Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

Vol. 12, No. 1, Juli 2023

ISSN 0216 – 0544
e-ISSN 2301– 6914

SENTIMENT ANALYSIS OF LEAGUE OF LEGENDS: WILD RIFT REVIEWS ON GOOGLE PLAY USING NAÏVE BAYES CLASSIFIER

[a]Nur Sabilly, [b]Khadijah, [c]Fajar Agung Nugroho

[a,b,c]Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro
Jl. Prof. Jacub Rais, Semarang 50275, Indonesia
E-mail: nursabilly@students.undip.ac.id, khadijah@live.undip.ac.id, fajar@lecturer.undip.ac.id

Abstract

League of Legends: Wild Rift is a mobile game with more than 48 million downloads worldwide. The game publishers could earn profit from selling item in the game (in-app purchases). Therefore, it is important to understand the player opinions in order to encourage the players making the in-app purchases. This research utilized sentiment analysis to study the player opinions about the League of Legends: Wild Rift game based on the reviews on the Google Play Store using Naive Bayes Classifier. In addition, data preprocessing and feature extraction should be carried out properly to increase the performance of the classifier. Therefore, this research investigated the impact of word preprocessing stages, then compared two feature extraction algorithms, namely Term Frequency – Inverse Document Frequency (TF-IDF) and Bag of Words (BOW). From the experiment, it is found that the use of stemming could decrease the accuracy of the classifier, while the use of transformation of non-standard words into standard words could improve the performance of the classifier, for both BOW and TF-IDF. In this case, BOW is better than TF-IDF. BOW could reach the best accuracy of 79.3%, while TF-IDF reach the best accuracy of 78.4%.

Key words: BOW, Game reviews, Naive Bayes Classifier, Sentiment Analysis, TF-IDF.

INTRODUCTION

The interest of Indonesian people in mobile online games has increased since 2018 until the end of the 2021 quarter. There has been an increase of 59.70% in the number of online game downloads on app stores like the App Store and Google Play Store [1]. League of Legends: Wild Rift as a popular free-to-play game has been installed on more than 48 million smartphones worldwide as of September 23, 2022. With the large number of game players, there is a great opportunity to gain profit from players, such as by selling items in the game which is also known as in-app purchases.

There are several main factors that determine the tendency of players to purchase items in the game, such as the player performance, the player's first impression of the game in the first week, and social interactions, such as player communities in the game [2]. Therefore, game publishers should monitor the behavior of players as an effort to encourage players to make in-app purchases. The reviews given by the game player in the application store, such as Google Play Store are important resource to find out player satisfaction and complaints about the game. The process of drawing conclusions from review data must be done accurately and quickly so that it can provide the right business decisions [3]. However, with a large number of review data, reading all reviews conventionally one by one will take a lot of time and there is a risk of subjectivity in the conclusions obtained [4].

Sentiment analysis is a branch of natural language processing which is aimed to analyze

text to determine the sentiment polarity of the given text, such as possitive, negative, or neutral. Sentiment analysis is very useful to analyze game player's opinion from text review. Thus, the polarity of the player sentiment about that game can be recognized. There are some approaches dealing with sentiment analysis, namely lexicon based, machine learning, and hybrid of both. However, the main problem with the lexicon base approach is domain dependence [5].

There are several machine learning algorithms commonly used for sentiment analysis such as Support Vector Machine (SVM), Random Forest Classifier (RFC), Naive Bayes Classifier (NBC), and Artificial Neural Networks (ANN). Previous research applied the SVM algorithm for sentiment analysis on provider by.U application reviews and obtained an accuracy of 83.30% [7]. However, the SVM algorithm is less efficient for large data classification because it needs to solve quadratic programming to find the hyperplane that causes complexity in computation [7].

NBC is a probabilistic classification algorithm based on Bayes Theorem by assuming the conditional independence among the data features. NBC predict the probability of a particular class label given the set of data features [5]. NBC is also able to achieve competitive result with lower computational cost [8]. Previous sentiment analysis studies using NBC algorithm has proven that NBC were able to achieve the accuracy of 86.76% on m-banking application reviews data [9] and the accuracy of 84.80% on smartphone product reviews data [10]. In addition, the NBC algorithm also has many advantages compared to the ANN algorithm, such as good generalization ability, simple computation, and the ability to handle incomplete data by assuming that each attribute is independent [11], [12].

The comparison of NBC and RFC has been carried out to perform sentiment analysis task on 3000 news headlines data. The study showed that the RFC algorithm required 250 times higher computation time than the NBC algorithm for the same accuracy [13]. Another study that compare NBC, decision tree, and RFC algorithm has also been carried out to perform sentiment analysis on tweet related to Anti-LGBT campaign. The result of the experiment shows that NBC is able to achieve the highest accuracy [14]. NBC algorithm has also been applied to perform sentiment analysis of precidential candidate of Republic Indonesia using data from Twitter. In that case, NBC could achieve the highest accuracy compared to SVM and K-NN (Nearest Neighbor) [15].

In addition to classification algorithm, data preprocessing and feature extraction have to be carried out properly to increase the accuracy of the classifier. Data preprocessing is crucial in data mining problems to improve the quality of data that can affect the performance of classification models [16], [17], [18]. The data preprocessing stages in the text domain involves cleansing, normalization, tokenization, and stemming. Stemming is known to enhance the performance of text classification models [9], [10], [12]. In addition, social media brings new challenges in natural language processing due to the emergence of new informal vocabulary. Such informal vocabulary needs to be converted into standardized forms to reduce the dimensionality of the model. This has been found to improve the performance of text classification models [19].

Feature extraction plays important role in sentiment analysis task which is aimed to extract the useful information from the text. In sentiment analysis, there are several popular algorithms for feature extraction, namely Bag of Words (BOW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word2Vec [5]. Previous research compared these algorithms on several models, namely RFC, SVM, and NBC algorithms using the Amazon Food review dataset. The results of experiment showed that TF-IDF can maximize the performance of NBC-based models with an accuracy of 80.60%, outperforming BOW, which only has an accuracy of 76.90% [20].

This research utilized sentiment analysis to study the player opinions about the League of Legends: Wild Rift game based on the reviews given by the players on the Google Play Store. The sentiment analysis was applied by using NBC algorithm which was well known for achieving good accuracy in the sentiment analysis task. The contribution of this research is producing a model that could be used to do sentiment analysis on Legends: Wild Rift game review data automatically. In addition, this research also investigated the impact of

stemming and transforming informal vocabulary into standardized forms in the preprocessing stage on the performance of NBC models. This research also compared two feature extractions algorithm, namely TF-IDF and BOW to determine which algorithm produces the best performance in the NBC models.

MATERIAL AND METHODS

This research performed sentiment analysis on the League of Legends: Wild Rift game review data from Google Play Store by using the Naive Bayes Classifier (NBC) algorithm. The methodology of this research is shown in Figure 1. There are several stages in this research methodology starting from data collection to model evaluation. The explanation of each stage is provided in the following sub sections.

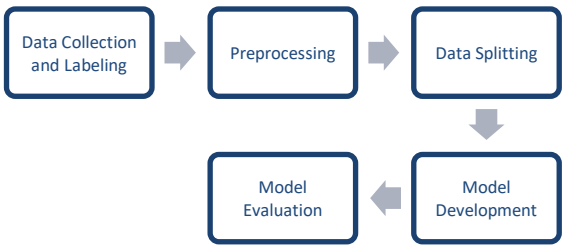

Fig 1. Research methodology

Data Collection and Labeling

This research used the League of Legends: Wild Rift game review data on the Google Play Store as the dataset. The data was collected by scraping the review column of the League of Legends: Wild Rift game download page on the Google Play Store website. Scraping was performed by using the google-play-scraper library in the Python programming language. From the scraping process, a total of 138,009 review data were collected. Each review has rating score from 1, 2, 3, 4, to 5. Rating score of 5 shows the best score, while rating score of 1 shows the worst score. The proportion of review data in each rating score are 31.90% of rating 1, 3.30% of rating 2, 4.70% of rating 3, 4.90% of rating 4, and 55.02% of rating 5.

Data labeling was conducted automatically by applying a mapping function from score sets to sentiment sets in order to avoid the subjectivity interpretation. The score set has a domain of {1,2,3,4,5} which represents the rating score in each review, while the sentiment set has a domain of {0,1} which represents the sentiment of each review that is used as the target class for classification. A sentiment value of 0 indicates a negative sentiment, while a sentiment value of 1 indicates a positive sentiment. A review with the rating score of 1 or 2 was mapped to sentiment value of 0, while a review with the rating score of 4 or 5 was mapped to sentiment value of 1. After each review is mapped, there are 104.069 reviews with 62.411 reviews (60%) belong to positive sentiment and 41.658 reviews (40%) belong to negative sentiment.

Preprocessing

There are four stages carried out in the data preprocessing, namely cleansing, tokenization, normalization of words, and stopword removal. Cleansing is the process of cleaning text data from dirty data based on certain rules [10]. Tokenization is the process of splitting sentences into word tokens. The implementation of tokenization in this research was carried out by using a function, namely word_tokenize provided by Natural Language Toolkit (NLTK) library. Normalization of words is the process of converting each word/token into the standard form [21]. In this reserach, the normalization of word conducted consist of converting each word into lowercase (case-folding), transforming non-standard words into standard words, and mapping into basic words without any affixes (stemming). Transforming non-standard words into standard words is aimed to transform the non-standard or slang word into standard word by using a corpus having the list of 3,592 pairs of slang word and standard word [19]. Stemming process was implemented by using Sastrawi library. Sastrawi stemmer was selected because previous research shows that Sastrawi was able to achieve better performance compared to Nazief-Adriani's algoritm and Arifin Setiono's algorithm [22]. The last stages in the text preprocessing, stopword removal, is the process of removing meaningless words in the Indonesian language using a stopword list from the NLTK library. After all stages in the text preprocessing were carried out, the total number of review data was reduced to 101,554.

Data Splitting

Data splitting is aimed to split data into two parts, namely training data and testing data. In this research the dataset of 101,544 reviews was

divided into 80% of training data (81,243) and 20% of testing data (20,231). Training data was used to train the classifier model, while the testing data was used to evaluate the resulting classifier model from the training process. The subset data used in training and testing process must be different to obtain the fair result and avoid bias in the model evaluation [23].

Model Development

This stage is aimed to develop a model to perform sentiment analysis. The model development consists of features extraction and classification. In this research two feature extraction algorithm, namely Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BOW) were utilized to find the best result in the classification model. Then, the classification was performed by using NBC algorithm. The version of NBC algorithm depends on the resulting feature extraction. When applied TF-IDF feature extraction, the domain of the extracted features is in continuous value, therefore the Gaussian Naive Bayes (GNB) algorithm was used as classification algorithm. When BOW feature extraction was applied, the domain of the extracted features is in discrete value, therefore the Multinomial Naive Bayes (MNB) algorithm was used as classification algorithm.

Bag of Words

BOW is a set of words that only stores the word frequency in the document while ignoring the word order. The vector representation of BOW can be created by inserting the frequency of each unique word in the dataset. Equation (1) shows the representation of BOW feature extraction where $tf_{t,d}$ is the term frequency of term t in document d [21].

$$tf_{t,d} = count(t, d) \quad (1)$$

The bag of words does not have to consist of every unique word, but can consist of two or more words (n-gram). The n-gram model is used as a way to break sentences into tokens in the feature extraction process. Word n-grams are n-word tokens taken from a sentence or a corpus of data. It is called unigram, bigram, or trigram when the size of n in n-gram model are 1, 2, or 3, respectively [5].

Term Frequency-Inverse Document Frequency

TF-IDF is a word weighting method by calculating the values of word frequency and document inverse frequency. Therefore TF-IDF not only considers the word frequency, but also the word relevance in the context of the document. This method is widely used for information retrieval and text mining. In the TF-IDF feature extraction, there are three steps to calculate the weight of each word in the dataset. The first step is to calculate the TF of each word as shown by equation (1). Then, the value of IDF can be calculated by using equation (2) where idf_t represents the IDF of term t in document d and df_t represents the number of documents in which the word or term t appears. The IDF of words that appear rarely in the documents will have the higher value compared to the IDF of word that appear frequently in the documents. The words that appear frequently in a document are assumed to have less significant meaning. Document frequency df_t serves as a divider to calculate the IDF. Finally, the word weight or TF-IDF can be calculated by multiplying TF and IDF as shown by equation (3) where $w_{t,d}$ is the TF-IDF weight for term t in document d [21].

$$idf_t = ln\left(\frac{N}{df_t}\right) + 1 \quad (2)$$

$$w_{t,d} = tf_{t,d}.idf_t \quad (3)$$

Naive Bayes Classifier

The extracted feature dataset can be used as features for training the NBC model. NBC is a type of probabilistic classification algorithm based on Bayes theorem. In the Bayes theorem a posterior probability of hypothesis H given evidence $\boldsymbol{X}$ can be calculated from the prior probability of hypothesis H, the prior probability of evidence $\boldsymbol{X}$, and the posterior probability of evidence $\boldsymbol{X}$ given hypothesis H, also called likelihood, as shown by equation (4). In the sentiment analysis task, evidence $\boldsymbol{X}$ refers to the vector of data features, where hypothesis H refers to the probability that **X** belongs to a certain class C.

$$P(H|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|H)P(H)}{P(\boldsymbol{X})} \quad (4)$$

Suppose there are two target classes in this sentiment analysis task, namely C_1 (possitive) and C_2 (negative), NBC algorithm predict $\boldsymbol{X}$ belongs to class C_1 or C_2 which has the highest posterior probalility, conditioned on $\boldsymbol{X}$. Therefore, the NBC calculates the posterior probability belongs to class C_i given a set data features $[x_1, x_2, \dots, x_n]$ as equation (5) for each

class based on Bayes theorem in equation (4). Since $P(x_1, x_2, \dots, x_n)$ is same in each class, the equation (5) can be simplified as equation (6).

$$P(C_i|x_1, x_2, \dots, x_n) = \frac{P(x_1, x_2, \dots, x_n|C_i)P(C_i)}{P(x_1, x_2, \dots, x_n)} \quad (5)$$

$$P(C_i|x_1, x_2, \dots, x_n) = P(x_1, x_2, \dots, x_n|C_i)P(C_i) \quad (6)$$

Prior probabilities can be calculated by using equation (7) where N_{C_i} is the number of documents with class C_i and N is the total number of documents.

$$P(C_i) = \frac{N_{C_i}}{N} \quad (7)$$

Since the data consist of many features, the computation of $P(x_1, x_2, \dots, x_n|C_i)$ would be extremely expensive. Therefore, NBC made a naive assumption of class conditional independence which means that there are no dependencies among the features given the class label. Then, the value of likelihood can be simply calculated as equation (8) [23].

$$P(x_1, x_2, \dots, x_n|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \quad (8)$$

The likelihood then can be estimated from the available training data. The calculation of the likelihood will depend on the type of data feature. If the data feature is discrete, then the MNB is used, while if the data feature is continous then, the GNB is used.

In this research MNB is used when the dataset is extracted by using BOW feature extraction algorithm. BOW simply count the term frequency so that the extracyed features are in discrete value. MNB algorithm calculates the likelihood with smoothing parameter α as shown by equation (9) where $P(x_k|C_i)$ represents the likelihood of feature or term x_k given class C_i, V represents the vocabulary or all unique words in the corpus, and α represents the smoothing parameter between 0 and 1 [21].

$$P(x_k|C_i) = \frac{count(x_k, C_i) + \alpha}{(\sum_{x \in V} count(x, C_i)) + \alpha|V|} \quad (9)$$

GNB is used when the dataset is extracted by using TF-IDF feature extraction. The TF-IDF calculate the weight of each term in the certain document which is in continuous value. GNB assumes that the distribution of attribute value follow the Gaussian or normal distribution. Therefore, to compute the likelihood, the mean and standar deviation of each feature or attribute must be calculated. GNB algorithm calculates the likelihood of feature or term x_k given class C_i, $P(x_k|C_i)$ as shown by equation (9), where $\sigma^2_{x_k,C_i}$ is the variance of the weight of term x_k with respect to class C_i, μ_{x_k,C_i} is the mean of the weight of term x_k with respect to class C_i, and $w_{x_k,d}$ is the TF-IDF weight of term x_k with respect to document d [25].

$$P(x_k|C_i) = \frac{1}{\sqrt{2\pi\sigma^2_{x_k,C_i}}} exp\left(-\frac{\left(w_{x_k,d} - \mu_{x_k,C_i}\right)}{2\sigma^2_{x_k,C_i}}\right) \quad (10)$$

Model Evaluations

Model evaluation is performed to measure the performance of the resulting model to classify the sentiment of review data. The evaluation or testing was carried out on 20% of the testing data from the entire dataset. This research used the confusion matrix to evaluate the performance of classifier or model. From the confusion matrix, several evaluation metrics could be obtained, namely accuracy, precision, recall, and f-measure.

RESULT AND DISCUSSION

There are several scenarios in this research experiment to observe the effect of preprocessing and feature extraction toward the result of classification using Naïve Bayes Classifier (NBC) algorithm. First, the experiment is divided into two main scenarios. The first scenario implemented BOW as feature extraction and MNB as classification algorithm, while the second scenario implemented TF-IDF as feature extraction and GNB as classification algorithm. Subsequently, in each scenario we compare between using stemming and ignoring stemming in the preprocessing step, and we also observe the effect of adding the transformation of informal words into the formal words in the preprocessing step. In each feature extraction process, we also experimented several n-gram model, namely unigram, bigram and trigram. Except for the second scenario when adding the transformation of informal word into the formal words in the preprocessing step, the bigram and trigram are excluded because of the limitation of computation. All experiments were run on Google Colab and Kaggle environment.

For classification, there is an α parameter in the NBC algorithm. The α parameter in the MNB algorithm is a smoothing constant which can be defined as a value added to overcome zero probability problems. Meanwhile, the α

parameter in the GNB algorithm is smoothing parameter which is defined as a percentage, how much the largest variance value of each feature is used to overcome the problem of division by 0. The range of values used as a sample point is 0 to 1. From this range of values, it is divided into 200 sample points. These 200 samples are placed into an array data structure and looped 200 times (iteration based on the number of array elements). In each loop, both NBC algorithm-based models are trained with α parameter values that correspond to the index value of the array containing 200 of α samples. The results of experiment when using BOW and TF-IDF feature extraction with the best value of α parameter found for each scenario are shown by Table 1 and Table 2, respectively.

Table 1. The results of experiment using BOW feature extraction and MNB classification algorithm

Preprocessing	N-gram	α	Accuracy	Precision	Recall	F1-Score
Stemming	Unigram	1.000	0.779	0.806	0.831	0.818
	Bigram	0.939	0.735	0.734	0.873	0.798
	Trigram	1.000	0.637	0.633	0.937	0.755
Non-stemming	Unigram	1.000	0.788	0.820	0.827	0.824
	Bigram	1.000	0.709	0.705	0.886	0.785
	Trigram	0.879	0.619	0.618	0.948	0.749
Transformation of informal words into formal words	Unigram	0.919	0.793	0.821	0.835	0.828
	Bigram	0.934	0.790	0.801	0.863	0.831
	Trigram	0.735	0.720	0.713	0.892	0.793

Table 2. The results of experiment using TF-IDF feature extraction and GNB classification algorithm

Preprocessing	N-gram	α	Accuracy	Precision	Recall	F1-Score
Stemming	Unigram	0.437	0.764	0.763	0.879	0.817
	Bigram	0.055	0.527	0.709	0.402	0.523
	Trigram	0.020	0.341	0.674	0.048	0.174
Non-stemming	Unigram	0.437	0.764	0.763	0.879	0.817
	Bigram	0.054	0.687	0.834	0.562	0.683
	Trigram	0.190	0.501	0.834	0.208	0.334
Transformation of informal words into formal words	Unigram	0.713	0.784	0.802	0.848	0.824

The selection of the value of n for n-gram in the feature extraction process affects the performance of the model. Based on Table 1 and Table 2, choosing a smaller value of n in n-gram model results in better performance, both when using BOW and TF-IDF feature extraction. A high value of n will result in a sparse matrix or vector representation in feature extraction because the resulting features will contain more variations of data. Although theoretically, the use of a high value of n can capture the context of words better, it can cause overfitting because the occurrence of n-grams seen during testing will be rarer.

In the case when BOW feature extraction were used, generally ignoring stemming may produce higher performance than using stemming when using unigram model. Otherwise, the using of stemming result in higher performance than ignoring stemming when using the higher model of n-gram model, namely bigram and trigram. However, in the case when TF-IDF feature extraction were used, ignoring stemming produce higher performance than using stemming in all n-gram model (unigram, bigram, and trigram). Then, adding transformation of informal words into formal words may increase the performance of the model, both when using BOW and TF-IDF feature extraction. Including this tranformation may eliminate the number of informal words with have the same meaning with the formal words.

The use of feature extraction can affect the performance of the model as shown in Figure 2. The model built by using BOW feature extraction resulted in higher accuracy than the model by using TF-IDF feature extraction generally in all preprocessing and n-gram scenario. In the TF-IDF feature extraction

process, the weighting of each token is based on how often the token appears in the document and how rarely the token appears in the corpus of documents. Tokens that frequently appear in the document but rarely appear in the corpus have a greater weight, indicating that they play an important role in the classification process.

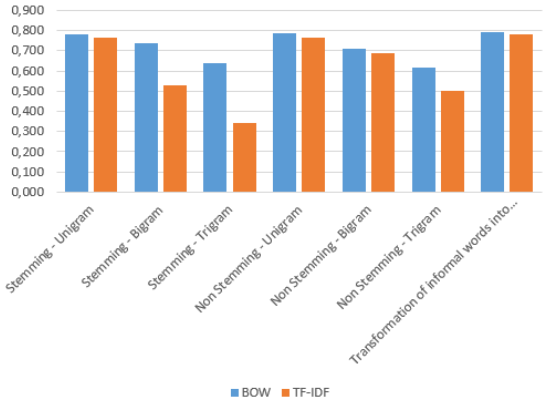

Fig 2. The comparison of accuracy between BOW and TF-IDF feature extraction

On the other hand, BOW simply counts the occurrence of each word in a document and ignores the context in which the word appears. This is in line with the nature of the NBC algorithm, which ignores sentence context and considers all features as independent. Finally, the best model was achieved when not using stemming, applying the transformation of informal words into formal words, and using BOW unigram feature extraction, with the accuracy of 79,3%, precision of 82.10%, recall of 83.50%, and f1-score of 82,8.10%.

CONCLUSION

In this research, the best model was selected for sentiment analysis of League of Legends: Wild Rift game review data on Google Play using the Naïve Bayes Classifier (NBC) algorithm. Several conclusions were obtained from the study. Firstly, the transformation process of non-standard words into standard words improved the performance of the NBC algorithm model in both feature extractions, namely BOW and TF-IDF. Secondly, the stemming process on the preprocessing dataset generally resulted in lower NBC algorithm classification performance in most experiments when using both TF-IDF and BOW features. And thirdly, the test results showed that the BOW feature extraction produced better NBC algorithm model performance than the TF-IDF feature extraction.

REFERENCES

[1] data.ai, “State of Mobile 2022,” San Francisco, California, USA, 2021. Accessed: Sep. 25, 2022. [Online]. Available: https://www.data.ai/en/go/state-of-mobile-2022/

[2] Y. Jiao, C. S. Tang, and J. Wang, “An empirical study of play duration and in-app purchase behavior in mobile games,” Prod Oper Manag, vol. 31, no. 9, pp. 3435–3456, Sep. 2022, doi: 10.1111/POMS.13772.

[3] T. U. Haque, N. N. Saber, and F. M. Shah, “Sentiment analysis on large scale Amazon product reviews,” 2018 IEEE International Conference on Innovative Research and Development, ICIRD 2018, pp. 1–6, Jun. 2018, doi: 10.1109/ICIRD.2018.8376299.

[4] S. W. H. Kwok, S. K. Vadde, and G. Wang, “Tweet Topics and Sentiments Relating to COVID-19 Vaccination Among Australian Twitter Users: Machine Learning Analysis,” J Med Internet Res, vol. 23, no. 5, May 2021, doi: 10.2196/26953.

[5] M. Wankhade, A.C.S. Rao, and C. Kulkarni, “A Survey on Sentiment Analysis Methods, Applications, and Challenges“, Artifcial Intelligence Review (2022) 55:5731–5780.

[6] S. Fransiska, Rianto, and A. I. Gufroni, “Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method,” Scientific Journal of Informatics, vol. 7, no. 2, pp. 2407–7658, 2020.

[7] X. Li and W. Yu, “Fast Support Vector Machine Classification for Large Data Sets,” International Journal of Computer Intelligent System, vol. 7, pp. 197–212, 2014.

[8] F. Xu, Z. Pan, and R. Xia, E-Commerce Product Review Sentiment Classification Based on A Naïve Bayes Continuous Learning Framework, Information Processing and Management 57 (2020)

102221.

[9] M. E. Permana, H. Ramadhan, I. Budi, A. B. Santoso, and P. K. Putra, "Sentiment analysis and topic detection of mobile banking application review," in 2020 5th International Conference on Informatics and Computing, ICIC 2020, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. doi: 10.1109/ICIC50835.2020.9288616.

[10] M. A. M. Salem and A. Y. A. Maghari, "Sentiment analysis of mobile phone products reviews using classification algorithms," in Proceedings - 2020 International Conference on Promising Electronic Technologies, ICPET 2020, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 84–88. doi: 10.1109/ICPET51420.2020.00024.

[11] D. Jurafsky and J. H. Asghar, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd ed., . 2nd ed., London: Pearson, 2009.

[12] A. K. Sharma, S. Kumar Prajapat, and M. Aslam, "A Comparative Study between Naïve Bayes and Neural Network (MLP) Classifier for Spam Email Detection," International Journal of Computer Applications (IJCA), pp. 12–16, 2014.

[13] A. Singh, M. N. Halgamuge, and R. Lakshmiganthan, "Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms," 2017.

[14] V.A. Fitri, R. Andreswari, M.A. Hasibuan, "Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm", Procedia Computer Science 161 (2019) 765–772.

[15] M. Wongkar and A. Angdresey, "Sentiment Analysis Using Naive Bayes Algorithm of The Data Crawler: Twitter", Proceedings of The Fourth International Conference on Informatics and Computing (ICIC), 16-17 October 2019, Semarang, Indonesia, doi: 10.1109/ICIC47613.2019.8985884.

[16] A. Y. Maghari and J. H. Zendah, "Detecting Significant Events in Arabic Microblogs using Soft Frequent Pattern Mining," Journal of Engineering Research and Technology, vol. 6, no. 1, Mar. 2019, Accessed: Oct. 03, 2022.

[17] G. Gautam and D. Yadav, "Sentiment analysis of twitter data using machine learning approaches and semantic analysis," 2014 7th International Conference on Contemporary Computing, IC3 2014, pp. 437–442, 2014, doi: 10.1109/IC3.2014.6897213.

[18] B. A. Alhaj and A. Y. A. Maghari, "Predicting User Entries by Using Data Mining Algorithms," Proceedings - 2017 Palestinian International Conference on Information and Communication Technology, PICICT 2017, pp. 110–114, Sep. 2017, doi: 10.1109/PICICT.2017.24.

[19] N. A. Salsabila, Y. A. Winatmoko, A. A. Septiandri, and A. Jamal, "Colloquial Indonesian Lexicon," Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018, pp. 226–229, Jan. 2019, doi: 10.1109/IALP.2018.8629151.

[20] F. Alzami, E. D. Udayanti, D. P. Prabowo, and R. A. Megantara, "Document Preprocessing with TF-IDF to Improve the Polarity Classification Performance of Unstructured Sentiment Analysis," Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, pp. 235–242, Aug. 2020, doi: 10.22219/kinetik.v5i3.1066.

[21] Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition First Edition.

[22] D.Mustikasari, I. Widaningrum, R. Arifin, W.H.E. Putri, "Comparison of Effectiveness of Stemming Algorithms in Indonesian Documents", Proceedings of the 2nd Borobudur International Symposium on Science and Technology (BIS-STE 2020), Atlantis Press, 154-158.

[23] S. Haykin, "Neural Ntworks and Learning Machines Third Edition", Pearson Education, New Jersey, 2009.

[24] J. Han, M. Kamber, and J. Pei, 2012, "Data Mining Concepts and Techniques Third Edition", Morgan Kaufmann, Waltham.

[25] Bishop, C. M. (2006). Pattern Recognition and Machine Learning (M. Jordan, J.