Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol 7, No.1, March 2024, pp. 39 – 46
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Sentiment Analysis and Topic Modelling on Crowdsourced Data

[1]Maria A. Hasiholan Siallagan, [2]Arie Wahyu Wijayanto
[1,2]Politeknik Statistika STIS
Email: [1]212011557@stis.ac.id, [2]ariewahyu@stis.ac.id

Article Info

Article history:
Received Sep 12th, 2023
Revised Nov 20th, 2023
Accepted Dec 18th, 2023

Keyword:
Crowdsourced Data
Latent Dirichlet Allocation
Sentiment Analysis
Topic Modelling

ABSTRACT

User reviews on applications are one form of crowdsourced data that can effectively capture the satisfaction levels of application users. However, user reviews often appear messy and contain various and abstract topics. Therefore, they need to be processed first to provide useful information for decision-makers. This study focuses on organizing and classifying application reviews by using machine leaning-based sentiment analysis with various classification algorithms, including Logistic Regression, Support Vector Machines, and Random Forest. Additionally, to address negative sentiment labels, topic modeling is conducted using Latent Dirichlet Allocation (LDA). This study demonstrates that the best sentiment classification model is logistic regression, achieving an average accuracy of 0.925 and an average F1-score of 0.763. Furthermore, the LDA analysis successfully generates topic models for negative reviews, revealing three key topics: price-related issues, accessibility concerns, and application accuracy, all of which require reevaluation and potential improvement.

Corresponding Author:
Maria A. Hasiholan Siallagan,
Departement of Statistics,
Politeknik Statistika STIS,
Jl. Otto Iskandardinata No.64C, Jakarta, Indonesia
Email: 212011557@stis.ac.id

DOI: http://dx.doi.org/10.24014/ijaidm.v7i1.24777

1. INTRODUCTION

The advancement of technology in data collection has led to the large-scale data generation contributed by individuals, known as crowdsourced data. This evolution has provided the potential for society to gain insights in helping the decision-making process through data. One implementation of using crowdsourced data for decision-making is to utilize application reviews to reveal users' satisfaction levels. Consumer satisfaction levels assist application developers in enhancing their quality and improving the shortcomings of their applications.

However, reviews data is often unstructured and large in quantity. As a result, many application developers become fatigued reading extensive text documents, leading them to potentially ignore significant parts and not grasp the overall picture of the reviews. Sentiment analysis, as a technique in Natural Language Processing (NLP), determines the sentiment or emotional meaning behind textual data [1]. It involves the examination of opinions, attitudes, emotions, and sentiments expressed in a written piece [2].

In addition, reviews need to be grouped to provide insights about the application. Positive opinions can indicate user satisfaction and compatibility with an application, while negative opinions can indicate issues or shortcomings that need to be addressed. Sentiment analysis can classify data into positive, neutral, or, negative sentiments [3] by transforming textual data into numerical features. The classification model is subsequently employed to categorize future review data, to help the policy-making process.

Furthermore, reviews often contain various and latent topics—abstract topics that are not directly observable. Therefore, the primary challenge is identifying the most crucial segments of the text and distinguishing them from less relevant ones [4]. LDA can address these problems by revealing latent topics

hidden within the thematic structure of the textual dataset [5] and discover the most frequently discussed topics across all reviews. Topic modeling is an innovative method designed to generate keyword-based representations of documents [6]. These keywords are utilized during indexing and document searching to enable easy retrieval based on user requirements.

Previous studies utilized the topic modeling LDA to identify the most frequently discussed topics in the data [6], [7]. Research [8] classified Shopee user reviews into positive or negative opinions. Meanwhile, the study by [9] employed a combination of topic modeling and lexicon-based sentiment analysis to gain interesting insights into user reviews. Lexicon-based analysis involves various words evaluated with polarity scores to discern user responses regarding a specific topic. However, its drawback lies in the exclusion of many words not present in the lexicon, and unable to identify sarcasm, negation, grammar mistakes, misspellings, or irony [10]. Machine learning-based sentiment analysis can address these limitations [11]. Therefore, this research adopts a combination of topic modeling and machine learning-based sentiment analysis on application reviews to ascertain the sentiment expressed in the reviews and identify areas for improvement in the application.

2. METHODOLOGY

The data used consists of all reviews from the Ask AI-Chat with Chatbox application in Indonesian language. The reviews used are those that first appeared, starting from March 8, 2023, until the end of the study period on May 25, 2023. The reviews were obtained by conducting data scraping using Python with the *google_play_scraper* package. The research diagram is presented in Figure 1.

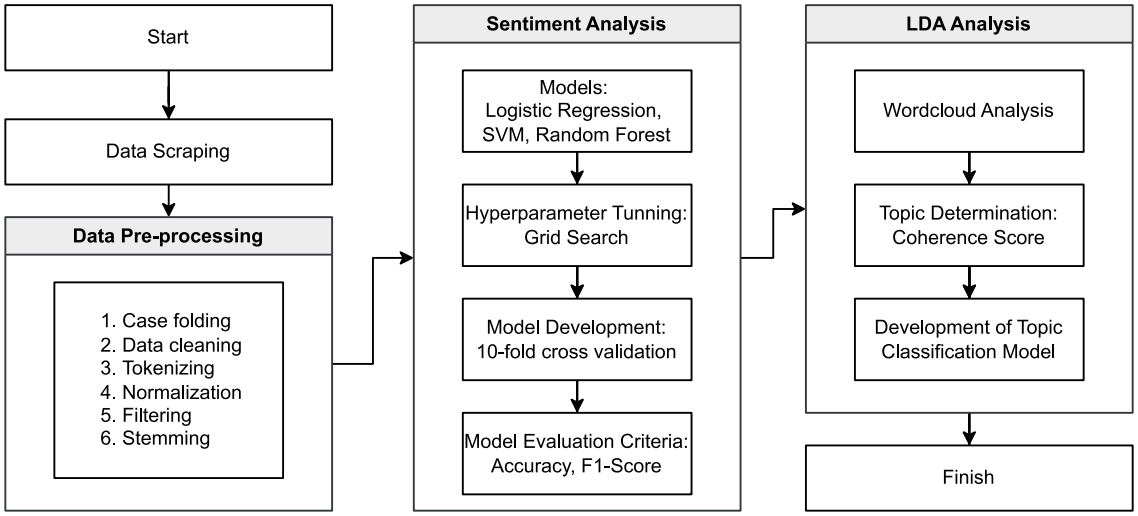

Figure 1. Research Diagram

This research employs machine learning-based sentiment analysis and topic modelling analysis. Sentiment analysis is aimed at classifying review data based on sentiments. The classification techniques applied include logistic regression, SVM, and Random Forest. Furthermore, LDA is employed as a topic modelling tool to extract most discussed topics from the review data. LDA is applied only to reviews labeled as negative to extract the necessary insights for application developers to enhance the quality of their app.

2.1. Logistic Regression

Logistic regression performs probabilities estimating a binary outcome based on an input variable [12]. In the first step, the data is converted into numerical features using the bag-of-words technique, which creates a vector representation for each document by counting the frequencies of its constituent words. After obtaining this numerical representation, logistic regression is applied to estimate the probability of each document belonging to a specific class. The model calculates weights based on the numerical features and employs these weights to predict the labels of the documents. The model's objective is to find the optimal weights by minimizing the cross-entropy loss function, which quantifies the dissimilarity between the predicted probabilities and the actual labels.

$$\hat{\theta} = argmin_{\theta} \frac{1}{m} \sum_{i=1}^{m} \mathrm{L_{CE}}(\mathrm{y}^{(i)}, \mathrm{x}^{(i)}; \theta) \quad (1)$$

2.2. Random Forest

Random Forest composed of multiple tree-structured classifiers $\{h(x, \Theta_k), k = 1, \dots\}$, where the $\{\Theta_k\}$ are random vectors that are independently and identically distributed [13]. Each classifier independently predicts the most popular class for a given input X and contributes with a unit vote towards the final classification. Random Forest combines multiple decision trees to enhance classification accuracy. In the context of text classification, this model creates multiple decision trees by training them on various subsets of the feature space. Subsequently, it amalgamates all the prediction results from these trees to reach the final classification decision.To develop a text classification model using the random forest algorithm, the following steps are undertaken[14].

1. First, prepare the text vector set by preprocessing the text dataset, creating a Text Vector Space Model (VSM) for the random forest algorithm.
2. Next, construct the random forest text classifier using the Bagging method, which involves creating multiple training sets (nTree) from the original set (D) of size N. For each of these nTree training sets, a Classification and Regression Tree (CART) is built without pruning, following these steps
   a. We assume there are M primitive attributes and select a positive integer mtry, using hyperparameter tunning. Throughout the process of generating the forest, mtry remains constant.
   b. At each internal node of the CART, a subset of mtry attributes is randomly chosen from the original M attributes as candidate attributes for creating the split node.
   c. The Gini index is then used to identify the best attribute among these mtry candidate attributes for splitting the node.
   d. The trees in the forest are grown fully to obtain the maximum tree, Tmax. Leaf nodes in this maximum tree are very small, representing either pure nodes (containing samples of the same class) or branches where no further attributes exist. A node is considered very small if the number of samples within it falls below a given threshold. The maximum tree, Tmax, is not pruned during this process.
3. Utilize the classifiers. The classifier's output is determined using the majority vote method.

$$\mathrm{c} = argmax_c(\frac{1}{ntree} \sum_{k=1}^{ntree} \mathrm{I}(\mathrm{h}(\mathrm{x}, \theta_\mathrm{k}) = \mathrm{c})) \quad (2)$$

2.3. Support Vector Machine

SVM seeks to identify the optimal hyperplane (decision boundary) that effectively separates distinct classes [15]. To achieve this, the review data is converted into vectors and given weights using Term Frequency-Inverse Document Frequency (TF-IDF) [16]. TF-IDF evaluates the significance of words in a review by calculating a score, which is derived by multiplying the Term Frequency (TF) of the word by its Inverse Document Frequency (IDF). This process helps to emphasize important words while de-emphasizing common ones, thus aiding in the classification task. TF-IDF formulated as below [17]

$$TF - IDF_{t,d} = TF_{t,d} \times IDF_t \quad (3)$$

$TF_{t,d}$ is the number of times the term ‘t’ appears in a document ‘d’. In contrast, the IDF represents the proportion of documents in the corpus that contain the term. N represents the total count of documents in the collection, while DF_t denotes the number of documents in the collection that include the term 't'.

$$IDF_t = \log\left(\frac{N}{DF_t}\right) \quad (4)$$

Descriptive function of SVM is as follow[18].

$$h(X) = z^x \emptyset(\mathrm{X}) + \mathrm{c} \quad (5)$$

In this context, X denotes the feature vector, while z indicates a vector representing various weights. The non-linear mapping function Ø is responsible for transformations, and c represents the bias vector. Both z and c are capable of automatic learning from the training dataset.

2.4. Latent Dirichlet Allocation

Subsequently, this research employs an unsupervised learning approach, LDA analysis, for topic modeling of all negative reviews. LDA operates on the assumption that every document comprises a blend of hidden topics, and each topic represents a probability distribution of words. This means that each topic has characteristic features based on the distribution of words [5]. When provided with a corpus of documents, the LDA model computes the topic distribution for each document and the word distribution for each topic [19]. This algorithm is useful for summarizing, classifying, connecting, and processing large datasets as it can reveal significant topics within each document [20]. The stages of topic modeling in this research are as follows.

1. Creating a dictionary and corpus. The dictionary contains a collection of unique words indexed. The corpus contains the composition of words and their frequency of occurrence.
2. Determining the number of topics(K) by evaluating the effectiveness in grouping topics from coherence score. A small value of K leads to topics that are overly general, while a large value of K results in uninterpretable topics [21]. Topic Coherence score measures the cohesion of a single topic by assessing the level of semantic similarity among the highly scored words within that topic [22].
3. Implementing LDA (Latent Dirichlet Allocation) using *Gensim* from Phyton. *Gensim* employs an online LDA technique, known as variational inference [23], to approximate the posterior distribution.
4. Evaluate the model. The evaluation of the topic modeling is based on topic coherence. A good model will have a high coherence score for its topics. The topics are deemed coherent when majority of their words exhibit strong assosiations [21].

3. RESULT AND DISCUSSION

3.1. Preprocessing Data

The initial stage of this analysis involved manual labeling of the collected data. Sentiment classification was performed as binary classification, with label "1" indicating positive sentiment and label "2" representing negative sentiment. The labeling results showed that 12.42% of the reviews were labeled as negative, while the remaining (87.58%) were labeled as positive.

Then, preprocessing data is performed to help the algorithm learning process by transforming unstructured data into structured data. The data preparation phase encompassed case folding, data cleaning, tokenizing, normalization, filtering, and stemming. Case folding aimed to convert all reviews into lowercase. Data cleaning involved the removal of non-alphabetic characters (e.g., emoticons, Chinese characters, etc.), punctuation marks, white spaces, and isolated single letters. Tokenizing was conducted to break down sentence-shaped reviews into words/tokens for easier analysis. Normalization was performed to transform abbreviations, non-standard words, and typos into standard forms, employing the Colloquial Indonesian Lexicon [24] for this process. The filtering process eliminated frequently occurring, irrelevant, non-essential, and meaningless words that have no impact on sentiment analysis, such as stopwords. This filtering process utilized the built-in dictionary of the nltk package. Subsequently, stemming was applied to remove word affixes using the Sastrawi package. An example of pre-processing steps is illustrated in **Table 1**.

Table 1. Example of pre-processing steps

Steps	Results
Initial Review	Kocak gak ad respon, gw tanya g dijawab jawab. Sampe abis terus. Gw apus ulang data juga tetep g ad respon apapun
Case folding	kocak gak ad respon, gw tanya g dijawab jawab. sampe abis terus. gw apus ulang data juga tetep g ad respon apapun
Data cleaning	kocak gak ad respon gw tanya dijawab jawab sampe abis terus gw apus ulang data juga tetep ad respon apapun
Tokenizing	['kocak', 'gak', 'ad', 'respon', 'gw', 'tanya', 'dijawab', 'jawab', 'sampe', 'abis', 'terus', 'gw', 'apus', 'ulang', 'data', 'juga', 'tetep', 'ad', 'respon', 'apapun']
Normalization	['kocak', 'enggak', 'ada', 'respon', 'gue', 'tanya', 'dijawab', 'jawab', 'sampai', 'habis', 'terus', 'gue', 'hapus', 'ulang', 'data', 'juga', 'tetap', 'ada', 'respon', 'apapun']
Filtering	['kocak', 'respon', 'habis', 'hapus', 'ulang', 'data', 'respon', 'apapun']
Stemming	['kocak', 'respon', 'habis', 'hapus', 'ulang', 'data', 'respon', 'apa']

3.2. Modelling

Next, modeling was performed to determine the sentiment type of each review. The data was divided into training and testing sets, with 80% of the data used for training and the remaining for testing. The initial step involved hyperparameter tuning for each classification algorithm. Hyperparameter tuning is the process of identifying parameter values that can yield the best-performing model. Grid Search was conducted on the training data, which involves evaluating each position on the hyperparameter grid to find the best combination of hyperparameter. This process resulted in the specification of the best-performing model as follows.

Table 2. Best Model Specification

Specification	Description
Logistic Regression (LR)	
Best penalty	l2
Best C	10000
Best multi_class	Multinomial
Best max_iter	100
Random Forest (RF)	
Best max_depth	40
Best n_estimator	400
Best min_samples_leaf	1
Best min_samples_Split	2
Support Vector Machine (SVM)	
Best Kernel	linear

Subsequently, modeling was performed using k-fold cross validation to prevent any overlap in the testing data. The data was divided into 10 equally-sized subsets (k=10), with each subset serving as the testing data while the rest acted as the training data. This process allowed for 10 iterations of modeling. In each iteration, the model's performance was evaluated using accuracy and F1-score as the following metrics.

Table 3. Model Evaluation

Fold	LR		SVM		RF	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
1	0.857	0.75	0.816	0.6	0.795	0.583
2	0.999	0.999	0.938	0.769	0.979	0.938
3	0.877	0.75	0.816	0.539	0.836	0.62
4	0.854	0.46	0.854	0.46	0.854	0.46
5	0.937	0.768	0.958	0.822	0.916	0.478
6	0.958	0.904	0.937	0.816	0.937	0.816
7	0.916	0.776	0.937	0.816	0.937	0.768
8	0.937	0.845	0.875	0.59	0.895	0.693
9	0.958	0.888	0.979	0.938	0.958	0.822
10	0.958	0.489	0.958	0.489	0.958	0.489
Mean	0.925	0.763	0.907	0.684	0.907	0.667

In Table 1, it is evident that all three models perform sufficiently well in classifying reviews, as indicated by accuracy values exceeding 0.9 and F1-Scores surpassing 0.6. However, the F1-score value of the LR model is significantly higher than SVM and RF model. This implies that the LR model is better suited for the data and captures the underlying patterns more efficiently compared to SVM and RF. The F1-score is used as a comparative evaluation metric for classification models because it is a better metric for imbalanced data compared to accuracy [25]. Therefore, LR model is chosen as the optimal classification model due to its highest accuracy and F1-Score.

3.3. Wordcloud Analysis

Furthermore, a word cloud analysis was conducted on reviews labeled as negative as this research has a specific objective to identify deficiencies in the application's services that need improvement. Word Cloud is one of the descriptive analysis methods to discover the most frequently appearing words in a collection of words. Word cloud utilizes the Term Document Matrix to visualize a set of words into an appealing display [26]. The word cloud from negative reviews is shown in **Figure 2**.

Figure 2. Wordcloud

The font size in the word cloud indicates the ranking of the frequency of a word obtained from the TF-IDF score. The larger the font of a word, the higher its ranking, indicating that the frequency of

occurrence of that word is higher. Based on **Figure 2**, the most frequently appearing words are "bayar", "sampah", and "limit". The word "bayar" represents reviews from users who have to pay Rp77.000 or $5.14 per week to access the premium version of the application. The word "sampah" translates to "trash" in English, according to the Cambridge Dictionary, meaning something that is worthless and of low quality. Then, the word "limits" signifies reviews from users who are limited to asking questions only three times per day on the application. Other words representing negative reviews include "expensive", "fake", "bug", "full server" and so on.

3.4. Latent Dirichlet Allocation Analysis

Topic modeling was conducted using LDA (Latent Dirichlet Allocation) exclusively on reviews labeled as negative to identify the most frequently discussed shortcomings of the application. The first step in conducting topic modeling is the creation of a dictionary and corpus using the Bag of Words method. The next step is to determine the number of topics. This research selected the number of topics by choosing the value that yielded the highest coherence score.

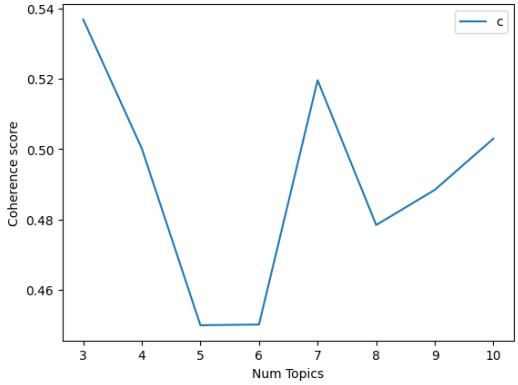

Figure 3. Coherence Score

During the parameter tuning process, the number of topics is determined through 1000 iterations, and coherence scores are calculated across a range of 1 to 10 topics. From Figure 3 it can be observed that the number of topics with the highest coherence score is 3, with a score of 0.536. The resulting topic modelling equations are presented in **Figure 4**.

```
Topic 1: 0.040*"bayar" + 0.026*"mahal" + 0.020*"kali" + 0.020*"gpt"
         + 0.020*"chatgpt"

Topic 2: 0.078*"bayar" + 0.023*"limit" + 0.018*"bagus" +
         0.017*"pakai" + 0.017*"beli"

Topic 3: 0.040*"tidak" + 0.030*"batas" + 0.022*"ai" +0.022*"gratis"
         + 0.022*"jelek"
```

Figure 4. Topic Modelling Equation

Figure 4 displays the topic models. Each model consists of top 5 words with the highest probability of occurrence within that particular topic. In topic 1, the word "bayar" (payment) has the highest probability, with a value of 0.040, followed by the words "mahal" (expensive), "kali" (times), and so forth. The outcomes of the LDA modeling with *gensim* package still have word distributions that overlap across topics, such as the term 'bayar' (pay) appearing in both topic one and topic three. However, since there are relatively few overlapping terms, the model can be interpreted. Independent interpretations of topic names and representative keywords for the formed topics can be extracted in **Table 4**.

Table 4. Representative Keywords

No	Topic	Keywords
1	Paid and Expensive Application	*bayar, mahal, kali, gpt, chatgpt* (pay, expensive, over, gpt, chatgpt)
2	Daily Limit on App Usage	*bayar, limit, bagus, pakai, beli* (pay, limit, good, use, buy)
3	Poor-quaity and Inaccurate Application	*tidak, batas, ai, gratis, jelek* (not, restrict, ai, free, poor)

In **Table 4**, three unique topics can be formed that highlight the most frequently shortcomings of application discussed by users. Topic 1 addresses expensive and paid applications. Topic 2 discusses applications that impose limits on the number of questions per day, and Topic 3 focuses on low-quality applications. This aligns with the words that most frequently appear in the word cloud in **Figure 2**.

Table 5. Representative Reviews

Topik	Contribution	Reviews
1	0.9491	*ChatGPT asli gratis, seseorang menggunakan sumber yang sama dari ChatGPT untuk membuat aplikasi ini dan memeras uang kita.* (Original ChatGPT is free, someone use the same source from ChatGPT to make this app and extort our money)
2	0.930	*apknya bagus cuma, kenapa harus pake limit kan saya ngga ada uang buat beli paket premium* (The app is good, but why should I use the limit? I don't have money to buy the premium package)
3	0.932	*Pertanyaan sensitif di batasi dan lebih cendrung text book. Jawaban sama kaya di google.. jadi nih ai paling ga di rekomendasiin* (Sensitive questions are restricted, and it tends to be more like a textbook. The answers are the same as on Google, so this AI is not recommended)

To obtain specific insights for each topic, the most representative reviews are presented in **Table 5**. It can be seen that review in topic 1 discuss the application being paid, and users suspect it to be a duplicate of the original free ChatGPT application. This review contributes 94.91% to the first topic. The next review in topic 2 focus on the application's limitations for non-premium users. This review contributes 93% to the second topic. Lastly, review in topic 3 address concerns about the application's responses resembling textbook and Google-like content and contributes 93.2% to the third topic.

4. CONCLUSIONS

This research aims to provide sentiment categories for reviews and identify the most frequently discussed topics within all negative-labeled reviews. The best sentiment classification model achieved was logistic regression, with an average accuracy of 0.925 and an F1-score of 0.763. The model classifies 12.42% of the reviews as negative sentiment. Furthermore, the LDA analysis succesfully yielded three dominant topics frequently addressed in negative reviews, such as "Paid and expensive application," "Daily limit on app usage," and "Poor-quality and inaccurate application". Therefore, it is recommended that the application developers reevaluate the pricing, accessibility, and accuracy of the application.

This study encountered several limitations. (1) The reviews used in this research were exclusively in the Indonesian language and were limited to a specific timeframe within the study period. (2) Some words lacked proper normalization, and (3) there are still overlapping terms from the LDA model. The study suggests several considerations for future research: (1) incorporating reviews from a broader temporal range and diverse languages, (2) using a more specific normalization dictionary tailored to application review data, and (3) exploring alternative topic modeling methods.

REFERENCES

[1] K. P. Gunasekaran, "Exploring Sentiment Analysis Techniques in Natural Language Processing: A Comprehensive Review," pp. 1–6, 2023.

[2] Q. Tul *et al.*, "Sentiment Analysis Using Deep Learning Techniques: A Review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, 2017, doi: 10.14569/ijacsa.2017.080657.

[3] A. D'Andrea, F. Ferri, P. Grifoni, and T. Guzzo, "Approaches, Tools and Applications for Sentiment Analysis Implementation," *Int. J. Comput. Appl.*, vol. 125, no. 3, pp. 26–33, 2015, doi: 10.5120/ijca2015905866.

[4] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 1–66, 2017, doi: 10.1007/s10462-016-9475-9.

[5] D. M. Blei and A. Y. Ng, "Latent Dirichlet Allocation," no. January 2001, 2014.

[6] E. S. Negara *et al.*, "Topic Modelling Twitter Data with Latent Dirichlet Allocation Method," 2019.

[7] M. Y. Hendrawan and N. W. K. Projo, "Topic Modelling in Knowledge Management Documents BPS Statistics Indonesia," *Proc. Int. Conf. Data Sci. Off. Stat.*, vol. 2021, no. 1, pp. 119–130, 2022, doi: 10.34123/icdsos.v2021i1.52.

[8] D. Pratmanto, R. Rousyati, F. F. Wati, A. E. Widodo, S. Suleman, and R. Wijianto, "App Review Sentiment Analysis Shopee Application in Google Play Store Using Naive Bayes Algorithm," *J. Phys. Conf. Ser.*, vol. 1641, no. 1, 2020, doi: 10.1088/1742-6596/1641/1/012043.

[9] T. Ali, B. Omar, and K. Soulaimane, "Analyzing tourism reviews using an LDA topic-based sentiment analysis approach," *MethodsX*, vol. 9, p. 101894, 2022, doi: 10.1016/j.mex.2022.101894.

[10] C. Guan, Y. C. Hung, and W. Liu, "Cultural differences in hospitality service evaluations: mining insights of user generated content," *Electron. Mark.*, vol. 32, no. 3, pp. 1061–1081, 2022, doi: 10.1007/s12525-022-00545-z.

[11] L. P. T. Chedia Dhaoui, Cynthia M. Webster, "SOCIAL MEDIA SENTIMENT ANALYSIS: LEXICON VERSUS MACHINE LEARNING," 2017.

[12] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The Impact of Features Extraction on the Sentiment Analysis," *Procedia Comput. Sci.*, vol. 152, pp. 341–348, 2019, doi: 10.1016/j.procs.2019.05.008.

[13] L. E. O. Breiman, "Random Forests," pp. 5–32, 2001.
[14] L. Xin, "A New Text Classifier Based on Random Forests," vol. 107, no. Meita 2016, pp. 290–293, 2017.
[15] C. C. Aggarwal and C. X. Zhai, "A SURVEY OF TEXT CLASSIFICATION ALGORITHMS," *Min. Text Data*, vol. 9781461432, pp. 1–522, 2013, doi: 10.1007/978-1-4614-3223-4.
[16] S. B. Bhonde and J. R. Prasad, "Sentiment Analysis-Methods, Applications & Challenges," *Int. J. Electron. Commun. Comput. Eng.*, vol. 6, no. 6, pp. 2278–4209, 2015.
[17] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621–633, 2013, doi: 10.1016/j.eswa.2012.07.059.
[18] A. Amolik, N. Jivane, M. Bhandari, and M. Venkatesan, "Twitter Sentiment Analysis of Movie Reviews using Machine Learning," vol. 7, no. 6, pp. 2038–2044, 2016.
[19] B. Liu, "Sentiment Analysis: Mining Opinions, Sentiments, and Emotions," 2015, doi: 10.1162/COLI.
[20] J. C. Campbell, A. Hindle, and E. Stroulia, "Latent Dirichlet Allocation : Extracting Topics from Software Engineering Data," pp. 1–21, 2014.
[21] S. Syed and M. Spruit, "Full-Text or abstract? Examining topic coherence scores using latent dirichlet allocation," *Proc. - 2017 Int. Conf. Data Sci. Adv. Anal. DSAA 2017*, vol. 2018-Janua, no. September, pp. 165–174, 2017, doi: 10.1109/DSAA.2017.61.
[22] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring Topic Coherence over many models and many topics," no. July 2012, 2013.
[23] M. D. Hoffman, D. M. Blei, and F. Bach, "Online learning for Latent Dirichlet Allocation," *Adv. Neural Inf. Process. Syst. 23 24th Annu. Conf. Neural Inf. Process. Syst. 2010, NIPS 2010*, pp. 1–9, 2010.
[24] N. A. Salsabila, Y. Ardhito, W. Ali, A. Septiandri, and A. Jamal, "Colloquial Indonesian Lexicon."
[25] C. H. Yutika and S. Al Faraby, "Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes," vol. 5, no. April, pp. 422–430, 2021, doi: 10.30865/mib.v5i2.2845.
[26] G. Pradana, "Penggunaan Fitur Wordcloud dan Document Term Matrix dalam Text Mining," *J. Ilm. Inform.*, vol. 8, no. 1, pp. 38–43, 2020.

BIBLIOGRAPHY OF AUTHORS

Maria A. Hasiholan Siallagan is a bachelor degree student at Department of Statistics, Faculty of Economy Statistics, Politeknik Statistika STIS, Jakarta, Indonesia. Her special interest are statistics models, data science and spatial analysis.

Arie Wahyu Wijayanto is a Lecturer at Department of Computational Statistics Politeknik Statistika STIS, Jakarta, Indonesia. He also currently serves as Head of the Center for Research and Community Service and Assistant Professor at Politeknik Statistika STIS. He completed his Doctor of Engineering in Computer Science from Tokyo University of Technology, Japan. His special interest lies in the field of data science, big data analytics, and geospatial artificial intelligence.