Classification Application”, IEEE Conference Anthology, China, 1-8 January 2013.

[4] Maynard and Greenwood, “Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis,” Proceedings of LREC, pp. 4238-4243.
Available: https://gate.ac.uk/sale/lrec2014/arcomem/sarcasm.pdf [Accessed: 1-Jan-2018]

[5] E. Lunando and A. Purwarianti , “Indonesian social media sentiment analysis with sarcasm detection,” *2013 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2013*, pp. 195–198, 2013.
Available: https://www.semanticscholar.org/paper/Indonesian-social-media-sentiment-analysis-with-LunandoPurwarianti/c82b544f6f43c69bd0b0bf58f4b963f6c4f31377
[Accessed: 23-Jan-2018]

[6] M. V. Datla, “Bench marking of classification algorithms: Decision Trees and Random Forests-a case study using R,” *Int. Conf. Trends Autom. Commun. Comput. Technol. I-TACT 2015*. Available:https://www.scopus.com/record/display.uri?eid=2-s2.0-84979282682&origin=inward&txGid=45c76f2dcf97f506cbcafd87959e7f31_[Accessed: 18-Nov-2018]

[7] D. Suswanto, “Analisis Perbandingan Metode Machine Learning untuk Prediksi Khasiat Jamu” , Thesis, Institut Pertanian Bandung, Bandung, 2016.

[8] F. Prawira, “Pengaruh Pendeteksian Sarkasme Terhadap Ukuran Kualitas Analisis Sentimen Pada Twitter”, Thesis, Universitas Gadjah Mada, Yogyakarta, 2017.

[9] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*, Cambridge University Press, New York, USA, 2008.

[10] M. Habibi, “Analisis Sentimen Dan Klasifikasi Komentar Mahasiswa Pada Sistem Evaluasi Pembelajaran Menggunakan Kombinasi Knn Berbasis Cosine Similarity Dan Supervised Model”, Thesis, Universitas Gadjah Mada, Yogyakarta, 2017.