Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 24, No. 2, March 2025, pp. 273∼284
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v24i2.4791

Sentiment Study of ChatGPT on Twitter Data with Hybrid K-Means and LSTM

Dimas Afryzal Hanan , Ario Yudo Husodo , Regania Pasca Rassy
Universitas Mataram, Mataram, Indonesia

Article Info

Article history:

Received January 08, 2025
Revised February 11, 2025
Accepted March 05, 2025

Keywords:

Chat-Gpt;
K-Means;
Long Short-Term Memory;
Sentiment Analysis;
TF-IDF;
Word2Vec.

ABSTRACT

The rapid development of artificial intelligence technology has significantly impacted various sectors, including communication, education, and business. ChatGPT, a Natural Language Processing based chatbot, has gained widespread attention due to its ability to generate human-like responses. However, concerns regarding ChatGPT's responses' accuracy, ethical implications, and reliability have emerged, necessitating a deeper understanding of public perception. This study analyzes public sentiment towards ChatGPT to understand its perception and implications for AI development and regulation. The research employs a hybrid approach by integrating K-Means clustering and Long Short-Term Memory for sentiment analysis. The dataset, sourced from Kaggle, underwent preprocessing, including text cleaning, tokenization, stopword removal, and lemmatization. Feature extraction was performed using TF-IDF and Word2Vec, while clustering optimization applied the Elbow Method and Silhouette Score to determine the optimal cluster count. The data was then grouped into three clusters based on ChatGPT's functionality, developer, and user engagement. The results indicate that sentiment analysis using LSTM achieved an accuracy of 98% after five training iterations. Most sentiments were negative, particularly concerning technical challenges and transparency, highlighting areas for improvement. While some positive sentiments were present, they were outnumbered by user criticisms. This study concludes that enhancing user trust, transparency, and overall user experience is essential for developing ethical and user-centered AI. The findings offer valuable insights for AI developers and policymakers in creating responsible AI systems. Future research should incorporate multilingual and cross-platform data to understand public sentiment toward AI-driven chatbots comprehensively.

Corresponding Author:

Dimas Afryzal Hanan, +6287712528868,
Departement of Informatics Engineering, Faculty of Engineering,
Universitas Mataram, Mataram, Indonesia,
Email: f1d021085@student.unram.ac.id.

How to Cite:
D. Hanan, A. Husodo, and R. Rassy, "Sentiment Study of ChatGPT on Twitter Data with Hybrid K-Means and LSTM", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol.24, No.2, pp. 273-284, March, 2025.

1. INTRODUCTION

In recent decades, the development of Artificial Intelligence (AI) technology has grown rapidly and significantly impacted various aspects of human life. One of the most prominent innovations is ChatGPT, a Natural Language Processing (NLP)-based chatbot designed to understand and answer questions quickly and accurately [1]. ChatGPT comes with a user-friendly and responsive interface, thus attracting the attention of various groups ranging from students to professionals. Advances in NLP technology make interactions between humans and machines easier and open up enormous opportunities in various industrial sectors, including Indonesia [2]. ChatGPT has become a very useful tool in supporting daily activities, such as text generation, language translation, and data-driven suggestions. In the business world, this technology is utilized to improve the quality of customer service, while in education, ChatGPT acts as an innovative learning partner. In the creative industries, ChatGPT supports content creation and the exploration of new ideas, helping expand the boundaries of human creativity in previously difficult ways.

While this technology offers many advantages, it also presents several challenges and concerns. The widespread use of ChatGPT raises issues concerning the accuracy of information and the risk of misuse of the technology for malicious manipulation of content [1]. In education, the use of ChatGPT also poses its own dilemmas. Excessive reliance on this technology can hinder the development of critical thinking and independent problem-solving skills among students and lecturers [2]. In addition, uncontrolled use of technology can reduce curiosity and initiative to explore other learning resources.

Analyzing public perception is an important step to understanding the impact of ChatGPT in various contexts. A study conducted in Indonesia showed that although most of the public had a positive view of ChatGPT, there were still concerns regarding privacy and the accuracy of the data generated [3]. Another study using the Unified Theory of Acceptance and Use of Technology (UTAUT) framework found that factors such as performance expectations and ease of use play an important role in adopting ChatGPT, especially in educational settings [4]. However, the answers generated by ChatGPT are not always appropriate for more complex educational contexts, which may ultimately affect the effectiveness of the learning process [3].

This research aims to explore public perception of ChatGPT using a sentiment analysis method that combines K-Means Clustering and Long Short-Term Memory (LSTM) algorithms. The K-Means algorithm is employed to group data based on specific patterns, while LSTM excels in understanding the temporal context of text. Combining these two approaches is expected to provide a more comprehensive analysis. In the past, algorithms such as Naïve Bayes and Support Vector Machine (SVM) were commonly used in sentiment analysis but had limitations in capturing complex relationships between words [5, 6]. To address these limitations, recent studies have explored hybrid methods, such as combining Self-Organizing Map (SOM) and LSTM, which have proven effective in analyzing public sentiment toward ChatGPT on Twitter [2]. However, unlike previous research, this study introduces K-Means clustering before classification, which allows data to be grouped into more structured clusters before analysis by LSTM. This innovation is expected to enhance classification accuracy by ensuring the model learns from well-organized data, ultimately improving its ability to interpret sentiment more effectively. As ChatGPT continues to gain popularity, discussions regarding its impact on various fields, such as education and research, have become increasingly widespread. Previous studies using SOM and LSTM for sentiment analysis of ChatGPT-related discussions on Twitter achieved a remarkable accuracy of 95.07% [2], highlighting the effectiveness of deep learning-based approaches. By integrating K-Means clustering with LSTM, this study aims to optimize the feature extraction process, resulting in better generalization and more reliable sentiment classification.

More broadly, technologies such as ChatGPT have not only transformed human communication but also reshaped the way individuals express opinions, collaborate and learn [7]. These advancements have enabled seamless cross-cultural and multilingual interactions, fostering more effective global collaboration. However, despite its widespread adoption, there are still concerns regarding ChatGPT's adaptability, ethical considerations, and alignment with user needs.

Currently, the use of ChatGPT raises questions about its impact on human interaction, the reliability of its responses and its potential for bias. Ideally, this technology should evolve to be more context-aware, ethically responsible, and capable of providing accurate and nonbiased assistance. Understanding how users perceive and interact with ChatGPT is critical in bridging this gap.

This research aims to analyze public perceptions of ChatGPT and identify its strengths and limitations in real-world applications. These findings will serve as a basis for developing more adaptive, ethical, and user-centered chatbot technologies, ensuring responsible integration into various domains.

2. RESEARCH METHOD

This research explores how data preprocessing, feature extraction, and machine learning algorithms affect the process of clustering and sentiment analysis of text data. The dataset derived from Kaggle underwent several preprocessing steps, including text cleaning, tokenization, removal of irrelevant words, and lemmatization. Once these steps were completed, features were extracted using TF-IDF and Word2Vec techniques to create a more precise numerical representation of the text data. These representations then act as input for the clustering stage, where the K-Means algorithm is used to define the ideal number of clusters after an optimization process.

In the final stage, the LSTM model dissects the data patterns and performs sentiment analysis on the previously formed clusters. Figure 1 illustrates the research workflow, outlining each stage from data collection to model implementation. During the pre-processing stage, the focus is on fine-tuning and structuring the raw text data to ensure it is suitable for further analysis. A hybrid feature extraction approach using TF-IDF and Word2Vec improves the accuracy of data representation. Subsequently, the K-Means clustering algorithm clusters data with similar patterns and the LSTM model provides in-depth sentiment analysis for each identified cluster.

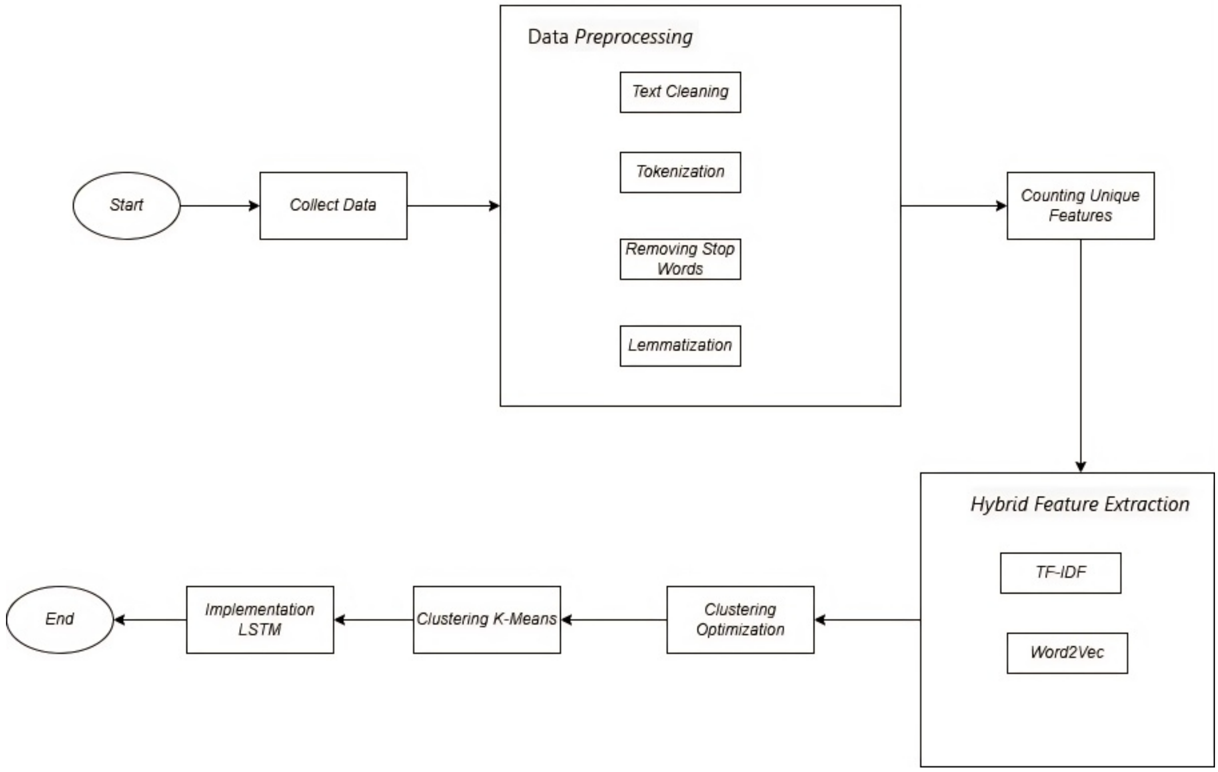

Figure 1. Project flow

2.1. Collect data

The data gathering method in this analysis uses data taken from the Kaggle site, which has 219,294 data with a variety of sentiments, such as positive, negative, and neutral. This data will pass through a series of preprocessing stages to get optimal data results. This research focused on international Twitter users sharing their views on ChatGPT. The objects analyzed were tweets in English to identify global sentiments and opinions on ChatGPT.

2.2. Data preprocessing

Data preprocessing is the first stage in data processing, aiming to transform raw data from various sources into information that is more structured and ready for further analysis [8]. Some steps in data preprocessing include data cleaning, tokenization, stopword removal, and lemmatization. In the text cleaning stage, this process identifies and corrects or deletes invalid or irrelevant data in a data set [9]. Once the cleaning process is complete, the data becomes more ready for the next stage data processing and integration. However, in some cases, the cleaning process can be done iteratively according to the analysis or modeling needs [10]. Next, tokenization breaks down the set of text into smaller units, which are words. Word analysis in text is very important because, through the tokenization process, the meaning of a text can be more easily understood and interpreted [11].

The next step is stopword removal, which removes words that are considered to have no significant meaning in the text. In text processing, stopwords are usually ignored because they only increase the file size without providing meaningful benefits [10]. However, in some cases, stopwords are retained if they have important context and meaning. Finally, lemmatization is the process of returning English words to their base form [12]. Unlike stemming, lemmatization considers the linguistic context of the word so that the results are more accurate and grammatically correct. For example, the word "running" or "ran" will be converted to its base form, which is "run," while maintaining the meaning and structure of the sentence. This technique is very important in text processing as it helps to simplify the data without losing the original meaning.

2.3. Feature extraction

Feature extraction involves identifying the main characteristics of a signal to ensure that the features are unique and show a striking difference compared to others [13]. Feature extraction techniques derive relevant features from pre-processed datasets and may include statistical methods, information theory-based measures, or field-specific algorithms. It is important to choose a variety of feature extraction techniques to explore various representations of data [14]. One of the techniques used is Word2Vec, a word embedding algorithm that maps each word in the text into a vector form. This word representation in vector format is considered effective in storing a word's semantic information. Word2Vec uses an unsupervised learning approach as a word embedding model by applying a neural network consisting of a hidden layer and a fully connected layer to process the data [15].

In addition, the TF-IDF (Term Frequency-Inverse Document Frequency) method is used to calculate the weight of a word (term) in a document, with the aim of indicating its importance. This weight is calculated based on the frequency of occurrence of a word in a document compared to the number of documents in the collection that also contain that word. The following is the equation used in the TF-IDF method, Equation 1.

$$TF - IDF(t_k, d_j) = TF(t_k, d_j) * IDF(t_k) \tag{1}$$

The TF-IDF formula is a way to define how important a word (t_k) is in a document (d_j) compared to all other documents. The first part, $TF(t_k, d_j)$, calculates how often the word appears in a particular document. The more often it occurs, the greater the value. The second part, $IDF(t_k)$, looks at how unique the word is in the entire document collection. If the word is rarely found in other documents, the value is higher as it is considered important and distinctive. When these two values are combined, we get a score that shows how relevant the word is in the context of a particular document. This helps highlight unique and important words, making them very useful for understanding the content of a document or finding keywords.

The first step is to calculate Term Frequency (TF), which is the frequency of occurrence of a word (term) in each document. After that, Inverse Document Frequency (IDF) is calculated, which is the weight of a word based on how often it appears in various documents. The more often the word appears in many documents, the smaller the IDF value. The following is the formula for TF and IDF, Equation 2.

$$TF(t_k, d_j) = f(t_k, d_j) \tag{2}$$

The equation $TF(t_k, d_j) = f(t_k, d_j)$ is a simple way to calculate the frequencies of a word (t_k) in a particular document (d_j). Here, TF or Term Frequency measures the number of times a word (t_k) appears in a document. The $f(t_k, d_j)$ part describes how many times the word appears specifically in the jth document (d_j). In other words, this formula shows how often a word appears in a particular document regardless of context or other documents.This value is fundamental in text analysis, as the more frequently a word appears, the more relevant it is in describing the document's content. However, since it only focuses on individual documents, TF is usually used alongside other components, such as IDF to get more insightful results. Equation 3 is used to quantify how important a word (t_k) is in a document based on how often it appears. IDF (Inverse Document Frequency) measures how unique the word is among multiple documents. Words that appear infrequently will have a greater weight, as they are more meaningful. In contrast, a word frequently appearing in many documents will have a low weight, as it is considered less specific.

$$IDF(t_k) = \frac{1}{df(t)} \tag{3}$$

IDF (Inverse Document Frequency) is a measurement that demonstrates how unique the word is in the context of that document. d_j refers to the J-th document being processed, while df is the number of documents that contain the word. This equation only applies when we only process a single document, so the value of a word is calculated based on how often it appears in that document, without factoring in other documents. In this way, less frequent words in documents will have a higher weight, as they are considered more significant and informed.

2.4. Clustering optimization

Clustering optimization is the process of improving the performance of clustering methods to produce more effective and meaningful data groupings. In clustering, data is grouped based on certain similarities or closeness without initial labels. Optimization is done to ensure that the clusters formed truly reflect the patterns or structures present in the data. In this research, the search for the optimal number of clusters is carried out using two methods, namely the Elbow Method and the Silhouette Score. The Elbow Method used in this research aims to determine the optimal number of clusters in the K-Means algorithm. This method works by calculating the SSE (Sum of Sequence Error) value for each possible number of clusters and visualizing the results using a graph [16]. The formula used in the elbow method is shown in the following Equation 4:

$$SSE = \sum_{i=1}^{k} \sum_{x \epsilon Ci} ||x - \mu_i||^2 \tag{4}$$

In the K-Means algorithm, the value of k represents the number of clusters used, X refers to the amount of data. At the same time, μ_i is the number of clusters at the kth stage in the elbow method algorithm used to determine the optimal k value in K-Means. After determining the optimal number of clusters, the next step is to evaluate the quality of the clustering results using Silhouette Score. Silhouette Score or Silhouette Coefficient is used to assess the quality and accuracy of clustering by measuring the extent to which a data point fits into the cluster to which it belongs. This method combines two main concepts: cohesion, which evaluates how close a point is to its cluster, and separation, which measures the extent to which the point is separated from other clusters formed [16]. Figure 2 illustrates this method visually.

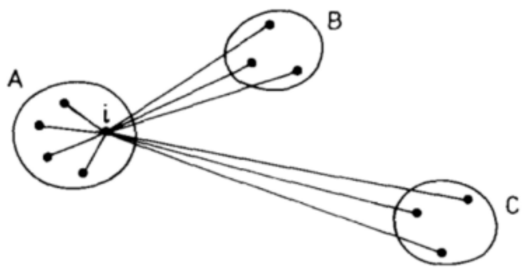

Figure 2. Illustration of the silhouette score determination process

Suppose s(i) represents the dissimilarity for element i in the dataset that belongs to cluster A. If cluster A contains objects other than i, then s(i) is calculated as the average dissimilarity between i and all data in A. As seen in Figure 2, this value represents the average distance of all data in cluster A. In addition, for cluster C, which is different from A, di(i,C) calculates the average difference between i and objects in C. The equation for calculating the Silhouette Score is given as in Figure 5. Where a(i) is the average distance between entity i and other entities in one cluster, while b(i) is the minimum distance of i to entities in other clusters [16]. With this method, the clustering quality can be measured more objectively so that the results of data grouping are more optimal and meaningful.

$$s(i) = \frac{b(i) - \alpha(i)}{max\{\alpha(i), b(i)\}} \tag{5}$$

2.5. K-Means Clustering

K-means is an unsupervised learning algorithm that identifies intrinsic patterns or characteristics in data. The main principle of this algorithm is to divide data into several clusters based on the level of similarity so that data in one cluster has a higher similarity than data in other clusters [17]. The main objective of K-means is to reduce the total sum of squared distances between each data and its cluster center. The K-means algorithm randomly selects cluster centers (centroids) in the feature space. These centers are represented as vectors in the same space as the data. The algorithm uses a distance metric to determine the degree of similarity or difference between data points. The most commonly used metric in K-means is the Euclidean distance, which measures the straight-line distance between two data points in the feature space [17].

2.6. Long Short Term Memory (LSTM)

LSTM is a popular algorithm among NLP researchers due to its ability to understand and process sequential data efficiently. It has been proven effective in handling various tasks in natural language processing, such as text classification and sequence-to-sequence prediction [18]. LSTM was developed to improve the RNN architecture by Hochreiter and Schmidhuber to address the challenge of processing data with long-term dependencies [18]. The algorithm is designed to store and use previously acquired information over a period of time [19]. LSTM has three main types of gates, namely input gate, forget gate, and output gate [20]. The input gate regulates the entry of new information into memory, while the forget gate determines whether old information is still relevant or needs to be deleted. If the value of the forget gate is close to zero, then the information will be ignored, but if it is close to one, the information will still be stored. Meanwhile, the output gate controls how much information will be removed from memory [19]. In addition, this research also uses a softmax layer to determine which information will be forwarded from the forget gate. Softmax helps in the decision-making or classification process of the information generated by the LSTM [18].

3. RESULT AND ANALYSIS

The following research results include data preprocessing, calculating unique features, combining two feature extractions and directly displaying the results of finding the optimal number of clusters, clustering using the K-Means algorithm and performing sentiment analysis using the LSTM (Long Short-Term Memory) model of clustering results using K-Means on each cluster that has been created.

3.1. The results of data preprocessing

Data preprocessing is a crucial stage in text analysis. It aims to improve the quality of the data so that it can be processed more effectively in the next modeling stage. The raw data obtained from Kaggle, which contains ChatGPT user comments via the Twitter platform, needs to go through a series of preprocessing stages to make it more structured and freer of irrelevant elements. The first stage is text cleaning, which involves removing elements with no significant information value in linguistic analysis, such as URLs, non-alphanumeric characters, numbers, and Unicode-based symbols or emojis. This process aims to eliminate noise in the data so that the text obtained is more representative of the actual content of user comments. After cleaning, a case normalization (case folding) process is performed, which converts all letters into lowercase letters to ensure consistency in further processing. The results of this text-cleaning stage are shown in Table 1, which shows the text changes before and after cleaning.

Table 1. Pembagian data untuk Training dan Testing

Cleaning Text Results	
Before cleaning text	After cleaning text
From ChatGPT, neat stuff https://t.co/qjjUF2Z2m0	from chatgpt neat stuff

Next, the cleaned data goes through a tokenization stage, which is the process of breaking the text into smaller word units and compiling a word index dictionary to convert the text into a numerical format. Tokenization is very important in natural language processing (NLP) because it allows each word to be represented as a unique index, which is then used in data-based analysis or modeling. The result of this tokenization process can be seen in Figure 3, where each word in the text is assigned a specific numerical index based on its order in the token dictionary.

```
Word Index (Kamus Tokenizer):
chatgpt: 1
the: 2
to: 3
a: 4
and: 5
is: 6
of: 7
it: 8
i: 9
for: 10
```

Figure 3. Tokenizing results

The next stage is stopword removal, which eliminates common words that do not significantly contribute to understanding the context of a text. Stopwords such as the, and, is, and similar words often occur frequently in the text but carry no specific meaning in the analysis. The model can focus more on words with higher information value by removing stopwords. The results of this process are shown in Table 2, which shows the changes in the text after the stopwords are removed.

Table 2. Pembagian data untuk Training dan Testing

Stopword Removal Results	
Text Data	Stopword Removal
from ChatGPT neat stuff	['chatgpt', 'neat','stuff']

As a final step in preprocessing, lemmatization is performed. This technique aims to simplify words by converting them into their base or root form. It uses linguistic dictionaries to convert words into standardized forms that are more appropriate to their context of use. This method aims to reduce data complexity and improve accuracy in text analysis. The results of this process are shown in Table 3, which compares words before and after applying the lemmatization technique.

Table 3. Pembagian data untuk Training dan Testing

Lemmatization Results	
Text Data	Lemmatization
chatgpt optimizing language models dialogue openai	chatgpt optimizing language model dialogue openai

By applying all these preprocessing steps, the unstructured text data can be converted into a cleaner, more uniform form, ready for use in NLP-based analysis. This systematic application of preprocessing aims to improve the model's accuracy in understanding the language patterns contained in the ChatGPT user comment dataset.

3.2. The results of counting unique features

At this stage, the data's unique features are counted. This step is important because it helps us understand how diverse the data is and recognize elements that only appear once or in limited numbers. Knowing the number of unique features, we can see patterns in the data, such as frequently used and infrequently occurring words or phrases. This information is also useful for determining which features are most influential in further analysis. In addition, understanding unique features can help filter the data by removing words that appear too infrequently or are too common to be meaningful. The results of this calculation can be seen in Figure 4, which shows the number and distribution of unique features in the dataset used.

```
Total Fitur Unik: 154991
```

Figure 4. Results of counting unique features

3.3. The results of hybrid features extraction and cluster optimization

This process aims to convert text into numerical representation using TF-IDF and Word2Vec methods so that the data can be processed further. After the data matrix is merged and standardized, clustering is performed with the KMeans algorithm. The Elbow Method and Silhouette Score determine the optimal number of clusters. The Elbow Method graph shows the ideal number of clusters when the decrease in inertia starts to slow down, which is about 4 or 5 clusters. Meanwhile, the highest Silhouette Score value is found at 3 clusters, indicating the best data division. The results of this process are shown in Figure 5.

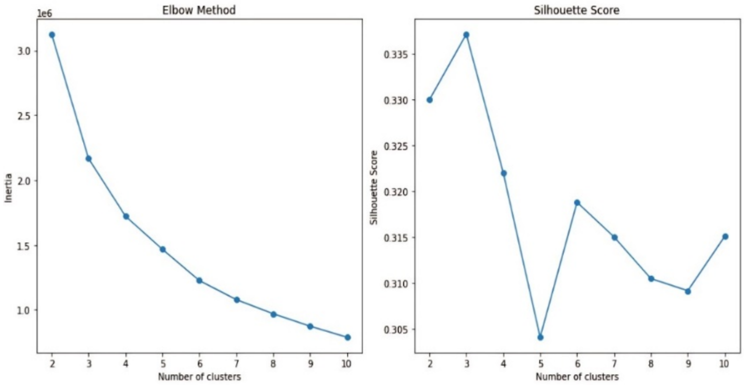

Figure 5. Results of hybrid features extraction and cluster optimization

3.4. The results of clustering with K-Means

After standardization and dimension reduction with PCA, the text data was grouped into three clusters using the KMeans algorithm. Cluster 0 contains 93,789 data, cluster 2 has 75,434 data, and cluster 1 includes 50,071 data. Each cluster was identified based on the most frequently occurring words. Cluster 0 is dominated by terms like ”chatgpt” and ”answer,” Cluster 1 focuses on ”chatbot” and ”openai,” while Cluster 2 includes words like ”write” and ”asked.”

The graph visualization shows the data’s distribution in a two-dimensional space, with clusters shown in different colors. The graph shows clearly separated patterns between clusters, giving an idea of the dominant patterns in the data. The results of this process are shown in Figure 6.

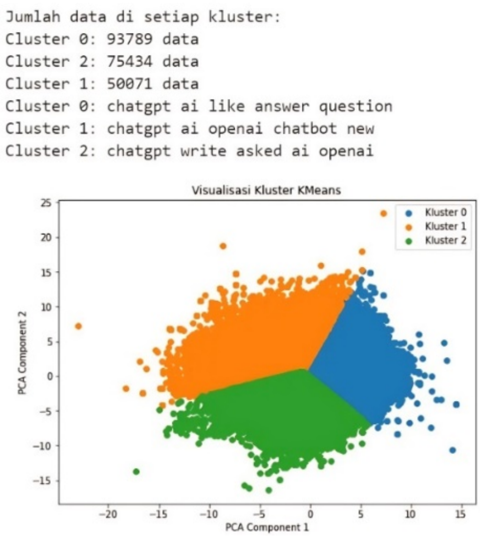

Figure 6. Results of clustering with k-means

3.5. The results of sentiment analysis with LSTM

The results of sentiment analysis using the LSTM model show a good ability to group text data into three clusters with accurate sentiment prediction. The process starts with tokenization and text conversion into numbers, followed by padding to equalize the length of the data. Sentiments were converted into a numeric format for easy classification. After the data was divided for training and testing, an LSTM model was built with an embedding layer to capture the relationship between words, two LSTM layers for sequence patterns, and a dense layer with softmax for sentiment classification. The model was trained for five epochs, with accuracy increasing significantly, from 83.53% in the first epoch to 98.01% in the last epoch. The sentiment distribution per cluster shows different results; Cluster 0 consists of 93,789 data with 29.91% good sentiment, 44.03% bad, and 26.06% neutral. Cluster 1, with 50,071 data, has 19.55% good sentiment, 54.54% bad, and 25.99% neutral, while Cluster 2, containing 75,434 data, shows 24.09% good sentiment, 52.01% bad, and 23.90% neutral. Model accuracy in clusters 0 and 2 reached 94%, while cluster 1 reached 97%. The results of this process can be seen in Figure 7.

```
Epoch 1/5
4386/4386 [==============================] - 4884s 1s/step - loss: 0.4474 - accuracy: 0.8353 - val_loss: 0.3220 - val_accuracy: 0.8924
Epoch 2/5
4386/4386 [==============================] - 3964s 904ms/step - loss: 0.2426 - accuracy: 0.9171 - val_loss: 0.3220 - val_accuracy: 0.8884
Epoch 3/5
4386/4386 [==============================] - 4026s 918ms/step - loss: 0.1484 - accuracy: 0.9508 - val_loss: 0.3431 - val_accuracy: 0.8841
Epoch 4/5
4386/4386 [==============================] - 3595s 820ms/step - loss: 0.0950 - accuracy: 0.9694 - val_loss: 0.4372 - val_accuracy: 0.8782
Epoch 5/5
4386/4386 [==============================] - 3340s 761ms/step - loss: 0.0642 - accuracy: 0.9801 - val_loss: 0.4872 - val_accuracy: 0.8796

Jumlah dan persentase sentimen di setiap kluster:

Cluster 0:
Total data: 93789
Good: 29.91%
Bad: 44.03%
Neutral: 26.06%

Cluster 1:
Total data: 50071
Good: 19.55%
Bad: 54.45%
Neutral: 25.99%

Cluster 2:
Total data: 75434
Good: 24.09%
Bad: 52.01%
Neutral: 23.90%

Akurasi untuk setiap kluster:
2931/2931 [==============================] - 149s 49ms/step
Cluster 0: Akurasi = 0.94
1565/1565 [==============================] - 70s 45ms/step
Cluster 1: Akurasi = 0.97
2358/2358 [==============================] - 172s 71ms/step
Cluster 2: Akurasi = 0.94
```

Figure 7. Results of sentiment analysis with long short-term memory

3.6. Analysis and Discussion

This research combines the K-Means algorithm for data clustering and LSTM for sentiment analysis, which aims to understand better how users perceive ChatGPT based on data from Twitter. The first process undertaken is to clean the text data, such as removing unnecessary words and tidying up the sentence structure, before the data is converted into numerical form using TF-IDF and Word2Vec techniques. This data will then be grouped into three main clusters. The first cluster will highlight topics regarding ”chatgpt” and ”answers,” which is a description of the focus on the main functions of the chatbot. The second cluster is more about ”chatbot” and ”openai” reflecting discussions about the technology and developers. The third cluster highlights user activities, such as ”writing” and ”asking”. The main reason for using clustering before classification is to ensure that the data provided to the LSTM model has a better structure to more accurately identify sentiment patterns within each cluster. In other words, clustering helps to reduce data complexity and improve the quality of sentiment model learning by ensuring that the model does not handle overly variable data in one straightforward classification process. Once the data is classified into three clusters, the LSTM model is used to analyze the sentiment in each cluster.

The LSTM model was used to analyze sentiment based on the cluster results and showed high accuracy, reaching nearly 98% after five training times. The sentiment distribution showed that most users reacted negatively, especially in clusters that discussed developers, which may reflect criticism or concern towards ChatGPT. Meanwhile, positive and neutral sentiments, although present, were not proportional to the number of negative sentiments. This suggests that although ChatGPT offers a wide range of benefits, there is still much room for improvement, especially regarding user expectations and experience.

With this approach, this research can provide better insights into how the public receives AI technologies such as ChatGPT.

However, the limited availability of data on Twitter and the use of the English language are drawbacks that need to be considered. Future research should include other platforms and involve bilingual data for more representative results. The findings in this study are important to guide the development of technologies that are more adaptive, ethical, and responsive to user needs. In addition, the findings in this study can also serve as a basis for policymakers to formulate supportive regulations related to ethical and responsible AI technology development in Indonesia. Thus, this research's cluster results provide academic insights and broad practical applications for society, especially in Indonesia.

4. CONCLUSION

Based on the research that has been done, namely analyzing the sentiment of ChatGPT users on Twitter using a combination approach of the K-Means algorithm for clustering and Long Short-Term Memory (LSTM) for sentiment analysis, where the data obtained through Kaggle is processed through the stages of text cleaning, tokenization, stopword removal, and lemmatization. Feature extraction uses TF-IDF and Word2Vec to produce an optimal numerical representation. Clustering results with K-Means produced three main clusters, each representing a different focus on ChatGPT usage. Sentiment analysis with LSTM showed that the model accuracy reached 98% after five epochs. The sentiment distribution revealed that most users had negative sentiments, mainly related to technical aspects and technology development by OpenAI. This research highlights the need for improvements in user experience and transparency in AI technologies, with recommendations for further studies covering more diverse platforms and languages.

5. ACKNOWLEDGEMENTS

The author would like to sincerely thank all parties who contributed to completing this research. Special appreciation goes to the author's mother for her continuous motivation and support and to the father for his financial assistance. The author is also grateful to the scientific contributors and academic supervisors for their valuable guidance. Additionally, the author extends heartfelt thanks to Baiq Rima, siblings, grandmothers, and all friends whose encouragement and input were invaluable throughout this research. The author would also like to express profound gratitude to the Informatics Engineering Study Program, Faculty of Engineering, Mataram University, for the invaluable knowledge and extensive facilities provided, which greatly contributed to the successful completion of this study.

6. DECLARATIONS

AUTHOR CONTIBUTION

Dimas Afryzal Hanan: coding, testing, writing, and editing. Ario Yudo Husodo: conceptualization, methodology, writing review, and validation, Regania Pasca Rassy: writing review, and validation.

FUNDING STATEMENT

COMPETING INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] D. Transiska, D. Febriawan, and F. N. Hasan, "Analisis Sentimen terhadap Penggunaan ChatGPT Berdasarkan Twitter Menggunakan Algoritma Naïve Bayes," vol. 8, no. 2, pp. 1077–1086, https://doi.org/10.30865/mib.v8i2.7540.

[2] F. M. Sinaga, S. J. Pipin, S. Winardi, K. M. Tarigan, and A. P. Brahmana, "Analyzing Sentiment with Self-Organizing Map and Long Short-Term Memory Algorithms," vol. 23, no. 1, pp. 131–142, https://doi.org/10.30812/matrik.v23i1.3332.

[3] Y. Akbar and T. Sugiharto, "Analisis Sentimen Pengguna Twitter di Indonesia terhadap ChatGPT Menggunakan Algoritma C4.5 dan Naïve Bayes," vol. 5, no. 1, pp. 115–122, https://doi.org/10.55338/saintek.v7i1.

[4] D. Setiawan, D. Arsa, L. Enggrani Fitri, and F. Fadhila Putri Zahardy, "Comparative Analysis of Clustering Approaches in Assessing ChatGPT User Behavior," vol. 10, no. 2, pp. 366–379, https://doi.org/10.31154/cogito.v10i2.661.366-379.

[5] P. Yani and H. Baturohmah, "Analisis sentimen terhadap chatgpt plus menggunakan naïve bayes di twitter," vol. 3, Januari, pp. 107–113, 2023, https://sismatik.nusaputra.ac.id/index.php/sismatik/article/view/210.

[6] I. Kurniasari, A. A. Alfin, and E. Widodo, "Implementasi Long Short-Term Memory (LSTM) dan Word Embedding Model pada Analisis Sentimen Layanan Uang Elektronik Ovo dan Link Aja," vol. 15, no. 2, pp. 237–246, https://doi.org/10.37424/informasi.v15i2.273.

[7] I. Dergaa, K. Chamari, P. Zmijewski, and H. Ben Saad, "From human writing to artificial intelligence generated text: Examining the prospects and potential threats of ChatGPT in academic writing," vol. 40, no. 2, pp. 615–622, https://doi.org/10.5114/biolsport.2023.125623.

[8] A. Nurkholis, D. Alita, and A. Munandar, "Comparison of Kernel Support Vector Machine Multi-Class in PPKM Sentiment Analysis on Twitter," vol. 6, no. 2, pp. 227–233, https://doi.org/10.29207/resti.v6i2.3906.

[9] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," vol. 3, no. 1, pp. 91–99, https://doi.org/10.1016/j.gltp.2022.04.020.

[10] R. R. Salam, M. F. Jamil, Y. Ibrahim, R. Rahmaddeni, S. Soni, and H. Herianto, "Analisis Sentimen Terhadap Bantuan Langsung Tunai (BLT) Bahan Bakar Minyak (BBM) Menggunakan Support Vector Machine: Sentiment Analysis of Cash Direct Assistance Distribution for Fuel Oil Using Support Vector Machine," vol. 3, no. 1, pp. 27–35, https://doi.org/10.57152/malcom.v3i1.590.

[11] Razia Sulthana A., Jaithunbi A. K., H. Harikrishnan, and V. Varadarajan, "Sentiment analysis on movie reviews dataset using support vector machines and ensemble learning," vol. 17, no. 1, pp. 1–23, https://doi.org/10.4018/IJITWE.311428.

[12] F. Muftie, K. M. Yafi, and Q. M. Addina, "Perbandingan Performa Deteksi Cyberbullying dengan Transformer, Deep Learning, dan Machine Learning," vol. 13, no. 1, pp. 75–87, https://doi.org/10.31571/saintek.v13i1.4002.

[13] S. Hidayat, Muhammad Tajuddin, Siti Agrippina Alodia Yusuf, Jihadil Qudsi, and N. N. Jaya, "Wavelet Detail Coefficient as a Novel Wavelet-Mfcc Features in Text-Dependent Speaker Recognition System," vol. 23, no. 1, pp. 68–81, https://doi.org/10.31436/iiumej.v23i1.1760.

[14] A. Sanmorino, S. Suryati, R. Gustriansyah, S. Puspasari, and N. Ariati, "Feature Extraction vs Fine-tuning for Cyber Intrusion Detection Model," vol. 16, no. 2, pp. 302–315, https://doi.org/10.20895/infotel.v16i2.996.

[15] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, dan Fasttext pada Klasifikasi Teks," vol. 14, no. 2, p. 74, https://doi.org/10.33365/jtk.v14i2.732.

[16] M. Guntara and N. Lutfi, "Optimasi Cacah Klaster pada Klasterisasi dengan Algoritma KMeans Menggunakan Silhouette Coeficient dan Elbow Method," vol. 2, no. 1, pp. 43–52, https://doi.org/10.26798/juti.v2i1.944.

[17] J. Xu, J. Gong, L. Wang, and Y. Li, "Integrating k-means clustering and LSTM for enhanced ship heading prediction in oblique stern wave," *Journal of Marine Science and Engineering*, vol. 11, no. 11, p. 2185, 2023, https://doi.org/10.3390/jmse11112185.

[18] M. F. Abdillah and K. Kusnawi, "Comparative Analysis of Long Short-Term Memory Architecture for Text Classification," *ILKOM Jurnal Ilmiah*, vol. 15, no. 3, pp. 455–464, 2023, https://doi.org/10.33096/ilkom.v15i3.1906.455-464.

[19] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2Vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews," vol. 179, pp. 728–735, https://doi.org/10.1016/j.procs.2021.01.061.

[20] A. W. Saputra, A. P. Wibawa, U. Pujianto, A. B. Putra Utama, and A. Nafalski, "LSTM-based Multivariate Time-Series Analysis: A Case of Journal Visitors Forecasting," vol. 14, no. 1, pp. 57–62, https://doi.org/10.33096/ilkom.v14i1.1106.57-62.

[This page intentionally left blank.]