Text Vectorization Techniques for Trending Topic Clustering on Twitter: A Comparative Evaluation of TF-IDF, Doc2Vec, and Sentence-BERT

2023 5th International Conference on Cybernetics and Intelligent System (ICORIS) | 979-8-3503-6948-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICORIS60118.2023.10352228

1st Alvian Daniel Susanto
Computer Science Department, School of Computer Science
Bina Nusantara University
Tangerang, Indonesia
alvian.susanto@binus.ac.id

2nd Steven Andrian Pradita
Computer Science Department, School of Computer Science
Bina Nusantara University
Tangerang, Indonesia
steven.pradita@binus.ac.id

3rd Caroline Stryadhi
Computer Science Department, School of Computer Science
Bina Nusantara University
Tangerang, Indonesia
caroline.stryadhi@binus.ac.id

4th Karli Eka Setiawan
Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
karli.setiawan@binus.ac.id

5th Muhammad Fikri Hasani
Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
muhammad.fikri003@binus.ac.id

***Abstract*— In this digital era, where technology is rapidly advancing, social media has become a primary platform for obtaining and disseminating information. Knowing what is being widely discussed and trending on social media is crucial for important aspects such as politics, economic, social and cultural issues. The objective of this research is to perform clustering on texts or sentences, and within each cluster, identify the most influential keywords that can serve as parameters to determine the topics being discussed in each cluster. Twitter was chosen as the social media platform to be analyzed in this research due to its text-based nature. The clustering method used is DBSCAN, considering that the number of clusters is unknown, and three text embedding techniques to be compared, namely TF-IDF, Doc2Vec, and Sentence-BERT. The performance of clustering with different text embedding techniques were evaluated using the silhouette coefficient. Hyperparameter tuning has been done to find the best-performing hyperparameters. From the best-performing technique, topic finding within the resulting clusters was conducted using Latent Dirichlet allocation (LDA). The results of this research indicated that clustering with DBSCAN and TF-IDF, with the highest silhouette coefficient, namely -0.00001, produced one cluster and 3342 outliers. DBSCAN and Doc2Vec, with the highest silhouette coefficient, namely 0.71590, produced one cluster and one outlier. DBSCAN and Sentence-BERT, with the highest silhouette coefficient, namely -0.02425, produced two clusters and two outliers. Based on the research findings, smaller silhouette scores tend to have a more varied number of clusters. DBSCAN with each tested text embeddings showed that the topic for every cluster, except for the first cluster of DBSCAN that use Sentence-BERT, were COVID-19 related topic. The DBSCAN and Sentence-BERT model, despite having a lower silhouette score, successfully identifies two separate clusters with distinct topics, whereas the other models only identify a single cluster.**

Keywords—clustering, vectorization, TF-IDF, Sentence-BERT, Word2Vec

I. INTRODUCTION

Along with technological developments and increasingly widespread internet penetration, social media has become one of the main sources of information that can influence people's views on certain events or issues. Therefore, it is very important to understand in depth what is currently trending on social media, especially in a wider context such as the current social, political, or economic situation. By analyzing social media data, researchers and organizations can gain a deeper understanding of their target audience, identify emerging trends, and make data-driven decisions.

Because there are a lot of wrong decisions made due to lack of insight about society, providing a better understanding and enhancing comprehension regarding the phenomenon of trending topics on social media, particularly on twitter is the objective of this research. In addition, this study also aims to identify the challenges faced in understanding trending topics and ways to overcome them. In a broader context, this research is expected to contribute to understanding the role of social media in shaping public opinion and behavior.

This research is expected to contribute to the fields of communication and social sciences in general. This research can provide a better understanding of how social media can influence people's views and behaviors. In addition, this research can also provide a practical contribution to organizations or companies that wish to utilize social media as a means of communication with the public.

In this paper, the data obtained from twitter is vectorized by TF-IDF (Term Frequency-Inverse Document Frequency), Doc2Vec, and Sentence-BERT, then the results are clustered by DBSCAN (Density-Based Spatial Clustering of Applications with Noise), refined by LDA (Latent Dirichlet Allocation) and evaluated by Silhouette Score. Then the three vectorizer is compared to get the best vectorizer for topic finding in twitter

The structure of this paper consists of five chapters. After this chapter, the second chapter will discuss the literature review related to the research topic. The third chapter will explain in more detail the methodology used in this research, while the fourth chapter will discuss the research results. The fifth and final chapter will contain conclusions and recommendations for further research.

II. Related Works

A text or a sentence must be converted to a numerical representation for a computer to understand it. The processes of transforming text into a numerical representation are referred to as text vectorization or text embedding. There are many text vectorization techniques, such as TF-IDF, Sentence-BERT, Word2Vec, Doc2Vec, etc. In 2021, a study was carried out that resulted in a system that predicts hate speech using machine learning with SVM (Support Vector Machine) and Bi-LSTM (Bidirectional Long Short-Term Memory) models [1]. The data that were used for the training comes from Twitter data collected by the organizers of PAN 2021, which consists of 200 English-language tweeters and 200 Spanish-language tweeters, with 200 tweeters for each author. The proposed models were SVM with TF-IDF and Bi-LSTM with Sentence-BERT. Research showed that the SVM model achieves 69.5% accuracy, and the Bi-LSTM model reaches 69% [1]. In 2018, a similar study was conducted using logistic regression. The results show that logistic regression performs better with the optimal N-gram range of 1 to 3 for the L2 normalization of TF-IDF. The model was assessed on test data and achieved an accuracy of 95.6%. However, a notable observation was that 4.8% of the offensive tweets were mistakenly labeled as hateful [2].

Word2Vec and Doc2Vec can be used to do keyword extraction for improving short text extraction. From research conducted in 2019, TextRank through Word2Vec and Doc2Vec can extract better than TFIDF and LDA [3]. In research conducted in 2022, using 312 pieces of data from Twitter, the outcomes of utilizing PV-DBOW with SVM, PV-DM with SVM, and logistic regression with Doc2Vec as the method for text vectorization demonstrated superior accuracy and F1-score compared to other models. The top-performing models achieved an accuracy rate of approximately 87% and an F1 score of around 81%. [4].

Kalavani and Thenmozhi showed that the performance of Doc2Vec will be below TF-IDF if the grammar in the dataset is not good or using English for daily conversation [5]. In a study conducted in 2020, researchers compared Doc2Vec and FinBERT on the IPTC Subject Codes prediction task in Finnish. The research results show that Doc2Vec works much better than FinBERT [6]. A study conducted in 2017 demostrated that support vector classifier and logistic regression using TF-IDF or CountVectorizer as the text embedding techniques achieved the highest accuracy and exhibit stability across various circumstances [7]. Research conducted in 2020 regarding event detection on Twitter showed that plain TF-IDF vectors outperformed more recent text vectorization techniques based on neural networks for this task [8].

To identify topics in a document, topic modeling can be used, such as LDA (Latent Dirichlet Allocation) and LSI (Latent Semantic Indexing). A study about topic identification of English and Hindi News Article showed that cosine similarity with LDA outperformed similarity with Doc2Vec and cosine similarity with HDP (Hierarchical Dirichlet Process) [9]. Another research using LDA based topic modeling (dynamic content-specific LDA) can track discussion topics about COVID-19 on social network that were changing rapidly reliably [10].

III. Methodology

A. Dataset

The dataset used was obtained through Kaggle by the name COVID 19 Tweets by GABRIEL PERDA that consist of 178683 rows from 25 july 2020 to 30 august 2020. The dataset contains 13 columns, namely username, user location, user description, user created date, user's number of followers, user's number of friends, user favourites, user verified date, tweet date, tweet text, hashtags, source, and retweet. The column used for clustering is the "text" column which contains user's tweets. The tweets were randomized and reduced to 15000 tweets with random state 42.

B. Doc2Vec

Word2Vec is a widely used model for learning word embeddings, which are vector representations that capture the semantic and syntactic properties of words. By training a neural network on a large corpus of text data, the model can operate. It consists of two main training algorithms: Continuous Bag-of-Words (CBOW) as well as Skip-gram. CBOW utilizes the context words surrounding a target word to predict it, whereas Skip-gram predicts the context words given a specific target word. Both algorithms train a neural network to learn word embeddings by adjusting the weights to maximize the likelihood of correctly predicting the target or context words [3].

The Doc2vec method for learning paragraph vectors takes its inspiration from the Word2vec approach [11]. In Doc2Vec, each paragraph is associated with a distinct vector represented by a column in matrix D, while each word is linked to a unique vector found in matrix W. These word and paragraph vectors are combined to forecast the subsequent word. The paragraph token serves as a form of memory, recollecting the missing word in the tweet, which is why it's referred to as the distributed memory model of paragraph vectors. The motivation behind employing Doc2vec lies in its ability to address the limitations of the bag-of-words model by considering word semantics.

C. TF-IDF

TF-IDF, is a mathematical and statistical measure employed to quantify mathematically the significance of a word in a document within a corpus using a mathematical approach. It involves two things. TF (Term Frequency) is a calculation that determines the frequency of words within a document by dividing the count of a word by the total number of terms in the document. IDF (Inverse Document Frequency) is a measure of the significance of a word in a document. TF-IDF evaluates the importance of words in a document based on the number of occurrences of a word in that document [12]. Mathematically, TF-IDF is expressed as follows:

$$TF - IDF_{ij} = TF_{ij} \cdot log\left(\frac{N}{DF_i + 1}\right) \quad (1)$$

TF_{ij} represents how frequently a word (i) appears in a specific document (j). A higher TF_{ij} indicates greater significance of that word within the document. On the other hand, DF_i, represents the count of documents in which word i is present

at least once. A higher DF_i, value suggests that the word is more commonly found across multiple documents [13].

D. Sentence-BERT

Sentence-BERT (BERT stands for Bidirectional Encoder Representations from Transformers) is a deep learning model designed to generate meaningful representations, or embeddings, for sentences. Unlike traditional word embeddings that capture meaning at the word level, Sentence-BERT focuses on capturing the semantic information of entire sentences. It achieves this by leveraging siamese and triplet network architectures combined with a contrastive loss function.

The siamese network architecture allows for the shared weights to encode two input sentences separately, generating their respective embeddings. This enables the model to capture the semantic information of each sentence. In the triplet network structure, there is an anchor sentence, a positive sentence (which is similar), and a negative sentence (which is dissimilar). The objective is to encourage the model to minimize the distance between embeddings of similar pairs and maximize the distance between embeddings of dissimilar pairs. This helps the model learn to differentiate between semantically related and unrelated sentences. The contrastive loss function further guides the training process by penalizing the model when the embeddings of similar sentences are far apart and when the embeddings of dissimilar sentences are close together [5].

E. DBSCAN

DBSCAN is a clustering algorithm specifically developed for identifying clusters and noise within a dataset [12]. DBSCAN takes two input parameters: epsilon (eps), which defines the radius used for searching neighbouring data points and minPts, which sets the minimum number of points required within this radius to form a cluster [14]. It starts by picking a data point that has at least minPts data points within the eps distance around it. These nearby points are then referred to as “reachable points”. The initial selected point, along with its reachable points, is collectively treated as a cluster. The cluster will expand iteratively by incorporating additional points located within the epsilon distance from the existing reachable points, effectively turning them into new reachable points. This process continues until all data points either have a cluster label assigned to them or have fewer than minPts points within their epsilon distance, causing them to be classified as noise or outliers. The presence of noises or outliers are closely related to the density and the parameters chosen for the algorithm.

In this experiment, Euclidian distance is used as a distance metric for the epsilon. Euclidian Distance is a distance measurement that determines the length of the straight line that is connecting two points of data on Cartesian coordinate. It is computed by taking the square root of the sum of the squared differences between the Cartesian coordinated of those two points of data [15]. It is derived from the Pythagorean theorem. In a two-dimensional plane, given the data points $(x_1,\ x_2)$ dan $(y_1,\ y_2)$, the Euclidean Distance is mathematically expressed as follows:

$$D(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{2}$$

In an n-dimensional space, generally, the Euclidean Distance can be mathematically expressed as:

$$D(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

F. Silhouette Coefficient

The effectiveness of grouping results is assessed using a metric called the silhouette score. It measures how well, in relation to other clusters, each sample in a cluster fit within that cluster. Silhouette Coefficient lies between -1 and 1. Silhouette coefficient of 1 implies the clusters are well separated while Silhouette coefficient of -1 implies the clusters are completely overlapping [16]. When calculating the score, two important factors are considered: the average distance between a sample and all other points within the same cluster denoted as $a(i)$ (intra-cluster distance) and the average distance between a sample and all other points in the cluster that is the closest neighbor denoted as $b(i)$ (nearest-cluster distance). Suppose there are two clusters, cluster N and M, then $a(i)$ is defined as:

$$a(i) = \frac{1}{|N| - 1} \sum_{i \in N,\ j \in N,\ j \neq i} d(i,j) \tag{4}$$

$|N|$ is the number of data point in cluster N and $d(i,j)$ is the distance between i^{th} data in N and j^{th} data in N. Additionally, the score considers the average distance between a sample and all other data points in the cluster that is the nearest neighbor (Nearest-cluster distance) or $b(i)$.

$$b(i) = \frac{1}{|M|} \sum_{i \in M} d(i) \tag{5}$$

$|M|$ is the amount of data point in cluster M and $d(j)$ is the distance between a sample data in N and i^{th} data in M. Then the silhouette coefficient is calculated by

$$s(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)} \tag{6}$$

G. LDA

LDA, or Latent Dirichlet Allocation, is a computational analysis method employed to explore and understand the thematic structure present in a collection of text data [17]. By utilizing statistical measures and an inductive approach, LDA enables researchers to perform exploratory and descriptive analyses on the data. It generates a list of weighted topics for each document. LDA takes number of documents M, number of topics t, and vocabulary matrix β as inputs. It chooses the topic distribution α then assigns each word W in a document D to one of the topics t. For each word W in document D, it calculates $P(topic\ t \mid document\ D)$ for each topic then calculates $P(word\ W \mid topic\ t)$ [18]. The choice of word W

to represent a topic t depends on the distribution of vocabulary words represented by β.

IV. Result and Discussion

A. Text Preprocessing

For the data or text to be consistent, several steps have been taken and applied to clean and prepare the data or text for further computation and analysis. The preprocessing steps included the following:

- Removing URLs: Removal of web addresses or hyperlinks that may be present in the data was aimed, as they don't have any contributions to the semantic meaning of the data.
- Removing mentions: mentions is indicated by the symbol '@' and followed by a username.
- Removing HTML entities: HTML entities, such as '&lt', were removed.
- Removing punctuation: Punctuation marks, including periods, commas, exclamation marks, and question marks, were removed from the text to eliminate any unnecessary symbols that hinder the subsequent analysis and interpretation of the text.
- Removing numbers: Numbers present in the text were removed to avoid numerical information from the text.
- Tokenization: Sentences were tokenized into individual words for next steps of preprocessing
- Lowercasing: All words in the tokenized sentence were converted into lowercase words to maintain and ensure the consistency of the text.
- Removing stop words: Stop words, such as 'the', 'is', and 'of', were removed, as they have no significant semantic meaning.
- Lemmatization: Lemmatization involves transforming a word into its base form. For instance, the word 'saw' is lemmatized to 'see'.
- Rejoining tokens: Rejoin tokens to form sentences.

B. The Experiments

In the experiment, hyperparameter tuning was performed on the parameter epsilon and minPts. The selected values for epsilon were 0.1, 0.25, 0.3, 0.5, 0.75, 0.8, and 1.0. The selected values for minPts were 2, 6, and 10. The choice of epsilon values within the interval [0,1] was made because when epsilon is greater than 1.0, there is a tendency for the number of clusters to converge to 1. The selection of values, specifically 2, 6, and 10, was done because when minPts is greater than 10, there is a tendency for the number of clusters to converge to 1. These values, 2, 6, and 10, were chosen because they can represent the results of other values within the interval [1,10].

a. TF-IDF

From the table I, the Eps means the minimum distance of a text to be considered a cluster and MinPts refers to how many texts is needed to be close to be considered one cluster. the best parameter found for DBSCAN with TF-IDF are 1 Eps and 10 MinPts which resulted in 1 cluster and 3442 outliers with a Silhouette Coefficient of -0,00001. DBSCAN with the hyperparameters of 0.8 for Eps and 2 for MinPts with the silhouette coefficient of -0.26414 produced the most various clusters which has 213 clusters.

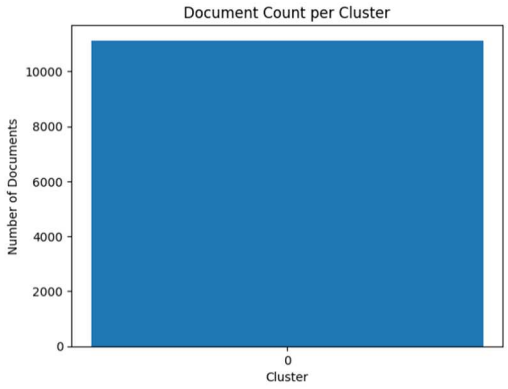

Fig. 1. Bar Chart TF-IDF

TABLE I. Table Clustering Result Summary with TF-IDF

No	Eps	MinPts	Silhouette Coefficient	Number of Clusters	Number of Outliers
1	0.1	2	-0.28987	6	14454
2	0.1	6	-	0	14467
3	0.1	10	-	0	14467
4	0.25	2	-0.28923	11	14444
5	0.25	6	-	0	14467
6	0.25	10	-	0	14467
7	0.3	2	-0.28849	17	14432
8	0.3	6	-	0	14467
9	0.3	10	-	0	14467
10	0.5	2	-0.28320	69	14311
11	0.5	6	-	0	14467
12	0.5	10	-	0	14467
13	0.75	2	-0.26761	182	13929
14	0.75	6	-0.01374	10	14343
15	0.75	10	-0.01052	4	14397
16	0.8	2	-0.26414	213	13803
17	0.8	6	-0.01497	13	14308
18	0.8	10	-0.00944	4	14381
19	1	2	-0.02345	20	3199
20	1	6	0.00005	1	3342
21	1	10	-0.00001	1	3442

b. Doc2Vec

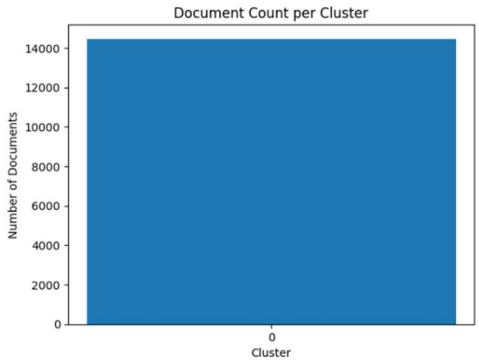

Fig. 2. Bar Chart Doc2Vec

TABLE II. TABLE CLUSTERING RESULT SUMMARY WITH DOC2VEC

No	Eps	MinPts	Silhouette Coefficient	Number of Clusters	Number of Outliers
1	0.1	2	-0.46863	46	14181
2	0.1	6	-0.33460	2	14304
3	0.1	10	-0.29695	1	14348
4	0.25	2	-0.37761	197	5482
5	0.25	6	-0.09625	7	6326
6	0.25	10	0.05892	2	6740
7	0.3	2	-0.26503	127	2901
8	0.3	6	0.01921	4	3453
9	0.3	10	0.11539	2	3706
10	0.5	2	0.32382	4	142
11	0.5	6	0.40353	1	151
12	0.5	10	0.39967	1	165
13	0.75	2	0.59066	1	4
14	0.75	6	0.59066	1	4
15	0.75	10	0.59066	1	4
16	0.8	2	0.60582	1	3
17	0.8	6	0.60582	1	3
18	0.8	10	0.60582	1	3
19	1.0	2	0.71590	1	1
20	1.0	6	0.71590	1	1
21	1.0	10	0.71590	1	1

The best hyperparameters found for Doc2Vec are 1 Eps and 10 MinPts which resulted in 1 cluster and 1 outlier with a Silhouette Coefficient 0f 0.71590. DBSCAN with the hyperparameters of 0.25 for Eps and 2 for MinPts with the silhouette coefficient of -0.37761 produced the most various clusters which has 197 clusters. From the tested hyperparameters, it can be observed that in DBSCAN with Doc2Vec, the resulting clusters do not show significant variations compared to DBSCAN with TF-IDF across the tested hyperparameters. This suggests that the choice of hyperparameters in DBSCAN with Doc2Vec may not have a strong impact on the clustering results, at least in terms of cluster diversity.

c. Sentence-BERT

TABLE III. TABLE CLUSTERING RESULT SUMMARY WITH SENTENCE-BERT

No	Eps	MinPts	Silhouette Coefficient	Number of Clusters	Number of Outliers
1	0.1	2	-	-	14467
2	0.1	6	-	-	14467
3	0.1	10	-	-	14467
4	0.25	2	-	-	14467
5	0.25	6	-	-	14467
6	0.25	10	-	-	14467
7	0.3	2	-	-	14467
8	0.3	6	-	-	14467
9	0.3	10	-	-	14467
10	0.5	2	-	-	14467
11	0.5	6	-	-	14467
12	0.5	10	-	-	14467
13	0.75	2	-	-	14467
14	0.75	6	-	-	14467
15	0.75	10	-	-	14467
16	0.8	2	-	-	14467
17	0.8	6	-	-	14467
18	0.8	10	-	-	14467
19	1.0	2	-0.02425	2	14463
20	1.0	6	-	-	14467
21	1.0	10	-	-	14467

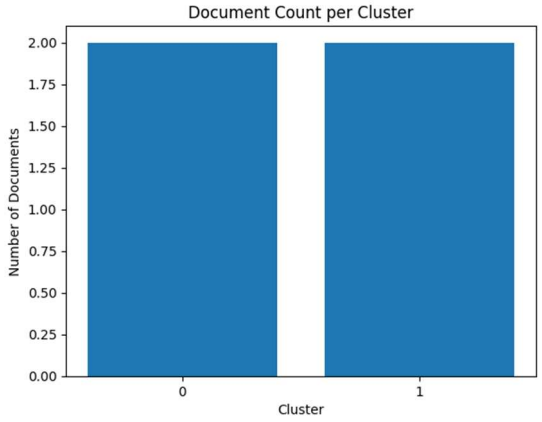

Fig. 3. Bar Chart Sentence-BERT

The result of clustering with DBSCAN using Sentence-BERT indicates that the only hyperparameters that produce clusters are Eps 2 and MinPts 1. With these hyperparameters, the clustering algorithm yields 2 clusters and 14463 outliers and the Silhouette Coefficient of -0.02425. The clustering using DBSCAN and Sentence-BERT didn't work. It might be due to the production of high dimensional data after Sentence-BERT embedding while DBSCAN is weak at processing high dimensional data.

C. Result Analysis

Word clouds were obtained from each best clustering result of DBSCAN with TF-IDF, Doc2Vec, and Sentence-BERT to visualize the most important words in the generated clusters. Additionally, to determine the importance of those words within specific clusters, LDA is used as a method to represent the importance of the word numerically and mathematically.

a. TF-IDF

DBSCAN with TF-IDF produce one cluster. Here is the visualization of prominent words within the cluster using a word cloud.

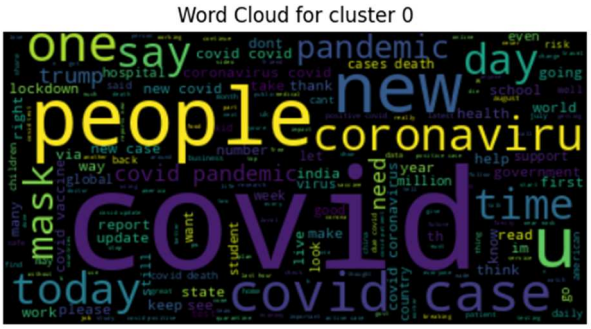

Fig. 4. Wordcloud for cluster with TF-IDF.

The importance of words in this cluster were assessed using LDA. Three words that have the highest importance or significance were obtained from this cluster. The word 'covid' has the weight of 0.05252, 'cases' has the weight of 0.00796, and 'coronavirus' has the weight of 0.00723.

b. Doc2Vec

DBSCAN with Doc2Vec produce one cluster. Here is the visualization of prominent words within the cluster using a word cloud.

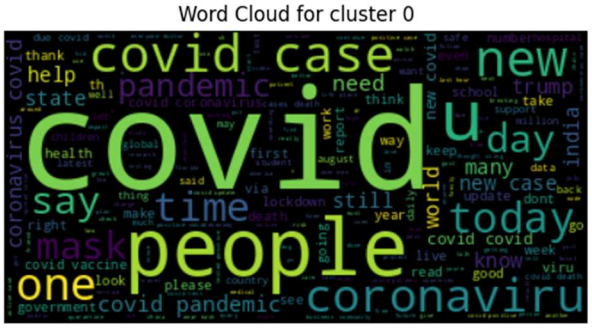

Fig. 5. Wordcloud for cluster with Doc2Vec.

Three words that have the highest importance or significance were obtained from this cluster. The word 'covid' has the weight of 0.05508, 'cases' has the weight of 0.00823, and 'coronavirus' has the weight of 0.00749.

c. Sentence-BERT

DBSCAN with a. Sentence-BERT produce two clusters. Here is the visualization of prominent words within the clusters using a word cloud.

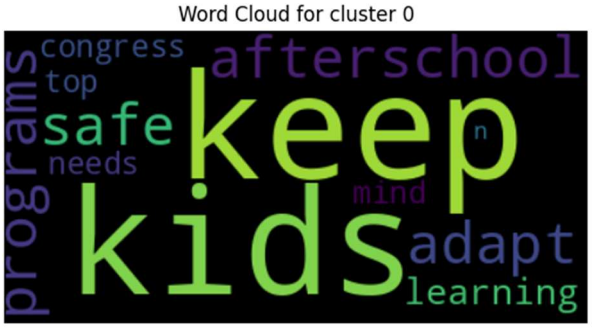

Fig. 6. Wordcloud for first cluster with Sentence-BERT.

Three words that have the highest importance or significance were obtained from this cluster. The word 'keep' has the weight of 0.00021, 'kids' has the weight of 0.00021, and 'mind' has the weight of 0.00012.

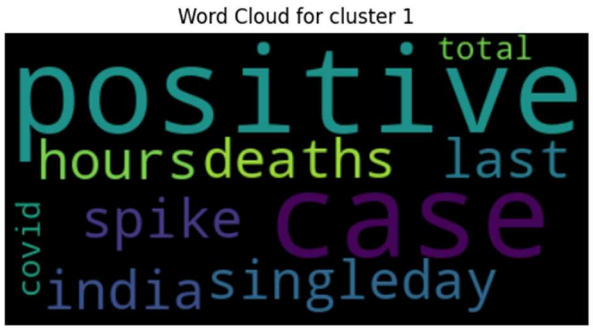

Fig. 7. Wordcloud for second cluster with Sentence-BERT.

Three words that have the highest importance or significance were obtained from this cluster. The word 'positive' has the weight of 0.00021, 'cases' has the weight of 0.00017, and 'deaths' has the weight of 0.00012.

The topic for each cluster produced by each DBSCAN with tested text embeddings can be seen from the significance of a word for each cluster by using LDA. The results showed that the topic for every cluster that was produced by every DBSCAN with tested text embeddings, except for the first cluster in clusters produced by DBSCAN with Sentence-BERT, are COVID-19 related topic.

V. Conclusion

From the conducted experiments, the best clustering results were obtained from each text embedding method evaluated using the Silhouette Coefficient. DBSCAN with TF-IDF yielded the best result with Eps 1 and MinPts 10, achieving a Silhouette Score of -0.00001, 1 cluster, and 3442 outliers. DBSCAN with Doc2Vec yielded the best result with Eps 1 and MinPts 2, achieving a Silhouette Score of 0.71590, 1 cluster, and 1 outlier. Lastly, DBSCAN with Sentence-BERT yielded the best result with Eps 1 and MinPts 2, achieving a Silhouette Score of -0.02425, 2 clusters, and 2 outliers. Therefore, based on the evaluation metric, Silhouette Score, the overall best result was achieved by DBSCAN with Doc2Vec.

For the topic modelling, by using LDA, the topic for the cluster produced by DBSCAN with TF-IDF and Doc2Vec is a COVID-19 related topic. For the clusters produced by DBSCAN with Sentence-BERT, the topic for the first cluster is unlikely to be a COVID-19 related topic, but the second cluster is likely to be a COVID-19 related topic since it has the significant and important word such as 'positive', 'cases', and 'deaths', which is likely to be COVID-19-related words. Those topics modelling results were expected since the datasets contains COVID-19 Tweets. From the results, despite having the lowest silhouette score, the model using DBSCAN and Sentence-BERT was able to capture two distinct clusters with different topics, while the other models only captured one cluster.

The novelty of this research lies in the dataset used and the application of text embedding techniques such as TF-IDF, Doc2Vec, and Sentence-BERT. This study is the first to utilize this dataset for research purposes.

References

[1] Vogel, Inna, and Meghana Meghana. "Profiling Hate Speech Spreaders on Twitter: SVM vs. Bi-LSTM." CLEF (Working Notes). 2021.

[2] Gaydhani, Aditya, et al. "Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach." arXiv preprint arXiv:1809.08651 (2018).

[3] J. Li, G. Huanng, C. Fan, Z. Sun, and H. Zhu, "Key word extraction for short text via word2vec, doc2vec, and textrank," *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, vol. 27, no. 3, pp. 1794–1805, 2019. doi:10.3906/elk-1806-38

[4] Hidayat, Tirta Hema Jaya, et al. "Sentiment analysis of twitter data related to Rinca Island development using Doc2Vec and SVM and logistic regression as classifier." Procedia Computer Science 197 (2022): 660-667.

[5] K. A. and T. D., "Sarcasm identification and detection in conversion context using Bert," Proceedings of the Second Workshop on Figurative Language Processing, 2020. doi:10.18653/v1/2020.figlang-1.10.

[6] Pranjić, Marko, Marko Robnik-Šikonja, and Senja Pollak. "An evaluation of BERT and Doc2Vec model on the IPTC Subject Codes prediction dataset." (2020).

[7] Y. Wang, Z. Zhou, S. Jin, D. Liu, and M. Lu, "Comparisons and selections of features and classifiers for short text classification," *IOP Conference Series: Materials Science and Engineering*, vol. 261, p. 012018, 2017. doi:10.1088/1757-899x/261/1/012018

[8] Mazoyer, Béatrice, et al. "A french corpus for event detection on twitter." Proceedings of the Twelfth Language Resources and Evaluation Conference. 2020.

[9] A. Srivastav and S. Singh, "Proposed model for context topic identification of English and Hindi news article through Lda Approach with NLP technique," Journal of The Institution of Engineers (India): Series B, vol. 103, no. 2, pp. 591–597, 2021. doi:10.1007/s40031-021-00655-w.

[10] M. Zamani et al., "Understanding weekly covid-19 concerns through dynamic content-specific LDA Topic Modeling," Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science, 2020. doi:10.18653/v1/2020.nlpcss-1.21

[11] A. Rane and A. Kumar, "Sentiment Classification System of Twitter Data for US Airline Service Analysis," *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018. doi:10.1109/compsac.2018.00114

[12] Mustakim, M. Z. Fauzi, Mustafa, A. Abdullah, and Rohayati, "Clustering of public opinion on natural disasters in Indonesia using DBSCAN and K-Medoids algorithms," *Journal of Physics: Conference Series*, vol. 1783, no. 1, p. 012016, 2021. doi:10.1088/1742-6596/1783/1/012016

[13] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec," *Information Sciences*, vol. 477, pp. 15–29, 2019. doi:10.1016/j.ins.2018.10.006

[14] Z. Ghaemi and M. Farnaghi, "A Varied Density-based Clustering Approach for Event Detection from Heterogeneous Twitter Data," *ISPRS International Journal of Geo-Information*, vol. 8, no. 2, p. 82, 2019. doi:10.3390/ijgi8020082

[15] S. Nayak, M. Bhat, N. V. Subba Reddy, and B. Ashwath Rao, "Study of distance metrics on K - nearest neighbor algorithm for Star Categorization," *Journal of Physics: Conference Series*, vol. 2161, no. 1, p. 012004, 2022. doi:10.1088/1742-6596/2161/1/012004

[16] G. Ogbuabor and U. F. N, "Clustering algorithm for a healthcare dataset using silhouette score value," *International Journal of Computer Science and Information Technology*, vol. 10, no. 2, pp. 27–37, 2018. doi:10.5121/ijcsit.2018.10203

[17] D. Buenano-Fernandez, M. Gonzalez, D. Gil, and S. Lujan-Mora, "Text mining of open-ended questions in self-assessment of University Teachers: An lda topic modeling approach," *IEEE Access*, vol. 8, pp. 35318–35330, 2020. doi:10.1109/access.2020.2974983

[18] E. S. Negara, D. Triadi, and R. Andryani, "Topic Modelling Twitter Data with Latent Dirichlet Allocation Method," *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, 2019. doi:10.1109/icecos47637.2019.89