A comparative study of sentiment analysis using SVM and SentiWordNet

Mohammad Fikri, Riyanarto Sarno
Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia

Article Info

Article history:

Received May 21, 2018
Revised Nov 13, 2018
Accepted Dec 5, 2018

Keywords:

Rule-based
Sentiment analysis
Sentiwordnet
Support vector machine
Wordnet

ABSTRACT

Sentiment analysis has grown rapidly and impacts on the number of services using the internet popping up in Indonesia. In this research, the sentiment analysis uses the rule-based method with the help of SentiWordNet and Support Vector Machine (SVM) algorithm with Term Frequency-Inverse Document Frequency (TF-IDF) as a feature extraction method. The data as the case study for the sentiment analysis is written in Indonesian language. Since the number of sentences in positive, negative and neutral classes is imbalanced, the oversampling method is implemented. For imbalanced dataset, the rule-based SentiWordNet and SVM algorithm achieve accuracies of 56% and 76%, respectively. However, for the balanced dataset, the rule-based SentiWordNet and SVM algorithm achieve accuracies of 52% and 89%, respectively.

Corresponding Author:

Riyanarto Sarno,
Department of Informatics,
Institut Teknologi Sepuluh Nopember,
Jalan Raya ITS, Keputih, Sukolilo, Kota Surabaya, Jawa Timur 60111, Indonesia.
Email: riyanarto@if.its.ac.id

1. INTRODUCTION

With the increase in the number of internets, users can open an opportunity to give good impact to an organization because of the data generated through internet user activity. These data can be opinions or facts about something. This research focuses on public opinions about products in the form of applications on smartphones. These opinions can be further analyzed for obtaining consideration of the decision-making in a company that creates the application. Among the various technical analyzes, the technique is called sentiment analysis [1]. This technique processes text documents in the form of opinions to generate a piece of information so that information can be used to divide opinions into positive, negative, or neutral opinions. In the development of information technology, opinion mining is one of the favorite research topics in the field of Natural Language Processing (NLP).

In this research, we compare the implementation of supervised machine learning and rule-based for sentiment analysis using data from Google Playstore and Apple Appstore written in Indonesian language. The method to get the data is the same methods as the method in these several papers [2-3]. Each product review a case folding process, normalization of punctuation, normalization of the slang word, stopword removal, transformation into single line, and tokenization will be carried out as stated on [4-5]. For implementations using supervised machine learning, we use Term Frequency-Inverse Document Frequency (TF-IDF) to convert text into classifiable features and Support Vector Machines to classify the processes. SentiWordNet does not support languanges other than English; whereas the language of the data is Indonesian. Therefore, translating the opinions into English is needed, so that result of the translation can be done by the analysis of the opinions [6-7]. This research consists of section 2 that explains the used method, section 3 that explains the results of the analysis, and section 4 that contains the conclusions.

2. RESEARCH METHOD

The main objective of this study is to compare text classification algorithms between using a rule-based algorithm with the help of SentiWordNet and using the combination of TF-IDF algorithm and Support Vector Machine (SVM) algorithm. TF-IDF extracts features from text to a vector and SVM classifies an imbalanced dataset into the number of positive, negative, and neutral classes. The results of each algorithm used will be sorted and compared based on the scores of the F-Score and Accuracy.

2.1. Data Construction

The dataset contains public opinions about some apps. Those opinions are written in Indonesian language and are taken from Google PlayStore and Apple AppStore. The data consists of "id_komen" as the identification code of comments, "title_komen" as the title of comments, and "Komen" as the detailed comments. The sentiments for each sentence are determined by humans into three classes, i.e. positive class, neutral class, and negative class. The dataset contains 553 sentences which are 259 positive class, 241 negative class, and 53 neutral class. The positive class and negative class are balanced; however, the neutral class is imbalanced because the neutral class has fewer sentences than the others. The data is stored in a data frame that has "COMMENT" column for comment, "SENTIMENT" column for the correct sentiment, and "SENTIMENT_ID" column for sentiment id, i.e. "0" as negative, "1" as positive, and "2" as neutral.

2.2. Balancing Data

Balancing an unbalanced dataset is a critical process in machine learning. The method used this time by oversampling the minority class [8] is shown in Figure 1. The explanation of Figure 1 is as follows:

1. Marking the Minority Class and Majority Class.
   First, creating one column named flag_balance. Then, marking the minority class (neutral) by filling in the flag_balance field with 1 and the majority class (positive and negative) with 0.
2. Splits into 2 Data frames.
   The data which has 1 in the flag_balance column become the minority data frame and the data which has 0 in the flag_balance column 0 become the majority data frame [9].
3. Resample The Minority Class Dataframe.
   The first task is oversampling the minority resampled class by using the existing algorithm in the scikit-learn [9]. After that, resampling randomly until the number of minority classes equals the average number of majority classes. In this research, neutral is the minority class, positive and negative is the majority class.
4. Combine The Majority Class Dataframe and The Upsampled Minority.
   First, merging both data frame (majority and minority). Then, randomizing the sequence on the data frame so that data are merged into random.

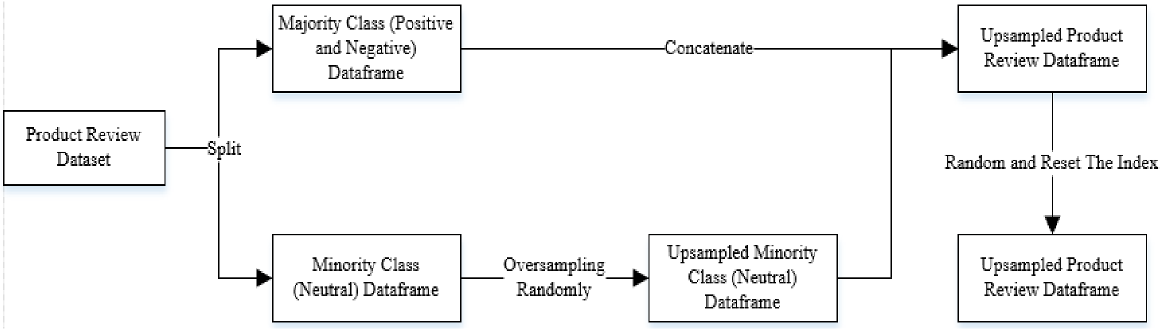

Product Review Dataset
Split
Majority Class (Positive and Negative) Dataframe
Minority Class (Neutral) Dataframe
Oversampling Randomly
Upsampled Minority Class (Neutral) Dataframe
Concatenate
Upsampled Product Review Dataframe
Random and Reset The Index
Upsampled Product Review Dataframe

Figure 1. Balancing Imbalanced Dataset Process

2.3. Preprocessing Data

Because Indonesian language using by the data is informal, the preprocessing is done to change the text into Indonesian language in the formal form. The following preprocessing steps are described as follows:

1. Enter" Character Normalization.
   Remove "\ n" or enter on the sentence to be a single line only.
2. Lowercase Normalization.
   Turn the sentence into all lowercase.
3. Unnecessary Character Normalization.

Delete a recurring character whether it is an alphabet or a punctuation. For example "Setiaaaa" to "Setia" and “Yah....." to "Yah".

4. Punctuation Normalization.
   Delete punctuation on the sentence.
5. Slang Word Normalization.
   Fixed informal words and abbreviations. The fix uses a manual way, not spellchecking. The manual way means matching the word with a hash map containing slang word if the word matches the key of the slang word hashmap then the word is changed into the value of the key of the hash map. For example, abbreviations such as "spt" to be "seperti" and informal words like "pake" to "pakai".
6. Stopword Removal
   Delete the words that often appear in each sentence. The type of the deleted word is a conjunction word, such as "dan", "serta", "serta", and others. Table 1 is an example of the preprocessing results. The original text uses Indonesian language in the informal form, and the preprocessing results changes the language of original text from informal form to formal form.

Table 1. Preprocessing Results in Indonesian Language

No	Original Text	After Preprocessing Text
1	BETULIN DONG APLIKASINYA , RUSAK MULU NIH!!!!	betulin dong aplikasinya rusak melulu nih
2	App nya crash terus!! Tolong diperbaiki agar service nya semakin baik	app nya crash terus tolong diperbaiki service nya semakin baik

2.4. Rule-Based Using Sentiwordnet

The purpose of this research is to compare two different methods and one of the methods is SentiWordNet. Meanwhile, the process of classification is different because the SentiWordNet is currently very limited and not yet available in Indonesian language.

1. Translate Data

Because Sentiwordnet is currently still limited to the Indonesian language, therefore the data is translated into English language. Google Translate is used as the language translator tool. The results of this translator tool can be assumed quite well although there are still some sentences that do not have the correct sentence structure.

2. Tokenization and POS Tagging

Tokenization is a process to split one sentence into a piece of the word. At this process, the sentence is split into unigram which means several parts consisting of 1 piece of a word.

After the tokenization process, each unigram is determined the part of speech [11]. There are 8 parts of speech which are nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions, and interjections. However, the part of speech tag is a Penn Treebank POS tag and SentiWordNet only has four general POS tags of noun (N), verb (V), adjective (A), and adverb (R). Finally, converting the POS tag to SentiWordNet POS tags is necessary [12] with the following rules :

a) Noun (N)
If POS tags are 'NN', 'NNS', 'NNP', 'NNPS', then the POS tags are changed into 'N'.

b) *Verb* (V)
If POS tags are 'VB', 'VBD', 'VBG', 'VBN', ‘VBP' or ‘VBZ', then the POS tags are changed into 'V'.

c) *Adjective* (A)
If POS tags are 'JJ', 'JJR', or 'JJS', then the POS tags are changed into 'A'.

d) *Adverb* (R)
If POS tags are 'RB', 'RBR', or 'RBS', then the POS tags are changed into 'R'.

The latter on this process is done lemmatization. Lemmatization is a process where a word is returned to its basic form back in accordance with the POS tag.

3. Sentiment Classification

The sentiment classification in this study uses SentiWordnet and Wordnet tools. SentiWordnet is used to find the score of each synset and Wordnet is used to search for synonyms of each word being analyzed.

Scores for each word are searched using SentiWordnet according to POS tags if the scores are more than 0 then it is taken, otherwise it is bypassed.

After sentiment scores per word are obtained, we have to do a total calculation to get sentiment score for one sentence. The semantic orientation calculation uses the method according to Equation (1) and (2).

$$Score_{positive} = \sum_{i \in t}^{n} \frac{positive\ score_i}{n} \quad (1)$$

$$Score_{negative} = \sum_{i \in t}^{n} \frac{negative\ score_i}{n} \quad (2)$$

Based on Equation (1), (2), Score$_{positive}$ is the final number of positive scores while the Score$_{negative}$ is the final number of negative scores. And n is the number of words whose first sentence value is above 0. Then, to get sentiment value, Equation (3) is applied.

$$Sentence_{sentiment} \begin{cases} positive\ if\ score_{positive} - score_{negative} \geq 0.05 \\ negative\ if\ score_{positive} - score_{negative} \leq -0.05 \\ neutral\ if\ -0.05 < score_{positive} - score_{negative} < 0.05 \end{cases} \quad (3)$$

Sentiment obtained value uses positive score difference and negative score. If the score difference is greater than 0.05 then the sentiment value is positive. If the score difference is smaller equal to-0.05 then the sentiment value is negative. And if the score difference is smaller than 0.05 and greater than-0.05 then the sentiment value is neutral.

2.5. Supervised Machine Learning using SVM

In this section, the TF-IDF method is used as the feature extraction process from text to vector and SVM is used as an algorithm for text classification.

2.5.1 Feature Extraction using TF-IDF

Term Frequency-Inverse Document Frequency is a method for converting a document (sentence) in a corpus into a statistically measurable weight in which this weight represents how important the word is in the document or phrase [13]. There are several tasks to transform a corpus into a weight using the TF-IDF method.

a) Tokenization

Documents that exist in a corpus are tokenized into unigram and bigram. Unigram consists of one word and bigram consists of 2 words. The tokenization process can be seen in Figure 2. Based on Figure 2, all of unigrams and bigrams are still in Indonesian language because the documents are written in Indonesian language.

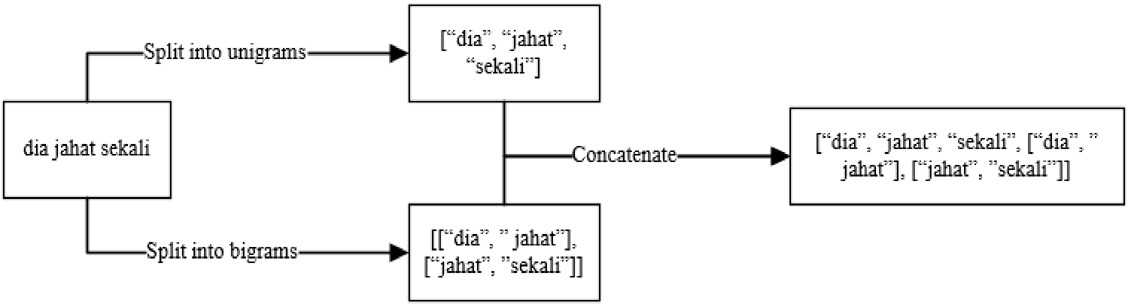

dia jahat sekali
Split into unigrams
Split into bigrams
[“dia”, “jahat”, “sekali”]
[[“dia”, ” jahat”], [“jahat”, ”sekali”]]
Concatenate
[“dia”, “jahat”, “sekali”, [“dia”, ” jahat”], [“jahat”, ”sekali”]]

Figure 2. Tokenization Process of Documents in Indonesian Language

b) Term Frequencies

Term Frequencies (TF) measures how often a word appears in a document. It is possible that a term would appear much more times in long documents than shorter ones. Term Frequencies is the total count of a term in a document.

c) Inverse Document Frequency

Inverse Document Frequency (IDF) measures how important a term to a document. When calculating term frequencies, assuming that all terms have the same importance in a document; whereas conjunctional words in Indonesian language such as "dan", "adalah", and "serta", are very often appear in several documents (sentences) thereby reducing how important the word is in a sentence.

$$IDF(t,d) = \ln\left[\frac{(1+n)}{(1+df(t,d))}\right] + 1 \tag{4}$$

In Equation (4), IDF(t, d) is the Inverse Document Frequency of a term in a document, n is the total number of the documents, DF(t, d) is the number of documents with a term (t) in it. The effect of adding “1” to the IDF in the equation above is that terms with zero IDF; i.e., terms that occur in all documents in a training set, will not be entirely ignored. The constant “1” is added to the numerator and denominator of the IDF as if an extra document was seen containing every term in the collection exactly once, which prevents zero divisions.

d) Calculate TF-IDF Weight

In the process of calculating weights using the TF-IDF method where all the Equations used are in accordance with Equation (4), (5). This section will be exemplified how the calculation of weights using the TF-IDF method.

$$\mathrm{TF-IDF(t,d)} = \mathrm{TF(t,d)\ x\ IDF(t,d)} \tag{5}$$

e) Normalize TF-IDF Weight

Normalization is done so that the TF-IDF value has a well-balanced weight. Normalization is done using L2 norm so that the weight of tf-idf for each term has a weight of 0-1 scale, see Equation (6).

$$v_{norm} = \frac{v}{\|v\|^2} = \frac{v}{\sqrt{{v_1}^2 + {v_2}^2 + \cdots + {v_n}^2}} \tag{6}$$

As an example, two documents (D1 and D2) are computed the TF-IDF values. Terms are obtained using the tokenization method as shown on Figure 2. DF is the document frequency for each Term in a document (Dn), IDF is the inverse document frequency for each Term calculated using Equation (4). Term Frequency–Inverse Document Frequency (TF-IDF) for each Term in a document (Dn) is calculated using Equation (5) and is normalized using L2 Norm as shown by Equation (6). The results are explained in Table 2. The terms are written in Indonesian language. D_1 is “dia baik sekali” and D_2 is “dia jahat sekali.”

Table 2. TF-IDF Weighting with Terms written in Indonesian language

Term	TF		DF	IDF	TF-IDF		TF-IDF (L2 Norm)	
	D_1	D_2			D_1	D_2	D_1	D_2
dia	1	1	2	1	1	1	0.355	0.355
baik	1	0	1	1.405	1.405	0	0.499	0
sekali	1	1	2	1	1	1	0.355	0.355
jahat	0	1	1	1.405	0	1.405	0	0.499
dia baik	1	0	1	1.405	1.405	0	0.499	0
baik sekali	1	0	1	1.405	1.405	0	0.499	0
dia jahat	0	1	1	1.405	0	1.405	0	0.499
jahat sekali	0	1	1	1.405	0	1.405	0	0.499

2.5.2 Support Vector Machine

Support Vector Machine (SVM) is a classification method for linear or nonlinear data by using nonlinear data mapped to convert training data to a higher dimension. This method find hyperplane by maximizing margin or distance between classes [14], [15].

Considering the class in classification, the one vs rest strategy is implemented, this strategy consists in fitting one classifier per class.

2.6. Comparing Results

Results from the classification of rule-based using SentiWordNet and supervised machine learning and using SVM algorithm with TF-IDF as feature extraction are compared by using Recall, Precision, F-Score parameters.

Precision is the ability of a classification model to identify only the relevant data points, see Equation (7). Recall is the ability of a model to find all the relevant cases within a dataset, see Equation (8). F-Score is the harmonic mean of precision and recall taking both metrics into account in the Equation (9). Accuracy is the quality or state of being correct or precise, see Equation (10).

$$Precision = \frac{TP}{(TP+FP)} \tag{7}$$

$$Recall = \frac{TP}{(TP+FN)} \tag{8}$$

$$F - Score = 2\, x \frac{(Precision\ x\ Recall)}{(Precision+Recall)} \tag{9}$$

$$Accuracy = \frac{T_{positive} + T_{neutral} + T_{negative}}{T_{positive} + T_{neutral} + T_{negative} + F_{positive} + F_{neutral} + F_{negative}} \tag{10}$$

Then for the split between training data and testing data using K-Fold Cross Validation method. So the data are divided into k fold and then will be executed classification process as much as the k and for the testing data is selected from one of k fold and training data is fold which are not used as the data testing [16]. The selection of data testing per round is selected in sequence starting from the folds 1, see Figure 3 for the illustration. For example, round 1 is used as data testing fold 1 and so on.

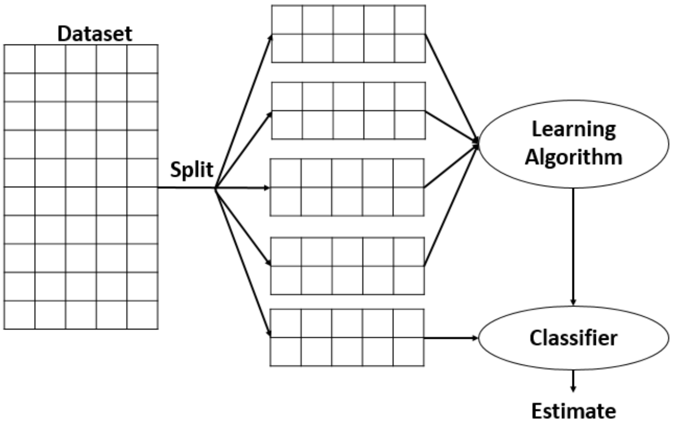


Figure 3. K-Fold Cross-Validation

3. RESULTS AND ANALYSIS

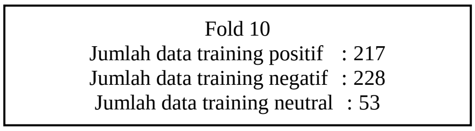
Implementation of this research is created by using Python Programming Language. The total number of the dataset is 553 data with detail for positive class 259 data, negative class 241 data, and neutral class 53 data. Then splitting between data training and data testing, we set k = 10 for the K-Fold Cross-Validation splitting method.

In Table 3, the results between F-Score and Accuracy obtained using SVM algorithm compared to using rule-based SentiWordNet is quite close. SVM algorithm is slightly better with an accuracy of 76% and f-score 51%. Rule-based SentiWordnet gets accuracy 56% and f-score 48%.

Table 3. Comparison of Results using 10-Fold Cross Validation Before Balancing Dataset

Method	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
SVM	48.74	53.23	50.89	75.75
Rule-based SentiWordNet	49.5	46.42	47.76	55.81

But that can be seen, there is a considerable difference between Accuracy and F-Score when using SVM algorithm, with a difference of 20% can be said there is an imbalance between the classes present in the dataset. For the 10th round K-Fold Cross-Validation, the data testing for the neutral class does not exist at all because all of the 53 neutral class data has become training data, see Figure 4.



```
Fold 10
Jumlah data training positif   : 217
Jumlah data training negatif  : 228
Jumlah data training neutral  : 53
```

Figure 4. Underfitting on Neutral Class

Due to the underfitting dataset, the dataset is balanced using balancing method as shown by Figure 1. During the process of balancing the dataset, the amount of neutral class data increases to 250 data due to the average of positive and negative data.

Table 4 shows that the results of SVM algorithm with the TF-IDF method as feature extraction (F-score 83% and Accuracy 89%) are better than the results of Rule-based SentiWordNet (F-Score 50% and Accuracy 51%). The results from Table 3 and Table 4 can be compared, the balanced datasets get better results when using SVM algorithm with TF-IDF as feature extractor, since it increases the Accuracy and F-Score because the neutral class has been balanced; however, the SentiWordNet rule-based algorithm has decreased both in Accuracy and F-Score. The experiment found the average number of words which were not in the synsets was 573 words. Therefore, the rule-based SentiWordNet considering those missing 573 synsets can increase the accuracy to about 20%.

Table 4. Comparison of Results using 10-Fold Cross Validation after Balancing Dataset

Method	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
SVM	82.02	85.45	83.69	89.06
Rule-based SentiWordNet	51.34	49.65	50.45	51.59

4. CONCLUSION

Based on the results of the classification using SVM and rule-based, we can conclude that:

1. Balancing datasets can improve both Accuracy and F-Score achieved by SVM algorithm with TF-IDF as feature extraction method; however balancing datasets can decrease both Accuracy and F-Score resulted by the ruled-based SentiWordNet.
2. SVM algorithm with TF-IDF as feature extraction method achieves better results than those resulted by the rule-based SentiWordNet.
3. There are still many words that do not have synset because Indonesian vocabulary is still incomplete. Using SentiWordNet and translator tools are still not good enough for translating Indonesian into English.

ACKNOWLEDGEMENTS

The authors would like to thank to Institut Teknologi Sepuluh Nopember, *Direktorat Riset dan Pengabdian Masyarakat*, *Direktorat Jenderal Penguatan Riset dan Pengembangan*, the Ministry of Research, Technology, and Higher Education of Indonesia for financing the research.

REFERENCES

[1] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Found. Trends® InformatioPang, B., Lee, L. (2006). Opin. Min. Sentim. Anal. Found. Trends® Inf. Retrieval, 1(2), 91–231. doi10.1561/1500000001n Retr.*, vol. 1, no. 2, pp. 91–231, 2006.

[2] M. R. Islam. Numeric rating of Apps on Google Play Store by sentiment analysis on user reviews. *1st Int. Conf. Electr. Eng. Inf. Commun. Technol. ICEEICT 2014*, pp. 1–4, 2014.

[3] E. Guzman and W. Maalej. How do users like this feature? A fine grained sentiment analysis of App reviews. *2014 IEEE 22nd Int. Requir. Eng. Conf. RE 2014-Proc.*, pp. 153–162, 2014.

[4] A. R. Naradhipa and A. Purwarianti. Sentiment Classification for Indonesian Messages in Social Media. *Int. Conf. Electr. Eng. Informatics*, no. July, pp. 2–5, 2011.

[5] D. Ayu and K. Khotimah. Sentiment Detection of Comment Titles in Booking . com Using Probabilistic Latent Semantic Analysis. *2018 6th Int. Conf. Inf. Commun. Technol.*, vol. 0, no. c, pp. 514–519, 2018.

[6] N. Farra, E. Challita, R. A. Assi, and H. Hajj. Sentence-level and document-level sentiment mining for arabic texts. *Proc.-IEEE Int. Conf. Data Mining, ICDM*, pp. 1114–1119, 2010.

[7] K. Denecke. Using SentiWordNet for multilingual sentiment analysis. *Proc.-Int. Conf. Data Eng.*, pp. 507–512, 2008.

[8] A. Sun, E. P. Lim, and Y. Liu. On strategies for imbalanced text classification using SVM: A comparative study. *Decis. Support Syst.*, vol. 48, no. 1, pp. 191–201, 2009.

[9] W. McKinney. Data Structures for Statistical Computing in Python," *Proc. 9th Python Sci. Conf.*, vol. 1697900, no. Scipy, pp. 51–56, 2010.

[10] F. Pedregosa *et al.* Scikit-learn: Machine Learning in Python Gaël Varoquaux. *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[11] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. 2009.

[12] B. S. Rintyarna and R. Sarno. Adapted weighted graph for Word Sense Disambiguation. *2016 4th Int. Conf. Inf. Commun. Technol. ICoICT 2016*, vol. 4, no. c, 2016.

[13] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, vol. 26, no. 3, pp. 1–37, 2008.

[14] B. Y. Pratama and R. Sarno. Personality classification based on Twitter text using Naive Bayes, KNN and SVM. *Proc. 2015 Int. Conf. Data Softw. Eng. ICODSE 2015*, pp. 170–174, 2016.

[15] F. H. Rachman, R. Sarno, and C. Fatichah. Music emotion classification based on lyrics-audio using corpus based emotion. *Int. J. Electr. Comput. Eng.*, vol. 8, no. 3, pp. 1720–1730, 2018.

[16] M. Jupri and R. Sarno. Taxpayer compliance classification using C4.5, SVM, KNN, Naive Bayes and MLP. in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018, pp. 297–303.

BIOGRAPHIES OF AUTHORS

	Mohammad Fikri is now fourth year student of Informatics Department at Institut Teknologi Sepuluh Nopember. His current interests are in Text Retrieval and Image Retrieval. E-mail: fikri.mohammad15@mhs.if.its.ac.id
	Riyanarto Sarno received M.Sc and Ph.D in Computer Science from the University of Brunswick Canada in 1988 and 1992. In 2003 he was promoted to a Full Professor. His teaching and research interests includes Internet of Things, Process Aware Information Systems, Intelligent Systems and Business Process Management. E-mail: riyanarto@if.its.ac.id