JOURNAL OF DATA SCIENCE | Vol.2024:25
eISSN:2805-5160

Analysis of Sentiment Based on Opinions from the 2019 Presidential Election

Nurul Adha Oktarini Saputri[1*], Misinem[2], Khoirul Zuhri[1]

[1]Faculty of Computer Science, Universitas Bina Darma, Palembang
[2]Faculty of Vocational, Universitas Bina Darma, Palembang, Indonesia

[*]**Email**: nuruladhaos@binadarma.ac.id

Abstract

Twitter is a popular social media platform where the public is free to comment and write about anything. It is common for people to post comments containing harsh words and even hate speech. The 2019 presidential election in Indonesia generated a significant amount of comments, with some users praising the candidates, others criticizing them, and some even resorting to insults. To extract meaningful information from these comments and classify the text, sentiment analysis is essential. In this research, sentiment analysis involves the process of categorizing textual documents into two classes: negative and positive sentiment. The opinion data was collected from the Twitter social network in the form of tweets related to the 2019 presidential election. The dataset used in the study consisted of 3,337 tweets, which were divided into 70% training data and 30% test data. The training data comprised tweets whose sentiment was already known, serving as a foundation for the model to learn and make predictions. The primary objective of this research is to determine whether the tweets, written in Indonesian, express positive or negative sentiments. The Naive Bayes Classifier algorithm was employed to classify the tweet data. This algorithm is well-suited for text classification tasks due to its simplicity and efficiency in handling large datasets. The classification results on the test data demonstrated that the Naive Bayes Classifier algorithm achieved an overall accuracy of 71%. Specifically, the accuracy for negative sentiment classification was 71%, while the accuracy for positive sentiment classification was 70%. These results indicate that the Naive Bayes Classifier is effective in distinguishing between positive and negative sentiments in tweets related to the presidential election.

Keywords

Sentiment Analysis, Classification, Naïve Bayes, Twitter, 2019 Presidential Election

Introduction

As telecommunications technology continues to advance, internet users around the world are experiencing rapid growth in their online activities. This increase is driven by the enhanced ease of conducting everyday tasks using the internet, whether it is communicating via social media,

Submission: 20 June 2024; **Acceptance:** 26 July 2024

searching for up-to-date information, or carrying out buying and selling transactions. According to data from Internet World Status, Indonesia is among the top countries with the highest number of Internet users globally, occupying the fifth position with 143,260,000 users (Akbar et al., 2023). The behaviour of these internet users in Indonesia is predominantly centered on accessing social media platforms such as Facebook, Twitter, Instagram, and YouTube, which accounts for 97.4% of total internet usage.

Social media is defined by Odewole (2017) as an online information technology tool that enables users to communicate easily via the Internet by sharing various forms of content, including text messages, audio, video, and images. In contemporary society, social media has evolved beyond its original purpose of making and finding friends. It now serves as a multifunctional platform for a variety of activities, such as marketing products, sharing news, and providing a space for public opinion and discourse. For example, during the 2019 Presidential Election in Indonesia, social media, especially Twitter, was widely used by the public to express their opinions and sentiments regarding the election results.

The 2019 Presidential Election concluded with a diverse range of opinions being shared by the public, particularly through tweets on Twitter. These tweets can be analyzed to gauge public sentiment about the election, as in the research by Barghuthi and Said (2020). While it is possible to classify these sentiments as positive or negative manually, the vast amount of data makes this process extremely time-consuming and labor-intensive. As a result, researchers are interested in leveraging machine learning methods to classify opinions from such large datasets more efficiently. One effective method for this task is the Naive Bayes Classifier (Al-Aidaroos et al., 2010).

Due to its simplicity and effectiveness, the Naive Bayes Classifier has been widely studied and applied in sentiment analysis. According to Pang, Lee, and Vaithyanathan (2002), the Naive Bayes approach is particularly suitable for text classification tasks, including sentiment analysis, because of its strong probabilistic foundation and ability to handle large datasets. Their study demonstrated that despite its simplicity, the Naive Bayes Classifier can achieve high accuracy in sentiment classification tasks.

Similarly, a study by Pak and Paroubek (2010) utilized the Naive Bayes Classifier for sentiment analysis of Twitter data. They highlighted the classifier's efficiency in handling the vast amounts of unstructured text data typical of social media platforms. Their research confirmed that the Naive Bayes Classifier is not only effective in sentiment analysis but also scalable for real-time applications.

Moreover, recent advancements in machine learning have further enhanced the performance of the Naive Bayes Classifier. Research by Zhang and Wang (2019) explored hybrid models that combine Naive Bayes with other machine learning techniques, such as support vector machines and deep learning, to improve accuracy and robustness in sentiment analysis tasks. These hybrid models leverage the strengths of multiple algorithms, leading to better performance in complex datasets.

In this study, sentiment analysis refers to the technique of classifying textual content into two categories: negative and positive sentiment. The opinion data was gathered from the Twitter social network in the form of tweets on the 2019 presidential election. The major goal of this study is to analyze whether tweets written in Indonesia convey positive or negative attitudes. The Naive Bayes Classifier method was used to categorize tweet data. This technique is ideal for text classification jobs since it is simple and efficient in dealing with huge datasets.

Methodology Research

Research Methods

The researchers used descriptive methodologies, sometimes referred to as descriptive research, in their study. Descriptive research, according to Sugiyono (2015), is a technique used to characterize or portray the item under investigation using data or samples that have been gathered without the goal of drawing widely recognized conclusions or generalizations. This method is useful for giving a thorough grasp of the qualities or present situation of the issue under investigation. Researchers may provide in-depth analysis and a comprehensive picture of the phenomenon they are studying by concentrating on presenting the data exactly as it is.

Data Collection Methods

To collect the necessary data, the researchers used the crawling approach. Crawling is a strategy for autonomously gathering information from the Internet, which Khder (2021) used in his research. This method involves the use of specialized software programs known as crawlers or spiders, which systematically browse the Internet to extract relevant data. For this study, the researchers focused on retrieving data from Twitter.

The researchers' initial stage of the data collection process required them to obtain a Twitter API Key and Secret Key, which are necessary for accessing Twitter's data through its Application Programming Interface (API). To get these keys, the researchers had to register as developers on the Twitter Developer platform and apply for the necessary credentials.

Once equipped with the Twitter API Key, the researchers were able to proceed with the data crawling process. Using the crawling technique, the researchers collected a total of 3,337 tweets related to the topic of interest. The crawling process was automated, with the crawler program scanning web pages and retrieving information based on predefined keywords provided by the researchers.

The data collection was split into two parts: 70% of the data was used as training data, and 30% was used as test data. This division is crucial for the subsequent steps in the research, as the training data helps in building and training the sentiment analysis model. In contrast, the test data is used to evaluate the model's performance.

For the implementation of the crawling process, the researchers used Python, a versatile programming language well-suited for web scraping and data collection tasks. Python's robust libraries and frameworks, such as Tweepy for accessing the Twitter API, enabled efficient and effective data retrieval. The crawler program was programmed with a specific algorithm that allowed it to scan and extract relevant tweet records based on the keywords provided by the researchers.

By employing these methods, the researchers were able to collect a comprehensive dataset that served as the foundation for their sentiment analysis study. This meticulous approach to data collection ensured the reliability and relevance of the data, which is essential for producing accurate and meaningful research outcomes.

Naïve Bayes Classifier

The application of the Naive Bayes Classifier in sentiment analysis involves a series of well-defined steps based on Qi and Shabrina's (2023) works. It begins with data collection, where a large dataset of tweets or other social media posts relevant to the 2019 Presidential Election is gathered. This extensive dataset forms the basis for the analysis and ensures that a wide range of opinions is included.

Once the data is collected, it undergoes data preprocessing to clean and prepare it for analysis. This step involves removing noise, such as irrelevant information and special characters, and normalizing the text for consistent analysis. Normalization might include converting all text to lowercase and eliminating stop words, which are common words that do not contribute to sentiment analysis.

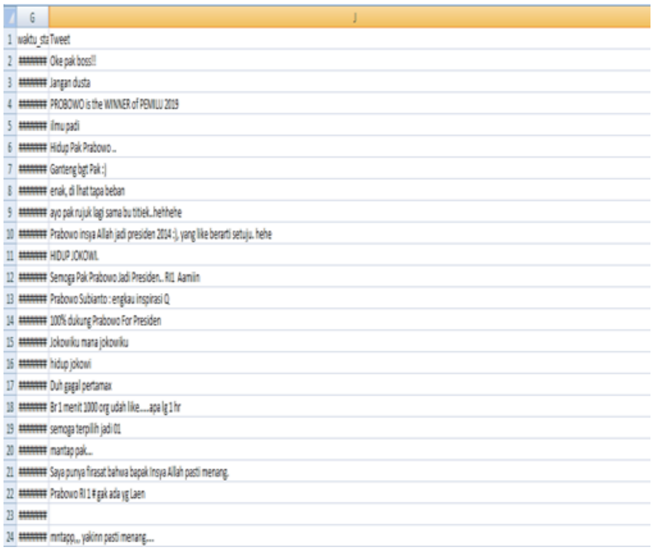
Following stop words, the model training phase begins. Using a labelled dataset, where the sentiment of each post is already known, the Naive Bayes Classifier is trained. During this phase, the classifier calculates the probabilities of different features occurring in positive and negative sentiments. This training helps the model learn the patterns and associations between features and sentiments, allowing it to make accurate predictions.

Finally, the trained model is applied to new, unlabelled data in the classification phase. The classifier uses the features extracted from the latest data to predict the sentiment, classifying the posts as either positive or negative based on the probabilities calculated during the training phase. This automated classification enables the efficient analysis of large volumes of social media data, providing valuable insights into public sentiment.

Results and Discussion

In the context of this research, data crawling was utilized as a fundamental technique to gather a substantial amount of data from Twitter, specifically tweets related to the 2019 Indonesian Presidential Election. The data crawling process aimed to compile a dataset that could provide insights into public sentiment and opinions expressed on social media during this significant political event. Utilizing the Twitter API, researchers were able to systematically extract and collect tweets based on predefined keywords relevant to the election.

The objective of this data-crawling endeavour was to amass a comprehensive and representative sample of tweets that could be analyzed to classify sentiments into positive and negative categories. By employing a Python-based crawler program, the researchers efficiently gathered a total of 3,337 tweet records. This dataset was then divided into training and test sets, with 70% designated for training the sentiment analysis model and 30% reserved for evaluating its performance, as shown in Figure 1.



	G	J
1	waktu_sta	Tweet
2	#######	Oke pak boss!!
3	#######	Jangan dusta
4	#######	PROBOWO is the WINNER of PEMILU 2019
5	#######	ilmu padi
6	#######	Hidup Pak Prabowo ..
7	#######	Ganteng bgt Pak :)
8	#######	enak, di lhat tapa beban
9	#######	ayo pak rujuk lagi sama bu titiek..hehhehe
10	#######	Prabowo insya Allah jadi presiden 2014 :), yang like berarti setuju. hehe
11	#######	HIDUP JOKOWI.
12	#######	Semoga Pak Prabowo Jadi Presiden.. RI1 Aamiin
13	#######	Prabowo Subianto : engkau inspirasi Q
14	#######	100% dukung Prabowo For Presiden
15	#######	Jokowiku mana jokowiku
16	#######	hidup jokowi
17	#######	Duh gagal pertamax
18	#######	Br 1 menit 1000 org udah like.....apa lg 1 hr
19	#######	semoga terpilih jadi 01
20	#######	mantap pak...
21	#######	Saya punya firasat bahwa bapak Insya Allah pasti menang.
22	#######	Prabowo RI 1 # gak ada yg Laen
23	#######	
24	#######	mntapp,,, yakinn pasti menang...

Figure 1. Data crawling result

The results highlight the effectiveness of the data crawling method in capturing a diverse range of public opinions and provide a solid foundation for further sentiment analysis using the Naive Bayes Classifier algorithm, as shown in Table 1.

Table 1. Result in the labeling process

Tweet	Text Cleaning	Label
b'RT@hd_hijau: Jokowi yessss #pilpres#jokowi#prabowo #pemilu#indonesia#prabowosandi#pileg	jokowi yes	0
B'Prabowo insyaAll jadi predisen 2014 :) yang like berarti setuju. Hehe #pilpres2014#pemilu	prabowo insya allah jadi presiden 2014, yang like berarti setuju hehe	0
B'@detik.com: Tolak Prabowo jd presiden masa lalunya yang penuh dengan dosa, sampai2 mantai mertuanya bilang probowo itu penghiatat, jd para pendukung Prabowo gak usah banyak omonglah #jokowi#prabowo#pilpres #jokowipredien	tolak prabowo jd predisen masa lalunyayg penuh dosa sampai mantan mertuanya bilang prabowo itu penghianat jd para pendukung prabowo gak usah banyak omonglah	1
B'Moga Bapak Jokowi Menjadi Pesiden NKRI And Tentunya Tegas, Ramah Dan Bertanggung jawab. :) #pilpres#jokowi	moga bapak jokowi menjadi presiden nkri, tentunya tegas, ramah dan bertanggung jawab	0

Subsequently, the categorization of opinions or points of view from the results of the tweets that were previously crawled will be determined by applying the outcomes of the data crawling technique above to a data labelling procedure. There are two classes involved in this labelling process: the positive class and the negative class. Below is an illustration of the data labelling procedure.

In this case, the positive class label 1 states that the tweet is words that contain elements of hate speech or hate speech, while the negative class label 0 is words that are neutral or do not contain hate speech elements.

Feature Extraction

In the feature extraction process, the first process the system carries out after tokenization is changing the dataset into a vector representation.

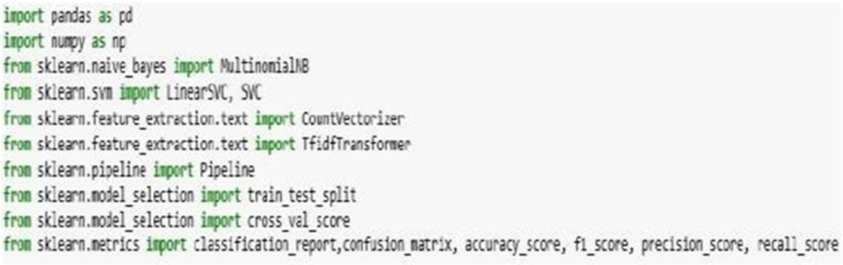
(DTweet1) "Jokowi Yes"
(DTweet2) "Prabowo, God willing, President 2019 Means Agree"
(DTweet3) "Campret Boasts to Kill President Jokowi Alive."

The algorithm returns four standard words from the three phrases above after preprocessing: "President," "Agree," "Live," and "Kill." After completing the stages above, each document is shown as a vector with elements. Words are assigned a value of 1 when they are found in the document and 0 when they are not. Following the conversion of documents into word vectors, the TF-IDF algorithm will be used. This formula produces weighted values in a word vector. The TF-IDF calculating procedure is as follows.

Implementation of Naive Bayes Classification in Python

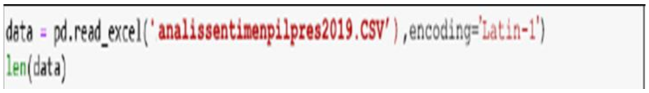
The Naïve Bayes classification and feature extraction processes will eventually be combined into a single class pipeline (vectorizer => transformer => classifier). Apart from Numpy and Pandas libraries for data reading, the scikit-learn libraries used here are Pipeline, CountVectorizer, Naïve Bayes, MultinomialNB, Confusion Matrix, TF-IDFTransformer, and f1 Score. The classification process is carried out with the aid of a Python 3 library called the sci-kit-learn library.

The initial step in the feature extraction and classification process is to install the required libraries. Next, after all the libraries have been installed, we proceed to declare all the libraries that will be used. The program code for the declaration and reading the CSV file are in Figures 2 and 3 below.



```
import pandas as pd
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC, SVC
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report,confusion_matrix, accuracy_score, f1_score, precision_score, recall_score
```

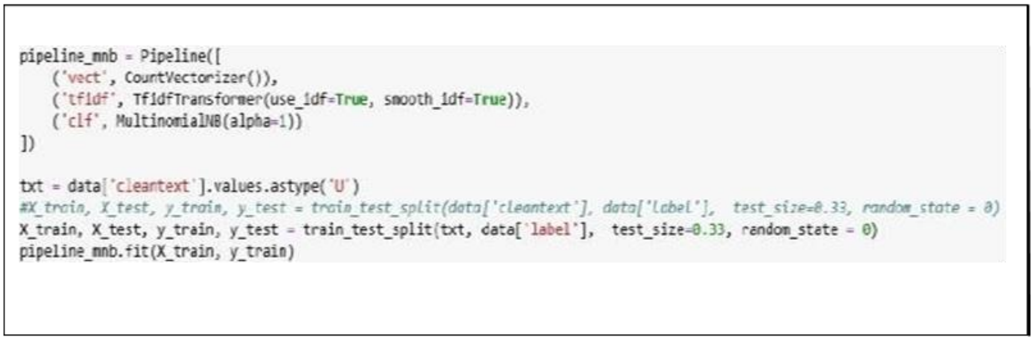
Figure 2. Library declaration function used



```
data = pd.read_excel('analissentimenpilpres2019.CSV'),encoding='Latin-1')
len(data)
```

Figure 3. Function to call data set

The next step involves building a class pipeline, which consists of three stages: first, using the Count Vectorizer library to transform the dataset obtained from crawling Twitter data into a vector representation (converting letters to numbers); second, using the Multinominal library Naïve Bayes for classification. Weighting is done using word vectors found in the TF-IDF Transformer library. Figure 4 below illustrates how the three-class pipeline generation operations are put into practice.



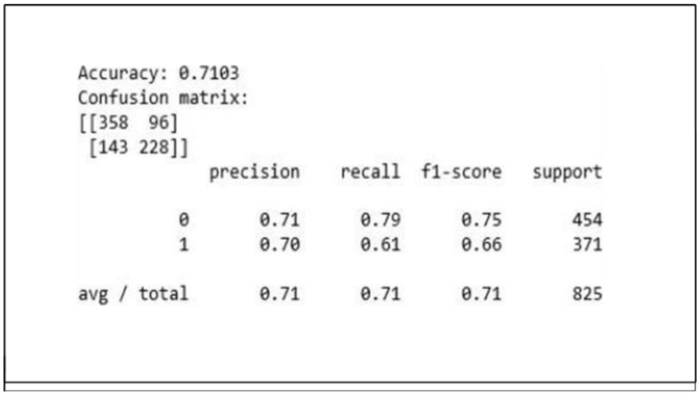
```
pipeline_mnb = Pipeline([
    ('vect', CountVectorizer()),
    ('tfidf', TfidfTransformer(use_idf=True, smooth_idf=True)),
    ('clf', MultinomialNB(alpha=1))
])

txt = data['cleantext'].values.astype('U')
#X_train, X_test, y_train, y_test = train_test_split(data['cleantext'], data['label'],  test_size=0.33, random_state = 0)
X_train, X_test, y_train, y_test = train_test_split(txt, data['label'],  test_size=0.33, random_state = 0)
pipeline_mnb.fit(X_train, y_train)
```

Figure 4. Class Pipeline Implementation Process

TF-IDF, or Term Frequency-Inverse Document Frequency, converts text data into numerical features by assessing each word's significance in relation to the total corpus of documents (after text data has been turned into numerical features using TF-IDF vectorization). Using this approach, a vector is created for every text, with each dimension representing a word's weight determined by its relevance and frequency throughout the corpus.

Next, the Naive Bayes Classifier is trained using the training data. This classifier, grounded in Bayes' theorem, is a probabilistic model that assumes independence between features. During the training phase, the algorithm learns the relationship between the TF-IDF vectorized features and their corresponding labels, which, in this context, represent sentiments.

Once the model is trained, it is then used to predict the sentiment of the test data. The test data, having undergone the same TF-IDF vectorization, is fed into the model, which outputs predictions on the sentiments, classifying them as positive, negative, or neutral. The result is shown in Figure 5.



```
Accuracy: 0.7103
Confusion matrix:
[[358  96]
 [143 228]]
             precision    recall  f1-score   support

          0       0.71      0.79      0.75       454
          1       0.70      0.61      0.66       371

avg / total       0.71      0.71      0.71       825
```

Figure 5. Results of the Model Evaluation Process

We can determine a system's level of success by examining its precision, recall, and F-1 Score across its entire performance. These metrics allow us to assess the system's ability to determine the accuracy or truth of the information a user requests based on the responses it provides. in recalculating data or a 71% accuracy rating. Following the aforementioned steps, the accuracy, recall, and F-1 score for each class may be used to assess how well the classification algorithm performed for that class. The evaluation scale for accuracy, recall, and F-1 Score is 0-1. The higher the value, the better, in the sense that the closer the level 1 value is to 0, the better the system. The results of the entire system model evaluation process are shown in the figure below, as shown in Table 2.

Table 2. Precision, Recall, and F-1 score

Classification	Precision	Recall	F-1 score
Positive	0.71	0.79	0.75
Negative	0.70	0.61	0.66

According to Table 2's model assessment findings, each class's precision and recall values can be determined by looking at the system's processing capabilities when determining the degree of accuracy between the information the user wants to view. For the positive class, this is 71%, and for the negative class, it is 70%. For the positive class, the system processing success rate is 79%; for the negative class, it is 61%. With these numbers, the system performs extremely well in terms of its ability to retrieve both positive and negative information from documents.

Conclusion

This research effectively demonstrates the application of data crawling techniques and the Naive Bayes Classifier algorithm in sentiment analysis, particularly in the context of the 2019 Indonesian Presidential Election. By utilizing Twitter as a data source, the study successfully collected a substantial dataset of 3,337 tweets, providing a rich foundation for sentiment classification.

The data crawling process, facilitated by the Twitter API and Python programming, proved to be a robust method for extracting relevant information from social media. The subsequent steps of data preprocessing and feature extraction ensured that the dataset was clean, normalized, and ready for accurate analysis. The division of the dataset into training and test sets allowed for the effective training and evaluation of the sentiment analysis model.

The Naive Bayes Classifier algorithm, known for its simplicity and efficiency, was employed to classify the tweets' sentiments into positive and negative categories. The classification results demonstrated an overall accuracy of 71%, with 71% accuracy for negative sentiments and 70% for positive sentiments. These findings indicate that the Naive Bayes Classifier is a reliable method for sentiment analysis in this context, providing valuable insights into public opinion as expressed on Twitter.

In conclusion, this research highlights the importance and effectiveness of combining data crawling techniques with machine learning algorithms for sentiment analysis. The ability to automatically classify large volumes of social media data into sentiment categories is invaluable for understanding public opinion. It can be applied to various fields, including politics, marketing, and social sciences. Future work could explore the integration of more advanced machine learning models and the expansion of data sources further to enhance the accuracy and scope of sentiment analysis.

References

Akbar, M., Sonni, A., Suhasman, S., Adawiyah, S., & Herald, Y. (2023). Correlation between Social Media Utilization and the Young Generation's Online Shopping Behavior in Eastern Indonesia. *Studies in Media and Communication*, *11*, 1. https://doi.org/10.11114/smc.v11i7.6223

Al-Aidaroos, K., Abu Bakar, A., & Othman, Z. (2010). *Naïve bayes variants in classification learning*. https://doi.org/10.1109/INFRKM.2010.5466902

Barghuthi, N., & Said, H. (2020). *Sentiment Analysis on Predicting Presidential Election: Twitter Used Case* (pp. 105–117). https://doi.org/10.1007/978-3-030-43364-2_10

Beineke, P., Hastie, T., Manning, C., & Vaithyanathan, S. (2004). Exploring Sentiment Summarization. In Y. Qu, J. Shanahan, & J. Wiebe (eds) Proceeding softhen {AA AI} Spring Symposiumon Explori ng Attitude and Affect in Text: Theories and Applications, AAAI Press.

Dr. Ghayda A. Al-Talib1, Hind S. Hassan, A . (2013). Study on Analysis of SMS Classification Using TF- IDFWeighting

Khder, M. (2021). Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. *International Journal of Advances in Soft Computing and Its Applications*, *13*, 145–168. https://doi.org/10.15849/IJASCA.211128.11

Odewole, S. (2017). Social Media as an Online Information Technology Tool. *Journal of Information Technology and Social Media Studies*, 5(2), 45-58

Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 1320-1326.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 10, 79-86. doi:10.3115/1118693.1118704

Qi, Y., & Shabrina, Z. (2023). Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach. *Social Network Analysis and Mining*, *13*(1), 31. https://doi.org/10.1007/s13278-023-01030-x

Sugiyono (2015). Metode Penelitian Kuantitatif, Kualitatif, dan R&D. Alfabeta.Bandung.

Zhang, L., & Wang, S. (2019). Sentiment Analysis with Deep Learning: A Comparative Study. *Journal of Data Science and Analytics*, 8(1), 15-23. doi:10.1007/s41060-019-00115-5