The Accuracy Improvement of Text Mining Classification on Hospital Review through The Alteration in The Preprocessing Stage

Triyas Hevianto Saputro
Information Technology Study Program Magister Program
of Technnology University of Yogyakarta
Yogyakarta, Indonesia
Email: triyasheviantosaputro [AT] student.uty.ac.id

Arief Hermawan
Information Technology Study Program
Magister Program of Technology University of Yogyakarta
Yogyakarta, Indonesia
Email: ariefdb [AT] uty.ac.id

Abstract—**Sentiment analysis is a part of text mining used to dig up information from a sentence or document. This study focuses on text classification for the purpose of a sentiment analysis on hospital review by customers through criticism and suggestion on Google Maps Review. The data of texts collected still contain a lot of nonstandard words. These nonstandard words cause problem in the preprocessing stage. Thus, the selection and combination of techniques in the preprocessing stage emerge as something crucial for the accuracy improvement in the computation of machine learning. However, not all of the techniques in the preprocessing stage can contribute to improve the accuracy on classification machine. The objective of this study is to improve the accuracy of classification model on hospital review by customers for a sentiment analysis modeling. Through the implementation of the preprocessing technique combination, it can produce a highly accurate classification model. This study experimented with several preprocessing techniques: (1) tokenization, (2) case folding, (3) stop words removal, (4) stemming, and (5) removing punctuation and number. The experiment was done by adding the preprocessing methods: (1) spelling correction and (2) Slang. The result shows that spelling correction and Slang method can assist for improving the accuracy value. Furthermore, the selection of suitable preprocessing technique combination can fasten the training process to produce the more ideal text classification model.**

Keywords-accuracy improvement; preprocessing; text classification

I. Introduction

Hospital is an instance which serves publics in health. Almost every day the instance is visited by patients. Thus, it is common when the hospital management tries to give satisfaction toward its customers. To gather the customers' review, it serves such a customer service for delivering the criticism and suggestion. Thus, the management will understand the customers' review toward the hospital's service. Customers, either offline or online, can deliver their criticism and suggestion through a provided suggestion box or via Short Message Service (SMS), Whatsapp, Google Map Review, or other social media. Along with the increasing of smartphone users and the change of habit during pandemic, the delivery of criticism and suggestion is mostly done by online. And to ease in digging up the customers' review, text mining approach is needed.

Nowadays, the rapid development of digitalization makes text mining a popular topic research. By the increasing the number of data on texts, sentiment analysis has become an important part to dig up the publics' review toward instances. It can be conducted by lexical and machine learning approach. In lexical approach, dictionary and corpus are the reference [1] while the machine learning approach itself is divided into four categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [1].

Sentiment analysis is a part of text mining used to dig up information from a sentence or document. This study focuses on text classification for the purpose of a sentiment analysis on hospital review by customers through the criticism and suggestion on Google Map Review. The data of texts collected still contain a lot of nonstandard words. These nonstandard words cause problem in the preprocessing stage. Mistyping, the use of nonstandard acronym, or the use of words which are not in accordance with the rules of Bahasa, will result a less ideal model. These problems may cause different meanings in each word or sentence from the view point of computation or classification machine. The acronym "yg", for instance. It has different vector value with "yang" in the computation machine. Thus, the selection and combination of techniques in the preprocessing stage emerge as something crucial for the accuracy improvement in the computation of machine learning. However, not all of the techniques in the preprocessing stage can contribute to improve the accuracy on classification machine. Several studies related to text classification have been observed, such as sentiment analysis on Twitter [2], algorithm of text classification [3], stop words [4], etc. The previous study suggests a compilation of preprocessing techniques for classifying texts in other languages (not Bahasa). This study will experiment with the dataset of Bahasa collected from

Google Map Review in which each language has its own uniqueness. Thus, it needs stages of preprocessing which are suitable with the objective of the accuracy improvement on the model of classification machine. Algorithm of text classification will work more efficient if the selection of feature extraction method and the way to evaluate are compatible [3]. Besides, cleaning texts and documents can also help the accuracy of algorithm [3].

The experiment in this study will apply several techniques in the preprocessing stage such as tokenization, case folding, stop words removal, stemming, and removing punctuation and number. The experiment then will be followed by adding two other techniques: spelling correction and Slang. This study will use each technique and combine them. The approach applied is a classical machine learning model: Naïve Bayes [3][5][6]. Studies related to this research such as The Effect of Stemmer of Bahasa [1], The Comparison of Stemmer of Bahasa [7], Text Classification on Sentiment Analysis [7][8][9], and etc. These studies were done to optimize the accuracy value. However, they only focused on the preprocessing techniques suggested so it needs to be tested in other domains. Thus, this study will combine those preprocessing techniques but not all of those techniques will be applied. The benchmark of the preprocessing technique is then used as the standard of comparison [1][8][10]. This study proposes the hypothesis of combination of the preprocessing techniques of dataset in Bahasa on hospital review by customers in which the corpus is labelled by positive and negative. The objective of this study is to improve the accuracy of text classification modeling on hospital review by customers for the purpose of sentiment analysis model. A high accuracy value of classification meets value of 100%. The accuracy value of previous studies on text classification was above 60% in average [8][12]. Through applying the combination of preprocessing techniques, it is expected that it can result a classification model with a higher accuracy.

II. RELATED WORKS

This study applied the model of text preprocessing toward comments on YouTube in Indonesian in sentiment analysis [11]. The main challenge in this study was to process noise with stop words and Slang technique. The preprocessing technique in this study used standard of text preprocessing in Bahasa, stop words removal, words referring subject or object, and changing Slang. The feature extraction used were count-vectorizer and Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. This study used 4 preprocessing scenarios. The combination of technique in the preprocessing stage consisted of standard preprocessing, stop words removal, and conversion of Slang vocabulary into standard Bahasa. In this study, the highest accuracy value was 88,8%. There was 3-3,5% of accuracy increasing.

This study used the combination of spelling correction and n-gram method and met the accuracy of 78,07% to produce one standard suggested word from one nonstandard word in a sentence [12]. The main obstacle in this study was mistyping in document caused by human error. This study used Peter Norvig spelling correction method. It experimented on 9 scenarios of mistyping case in 160 sentences. It used dictionary of document -Advanced Research Project Agency (ARPA). ARPA is the result of process of building language model using Sri Language Modeling (SRILM) toolkit.

The error on word (mistyping) is crucial case during document writing. This research identified word error in document in Bahasa [13]. It used n-gram and Levenshtein method. It focused on thesis document. The counting of TF-IDF was used to gain the cosine similarity value. In the testing step, it resulted the highest precision value (0.97) on the error of insertion while the precision value of 1 was on the error of substitution.

This research implemented sentiment analysis on comments on YouTube [8]. It used Naïve Bayes Classifier (NBC) method to analyze the sentiment on government instances. It resulted the accuracy value of 69,23% and 64,10% in two different domains.

III. RESEARCH METHODS

The steps of text classification process in this study can be seen in the Figure.1. The dataset was collected from Google Map Review and then it was labelled by positive and negative. Before the preprocessing stage was done, the dataset was divided. After that, the text classification process was done. It started from the preprocessing stage until model evaluation.

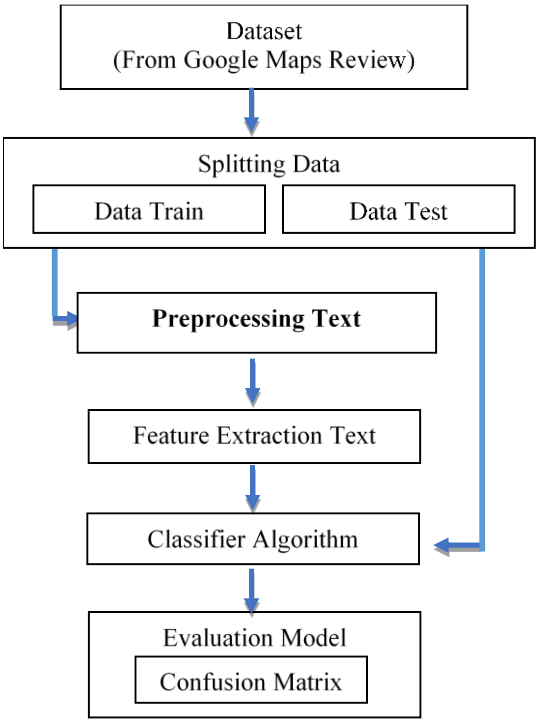

Figure 1. Schema of Text Classification

The preprocessing methods in this study were removing punctuation and number (rpn), low case (lc), stop words removal (rs), stemming (st), spelling correction (sc), and Slang (sl). After that, it was given acronym on each of the preprocessing technique used. The result alteration of each preprocessing technique used in this experiment can be seen in the TABLE.I.

A. Dataset

The dataset was collected from Google Maps Review of several hospitals in Yogyakarta. The data were taken from columns of commentary using crawling tools automatically in July 19th 2020. This dataset had positive and negative label with total number of 300 rows of review. Positive label is a sentence which has positive meaning, such as fondness, contentment, happiness, compliment, satisfaction, etc. Otherwise, negative label is a sentence which has negative meaning, such as anger, disappointment, sadness, etc.

B. Preprocessing Methods

This study experimented with this following combination of several preprocessing methods:

1) Case Folding, this technique is used to standardize the use of letter in the dataset [3]. The standardization changes all letters into lowercase size with the purpose of equaling the vector.

2) Remove Punctuation and Number, this technique is used to remove the punctuation or special characters and meaningless numbers [14].

3) Stop Words Removal, this technique is used to reduce the number of words in a document which has a great number of appearance [14]. Connector "yang", "dan", "akan", for instance.

TABLE I. *COMBINATION OF PREPROCESSING TECHNIQUE*

Preprocessing	**rpn**	**lc**	**sc**	**sl**	**rs**	**st**
1	yes	yes	**no**	**no**	**yes**	yes
2	no	yes	no	no	yes	no
3	yes	yes	yes	no	yes	yes
4	yes	yes	yes	no	yes	no
5	yes	yes	yes	no	no	yes
6	yes	yes	yes	no	no	**no**
7	yes	yes	no	yes	yes	yes
8	yes	yes	no	yes	yes	no
9	yes	yes	no	yes	no	yes
10	yes	yes	no	yes	no	no
11	no	no	yes	no	no	**no**
12	no	no	no	yes	no	**no**
13	no	no	yes	yes	no	no

4) Stemming, it is used to reduce the form of inflection and word form which is derivatively related and becomes the same basic form [3]. Stemming technique used in this study was literary stemming of Bahasa. This technique returns the inflectional word form into the basic form.

5) Slang, this technique is used to convert non-standard language. The Slang dictionary used was Colloquial Indonesian Lexicon [15].

6) Spelling Correction, this technique uses algorithm of Norvig spelling correction. The algorithm is able to give one suggested word and directly repairs it [12]. The algorithm of Peter Norvig combines the process of removing, adding, separating, replacing, and moving letter of nonstandard word. The process of searching the nonstandard words is based on the corpus. The suggested words are given based on the algorithm.

C. Word Feature Extraction

The word feature extraction in this study used TF-IDF [16]. Commonly, feature extraction converts a compilation of unstructured texts and documents into structured dimension of features [3]. This method uses library provided in the Scikit-learn. One of text feature extraction techniques is TF-IDF [17]. The weighting of TF-IDF can be counted by this following equation:

$$TF - IDF(d,t) = TF(d,t) \cdot \log\frac{N}{df(t)} \qquad (1)$$

TF is term of frequency, IDF is inverse document frequency, D is document, T is term, N is total number of document in corpus.

D. Algorithm of Text Classification

The algorithm of text classification in this study used multinomial (NBC) algorithm. NBC has been widely used to categorize documents since 1950s [3]. The method of NBC [5] [6] is theoretically based on Bayes theorem which is formulated by Thomas Bayes during 1702-1761 [17]. This technique is a generative model which is the most traditional-text categorization method. NBC implements statistic approach in classifying the data by counting data probability to be classified into certain classes of the training data. The probability of counting using Bayes theorem is as the following equation [2].

$$P(H|X) = \frac{P(X|H)\,P(H)}{P(X)} \qquad (2)$$

X is the proof, H is hypothesis. P(H|X) is probability that H is true if it is given the proof of X. P(H|X) is probability that proof of X is true if it is given hypothesis of H. Otherwise, P (X) is probability proof of X and P (H) is probability hypothesis of H [18].

NBC for multinomial model is suitable to discrete feature (total number of words for text classification). The distribution of multinomial usually needs the number of integer features. However, practically, the counting of fraction, such as TF-IDF, can also be functioned [19].

Multinomial NBC can overcome the problem of the losing value through smoothing. This process adds small value (alpha) in the counting of each feature during the probability counting process so as to the combination of feature and class have no null probability. This process is called Laplace smoothing with the alpha value = 1 and Lidstone smoothing with alpha value > 1. The value of smoothing feature is formulated as the following equation (3) [20].

$$\widehat{E_{cj}} \frac{E_{cj}+\propto}{E_{cj}+\propto n} \tag{3}$$

Ecj is the total number of feature value j that is given by class C seen in the training, Ec is the total number of Ecj for all classes, and n is the number of feature. The classification of Multinomial Naïve Bayes is mainly used in text mining in which its feature is rare number of words.

E. Cross Validation

Cross validation is a technique to evaluate the work of machine learning model [19]. In [19], learning the parameter of prediction function and testing them on the same data are a methodological failure. The model will only repeat the label of new-identified sample and will meet the highest value, yet it fails to predict anything useful of the unidentified data. Such this condition is called overfitting. To avoid this case, the machine learning experiment is observed by saving most of the provided data as the testing dataset. Cross validation procedure is done by dividing the dataset into three parts: training dataset, validation dataset, and testing dataset [19].

F. Randomized Parameter Optimization

Randomized parameter optimization in [19] is one of the hyper-parameter techniques with randomly searching technique on the parameter in which the sample of each setting is taken from the distribution by the probability of parameter. The purpose of this technique is to obtain the highest cross validation value in the hyper-parameter domain [21].

G. Evaluation

Evaluation is an understanding on how the model works in which it is really important to use and improve text classification model [3]. To evaluate the performance of machine learning, it involves two data collections: training data and testing data. It means that the dataset must have had the label. The machine learning model is tested through the training data, then the performance can be tested through the testing data which already have had the label so as to the correctness level of machine learning can be seen on the prediction or classification related to its work. Confusion Matrix, also called as error matrix, is a specific table to visualize the work of an algorithm. It is usually used as supervised learning. Otherwise, as unsupervised learning, it is usually called as matching matrix [22] in which each row of matrix represents the predicted class and each column of matrix represents the authentic class. The following is an equation for counting the work evaluation of text classification model and TABLE.II shows Confusion Matrix [23]. The equation 4 to count True Positive Rate (TPR), recall, sensitivity, probability of detection, and power. The equation 5 is to count False Positive Rate (FPR), fall-out, probability of false alarm. The equation 6 is to count Positive Predictive Value (PPV) and precision. The equation 7 is to count Accuracy (ACC). The equation 8 is to count F1 score.

$$\text{TPR} = \frac{\Sigma\,\text{True Positive}}{\Sigma\,\text{Condition Positive}} \tag{4}$$

$$\text{FPR} = \frac{\Sigma\,\text{FP}}{\Sigma\,\text{Condition negative}} \tag{5}$$

$$\text{PPV} = \frac{\Sigma\,\text{True Positive}}{\Sigma\,\text{Predicted condition positive}} \tag{6}$$

$$\text{ACC} = \frac{\Sigma\,\text{True Positive} + \Sigma\,\text{True Negative}}{\Sigma\,\text{Total population}} \tag{7}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{PPV}\cdot\text{TPR}}{\text{PPV+TPR}} \tag{8}$$

TABLE II. *CONFUSION MATRIX* [23]

Total Population	**True Condition Positive**	**True Condition Negative**
Predicted Condition Positive	True Positive (TP)	False Positive (FP)
Predicted Condition Negative	False Negative (FN)	True Negative (TN)

IV. Findings and Discussion

TABLE.III shows the result of the preprocessing method combination of text classification. In this classification process, the dataset is divided into training data and testing data with comparison of 80 and 20. The dataset is divided randomly. Then, the training data is preprocessed based on the stages of each method used. Afterwards, the training data is filled into a pipeline. Inside the pipeline, the counting of TF-IDF vectorizer and text classification through multinomial Naive Bayes algorithm are conducted. Parameter tuning is done using randomized search cross validation. In this study, the parameter which is treated by tuning is alpha value between 0,1-1 and cross validation is treated using 3 k-fold.

The stage number 1 on the table is a basic preprocessing stage used to analyze text mining [1]. This experiment uses preprocessing method from the previous study and applies domain from this study. In fact, it meets the accuracy of 76.7%. Later, it experiments by removing stemming method on the stage number 2. In this stage, it results the accuracy of 80%. It appears that stemming method does not improve the accuracy of classification model but it tends to decrease the time during training model of 29.7% or 1.9 seconds. This experiment is used as the model of benchmark.

The stage number 3 is the experiment using standard method and adding spelling correction method applying Peter Norvig algorithm. In this stage, the accuracy value is 63.3%. In fact, this method combination tends to decrease the accuracy of 13.4% from the preprocessing method number 1. And from the point of view of the training model time, it increases 0.2 seconds.

TABLE III. RESULTS OF TEXT CLASSIFICATION ACCURACY

Number	Combination Preprocessing Methods	Train Accuration	Time Train	Tes Accuration
1	rpn+lc+rs+st	0.95	**4.5 s**	0.767
2	rpn+lc+rs	0.958	6.4 s	0.80
3	rpn+lc+sc+rs+st	0.46	4.7 s	0.633
4	rpn+lc+sc+rs	0.963	4.8 s	0.70
5	rpn+lc+sc+st	0.971	5.5 s	0.8
6	rpn+lc+sc	0.979	5.7 s	**0.867**
7	rpn+lc+sl+rs+st	0.954	5.7 s	0.75
8	rpn+lc+sl+rs	0.963	4.8 s	0.783
9	rpn+lc+sl+st	0.958	5.4 s	0.817
10	rpn+lc+sl	0.979	5.9 s	0.85
11	sc	0.983	7.6 s	**0.867**
12	sl	0.975	9.3 s	**0.867**
13	sl+sc	0.979	6.4 s	0.85

This study experiments on method number 3 and removes stemming method. In this method (method number 4), it gets accuracy value of 70% with training time of 4.8 seconds. Through this combination, this method increases the accuracy of 6.7% from the previous method (method number 3). Yet, from the point of view of the training time, it increases 0.1 second.

This study, later, experiments using method number 3 and removes stop words removal method. From this method (method number 5), this study results the accuracy value of 80% with 5.5 seconds of the training time. The combination of this method increases the accuracy of 16.7% compare to the method number 3 and increases 0.8 second of the training time.

In the last use of combination of spelling correction method, this study removes stop words removal and stemming method from method number 3. This method can be seen in the preprocessing method number 6. This method combination gains the accuracy value of 86.7% and the training time of 5.7 seconds. It seems that this method increases the accuracy of 23.4% from the method number 3. Yet, it decreases the training time of 0.8 seconds from method number 3.

From the previous discussion, the addition of spelling correction is able to decrease and increase the accuracy of the classification model. Through the method combination number 3 and 4, it seems that this method combination decreases the accuracy. Otherwise, the method combination number 6 tends to increase the accuracy. Thus, from the combination number 3 until number 6, it can be said that the addition of spelling correction increases the accuracy. Furthermore, stop words removal and stemming method decrease the accuracy. However, to increase the speed during the training time, these methods are able to contribute to decrease the training time.

Furthermore, from method number 1, this experiment adds a preprocessing method: Slang. This study, later, combines inter-methods. The first combination can be seen on the method number 7. This combination results the accuracy of 75% and the training time of 5.7 seconds. Compared to the standard preprocessing method (method number 1), it decreases in a small amount of accuracy and has a longer training time.

The next experiment is removing stemming method from the combination number 7. This method can be seen in the combination number 8. This method combination gets the accuracy of 78.3% and the training time of 4.8 seconds. Compared to the combination of accuracy value number 7, it increases the accuracy and gets a faster training time.

The next experiment is removing stop words removal method. This stage can be seen in the method combination number 9. This combination meets the accuracy value of 95.8% and the training time of 5.4 seconds. Compared to the method number 7, it increases the accuracy of 6.7% and has a faster training time.

From the method combination number 7, the next experiment is removing stop words removal and stemming method. This combination can be seen in the method combination number 10. This combination increases the accuracy value of 97.9% and the training time of 5.9 seconds. Compared to the combination number 7, it increases the accuracy but it gets a longer training time.

In the last experiment, this study tests spelling correction and Slang method, both separately and combinational. First, it tests spelling correction method (method number 11). In this method, it seems that its accuracy value is 86.7% and the training time is 7.6 seconds. Compared to the method combination number 1, the accuracy value increases but the training time becomes longer. Second, it tests Slang method (method number 12). It meets the accuracy value of 86.7% and the training time of 9.3 seconds. Compared to the method number 1, it also increases the accuracy value but gets a longer training time. The method of spelling correction and Slang increase the accuracy value compared to the standard method (method number 1), yet gets a longer training time. Both of them have a similar accuracy value but the training time of Slang method is longer than spelling correction method. Later, it combines these two methods. This method (method number 13) meets the accuracy value of 85% and the training time of 6.4 seconds. Separately compared to Slang and spelling correction method, this combination decreases the accuracy but meets a faster training time.

Based on the testing on several of these combinations, a suitable preprocessing method for classifying the review texts on hospital review by customers can be selected. If it is seen from the point of view of accuracy in the TAB.III, the method number 6, 11, and 12 are more suitable. However, from the point of view of training time, the standard method is faster during the training of classification model. From the result of this testing, spelling correction and Slang method can help to increase the accuracy value compared to the accuracy of standard method. These two methods can give a correction to the non-standard words during preprocessing stage. Thus, when the feature extraction process is conducting, this method is able to give the same vector toward the words which are similar in meaning but different in writing because of mistyping or acronym. Yet, both of these methods still lack of perfection in correcting words. It is because these methods depend on the corpus used to correct the non-standard words. In the selection of suitable combination, it can meet a high accuracy value and a faster training time.

V. Conclusion

This study experimented with several preprocessing techniques: tokenization, case folding, stop words removal, stemming, and removing punctuation and number. It also experimented by adding preprocessing techniques: spelling correction and Slang. This experiment tested each combination of those preprocessing methods. This experiment shows that spelling correction and Slang can assist to improve the accuracy value. Furthermore, the suitable selection of preprocessing technique combination is able to fasten training process so as to produce the more ideal model of text classification.

References

[1] I. M. A. Agastya, “Pengaruh Stemmer Bahasa Indonesia Terhadap Peforma Analisis Sentimen Terjemahan Ulasan Film,” Jurnal Tekno Kompak, vol. 12, no. 1, pp. 18–23, 2018.

[2] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, “A comparative Evaluation Of Pre-Processing Techniques and Their Interactions for Twitter Sentiment Analysis,” Expert Syst. Appl., vol. 110, pp. 298–310, 2018.

[3] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text classification algorithms: A survey,” Informasi, vol. 10, no. 4, pp. 1–68, 2019.

[4] J. Nothman, H. Qin, and R. Yurchak, “Stop Word Lists in Free Open-source Software Packages,” in Proceedings of Workshop for {NLP} Open Source Software ({NLP}-{OSS}), pp. 7–12, 2018.

[5] K. S. Nugroho, I. Istiadi, and F. Marisa, “Optimasi naive Bayes classifier untuk klasifikasi teks pada e-government menggunakan particle swarm optimization,” Jurnal Teknologi dan Sistem Komputer, vol. 8, no. 1, pp. 21–26f, 2020.

[6] Haniah Mahmudah, Okkie Puspitorini, Nur Adi Siswandari, Ari Wijayanti, and Eliya Alfatekha, “Metode Naive Bayes Classifier – Smoothing pada Sensor Smartphone untuk Klasifikasi Aktivitas Pengendara,” Jurnal Nasional Teknik Elektro dan Teknologi Informasi, vol. 9, no. 3, pp. 268–277, 2020.

[7] R. Rianto, A. Mutiara, E. Prasetyo, and P. Santosa, “Improving the Accuracy of Text Classification using Stemming Method, A Case of Nonformal Indonesian Conversation.”, 2020.

[8] P. Y. Saputra, D. H. Subhi, and F. Z. A. Winatama, “Implementasi Sentimen Analisis Komentar Channel Video Pelayanan Pemerintah di YouTube Menggunakan Algoritma Naïve Bayes,” Jurnal Informatika Polinema, vol. 5, no. 3, pp. 209–213, 2019.

[9] M. Zidny, “Pengaruh Semantic Expansion pada Naïve Bayes Classifier untuk Analisis Sentimen Tokoh Masyarakat,” Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 3, no. 2, pp. 141–147, 2019.

[10] A. Rakhman and M. R. Tsani, “Analisis Sentimen Review Media Massa,” Smart Computer, vol. 8, no. 2, 2019, pp. 78–82.

[11] S. Khomsah and A. S. Aribowo, “Model Text-Preprocessing Komentar YouTube Dalam Bahasa Indonesia,” Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 1, no. 10, pp. 648–654, 2021.

[12] M. S. Simanjuntak, H. Sujaini, and N. Safriadi, “Spelling Corrector Bahasa Indonesia dengan Kombinasi Metode Peter Norvig dan N-Gram,” Jurnal Edukasi dan Penelitian Informatika, vol. 4, no. 1, 2018, p. 17.

[13] A. I. Fahma, “Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance,” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 2, no. 1, 2018, pp. 53–62.

[14] E. Loper and S. Bird, “NLTK: The Natural Language Toolkit,” in In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics, 2002.

[15] N. Aliyah Salsabila, Y. Ardhito Winatmoko, A. Akbar Septiandri, and A. Jamal, “Colloquial Indonesian Lexicon,” in 2018 International Conference on Asian Language Processing (IALP), pp. 226–229, 2018.

[16] C. Sammut and G. I. Webb, Eds., “TF--IDF,” in Encyclopedia of Machine Learning, Boston, MA: Springer US, 2010, pp. 986–987.

[17] G. I. Webb, “Naïve Bayes,” in Encyclopedia of Machine Learning, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, pp. 713–714.

[18] M. Fachrie, “Machine Learning for Data Classification in Indonesia Regional Elections Based nn Political Parties Support,” Jurnal Ilmu Komputer dan Informatika (Journal Computer Scence Information), vol. 13, no. 2, pp. 89–96, 2020.

[19] F. Pedregosa et al., “Scikit-learn: Machine Learning in {P}ython,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[20] B. N. Province, “The effects of parameter tuning on machine learning performance in a software defect prediction context,” 2015.

[21] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” J. Mach. Learn. Res., vol. 13, pp. 281–305, 2012.

[22] S. V Stehman, “Selecting and interpreting measures of thematic classification accuracy,” Remote Sensensory Environment, vol. 62, no. 1, pp. 77–89, 1997.

[23] T. Fawcett, “An introduction to ROC analysis,” Pattern Recognition Lett., vol. 27, no. 8, pp. 861–874, 2006.