Vol. 7, No. 4, December 2025

ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Semantic-Enhanced News Clustering Using TF-IDF and WordNet with K-Means

Mohammad Yusuf Hidayat[1], Muhammad Ainul Yaqin[2], Zainal Abidin[3]

[1]Postgraduate Program, State Islamic University of Maulana Malik Ibrahim, Malang, Indonesia

[2,3]Faculty of Sciences and Engineering, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia

Email: 220605210011@student.uin-malang.ac.id[1], yaqinov@ti.uin-malang.ac.id [2], zainal@ti.uin-malang.ac.id [3]

Received: Oct 29, 2025

Revised: Nov 25, 2025

Accepted: Dec 1, 2025

Published: Dec 16, 2025

Corresponding Author:

Author Name*:

Mohammad Yusuf Hidayat

Email*:

220605210011@student.uin-malang.ac.id

DOI:

10.63158/journalisi.v7i4.1260

Abstract. Text clustering of news articles falls under unsupervised learning, where models operate on unlabeled data unless partially annotated. K-Means Clustering remains one of the most commonly applied algorithms due to its efficiency and simplicity. Likewise, TF-IDF is a widely used approach for generating document feature matrices through statistical term weighting. Although still relevant, TF-IDF lacks the ability to represent contextual meaning, which often prevents semantically related news articles from forming coherent clusters when different syntactic variations are used. This limitation is evidenced by the baseline experiment, in which TF-IDF obtained a silhouette score of 0.011 at the optimal cluster configuration (k = 5). To overcome this limitation, this study introduces semantic enrichment using WordNet to improve similarity representation based on keywords extracted through TF-IDF, evaluated on 1000 documents sampled from 21,495 filtered records. The elbow method was applied to determine the optimal number of clusters. At the optimal k-value of 3, the proposed method achieved a silhouette score of 0.505, significantly outperforming the baseline TF-IDF representation despite utilizing fewer clusters. These results demonstrate that incorporating semantic information can enhance statistical text representations and produce more contextually coherent news clusters. To manage computational task, the model applies a first-POS strategy, where only the first synset derived from POS tagging is considered. While this reduces processing complexity, it may limit the model's ability to fully capture polysemy.

Keywords: News Clustering, TF-IDF, Keyword Extraction, WordNet, Semantic Similarity, K-Means

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

1. INTRODUCTION

News text clustering is a task within the domain of unsupervised learning, aimed at organizing news articles into meaningful groups or clusters [1]. Although news websites typically categorize articles prior to publication, the assigned labels are not always comprehensive or consistently applied [2]. This inconsistency becomes more apparent when readers access content from multiple news providers, each applying different classification schemes. As a result, relying solely on publisher-defined categories may lead to difficulties in locating articles with relevant or similar topics. In the dataset used for this study, several articles were also identified as miscategorized when compared with their actual content, as discussed in Table 9.

This study aims to cluster news articles obtained from multiple sources that apply inconsistent or unfamiliar categorization schemes. Several clustering algorithms—such as K-Means, SOM, and DBSCAN—have been widely applied to similar tasks, each offering distinct characteristics and advantages. Prior research commonly groups clustering approaches into two main types: partitional and hierarchical [3]. The hierarchical approach forms clusters progressively, starting from broader groupings and refining them into more specific sub-clusters [4], often represented using a tree structure. For example, biological classification begins at the kingdom level and is subsequently divided into multiple levels of specificity. In contrast, partitional clustering divides data directly into partitions based on similarity measures. In text clustering, this similarity is typically calculated using statistical features such as TF-IDF weighting [5].

A fundamental aspect of text clustering involves two key components: text representation and the clustering method itself [6]. Among various representation models, TF-IDF remains one of the most commonly used. However, it often leads to high computational cost due to its high-dimensional feature space [7]. Despite this limitation, TF-IDF remains relevant, as it effectively extracts core terms from a document and uses them as distinguishing features for clustering and analysis [8]. For text clustering tasks, one of the most commonly used algorithms is K-Means Clustering[9], primarily due to its simplicity and ease of implementation [10]. The method partitions data by initializing a predefined number of clusters (k) and randomly selecting initial centroids[11]. Several studies have previously applied clustering approaches to news articles or similar textual

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

datasets. For example, Ravi and Kulkarni [12] conducted clustering on Twitter News Channel data from India and achieved an F-measure of 0.98. Saravanakumar [13] applied K-Means to a Multilingual News Stream dataset and reported an F-measure of 0.9476. Aubaidan[10] compared traditional K-Means with K-Means++ using cosine and Jaccard similarity on crime news and Barnama News datasets. The results showed that traditional K-Means with cosine similarity produced an F-measure of 0.819, which was lower than K-Means++ under the same similarity measure. This performance difference is attributed to the centroid initialization strategy: K-Means selects initial centroids randomly, while K-Means++ chooses them probabilistically. Across multiple studies, cosine similarity consistently outperforms Jaccard because it is independent of document length. The findings also indicate that clustering performance tends to decrease when the number of clusters is too large, as detecting similarity becomes more challenging.

TF-IDF remains one of the most widely used approaches in text-based research. However, it is limited in its ability to capture semantic information, particularly when two different terms share similar or related meanings [14]. As a result, news documents with similar contextual meaning may be assigned to different clusters, since the computation relies solely on syntactic similarity. This limitation is also reflected in the baseline TF-IDF experiment in this study, which produced a silhouette score of 0.011 at its optimal clustering configuration. This result indicates that TF-IDF cannot capture contextual meaning, causing semantically similar articles with different surface forms to be misclustered. Several researchers have attempted to incorporate semantic information to improve clustering quality. For example, Wei [15] applied WordNet-based lexical semantics during clustering on the Reuters-21578 dataset and achieved an F-measure of 0.728. Their approach aimed to integrate meaning-based similarity rather than relying solely on statistical frequency. Earlier, Bouras and Tsogkas [3] proposed a hybrid approach combining Bag-of-Words or term frequency with semantic enrichment using WordNet, resulting in the W-KMeans model, which produced a clustering index of 0.84 with relatively low execution time. In contrast to [3] and [15], which enrich semantic information across the entire vocabulary or broader conceptual structures within documents, the proposed approach selectively applies semantic enrichment only to the top keywords identified through TF-IDF and POS-tagging. This selective strategy produces a more compact yet contextually meaningful representation, while improving the computational efficiency of the clustering process.

Based on the identified problem and supporting literature, this study adopts TF-IDF to extract representative keywords from each document and incorporates semantic information from WordNet to account for relationships between terms across documents. The K-Means algorithm is retained as the clustering method due to its demonstrated performance in previous studies and its computational simplicity, making it suitable for large-scale text data.

2. METHODS

2.1. Data Collection

To provide a clearer understanding of the methodological framework applied in this study, a flowchart is presented below. This diagram summarizes the major stages involved in the proposed approach, including preprocessing, feature extraction, clustering, and evaluation. As illustrated in Figure 1, the workflow starts with data collection, followed by a filtering stage to exclude noisy, incomplete, or irrelevant records.

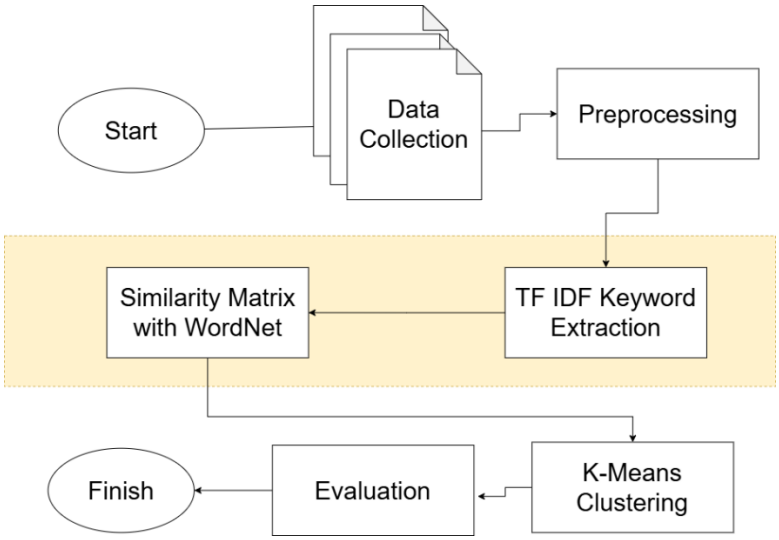


Figure 1. Research Process Flow Design

The dataset used in this study was obtained from the Kaggle platform (Global News Dataset)[16] and is available at: https://www.kaggle.com/datasets/everydaycodings/global-news-dataset. The dataset contains news articles published between September 30, 2023, and October 10, 2023. This dataset has been widely utilized in various natural language

processing tasks, including sentiment analysis, topic modeling, text summarization, and other NLP related research. The total number of records, based on the *article_id* attribute, is 105,375 entries.

Table 1. Sample Records from the Global News Dataset

article_id	source_name	author	title	description	published_at	category	full_content
89541	International Business Times	Paavan MATHEMA	UN Chief Urges World To…	UN Secretary-General Antonio Guterres urged th…	2023-10-30 10:12:35.000000	Nepal	UN Secretary-General Antonio Gute…
89545	The Indian Express	Editorial	Sikkim warning: Hydroelectricity push…	Ecologists caution against the adverse effects…	2023-10-06 01:20:24.000000	Nepal	At least 14 persons lost their live…
89551	Al Jazeera English	Kaushik Raj	Pro-Israel rallies allowed in India…	India, the first non-Arab country to recognise…	2023-10-25 09:58:17.000000	Nepal	India, the first non-Arab country…

The dataset contains news articles from various categories, multiple languages, and includes incomplete records. Therefore, a multi-stage filtration process was applied to ensure that only clean and processable data were retained to prevent computational bias. The first filtering step involved removing entries with missing content, specifically those labeled as "No Description." Next, categories considered irrelevant—such as country names and application-based labels, as shown in Table 1 were excluded. Finally, only

articles written in English were retained. After the filtration process, a total of 21,495 records across 23 categories remained and were used for further experimentation.

Table 2. News Category Distribution After Filtering Process

Category	Count	Category	Count	Category	Count
Stock	3660	Weather	867	Beauty	470
Health	1995	Science	846	Relationships	458
Technology	1927	Artificial Intelligence	644	Movies	394
Real estate	1909	Climate	611	Architecture	300
Finance	1737	Fashion	623	Art	159
Education	1262	Sports	569	Motivation	73
Food	1037	Music	528	Entrepreneurship	51
Jobs	882	Politics	493		

Based on the data presented in Table 1, this study utilizes the title and description fields as the primary textual representation of each news article. Meanwhile, the *article*_id does not indicate the number of documents used but instead serves only as a unique identifier for each record. In this study, only 1,000 samples were selected out of 21,495 filtered records due to the high computational cost associated with semantic enrichment using POS-tagging and WordNet similarity calculations, which require complex pairwise operations.

2.2. Preprocessing Data

During preprocessing, all text was converted to lowercase and paragraph contents were segmented into individual sentences using sentence tokenization. Each sentence was cleaned by removing special characters, symbols, and digits, followed by lemmatization with POS-tagging using the NLTK library to obtain context-appropriate base word forms. This process produced two outputs: a lemmatized sentence and a dictionary of word–POS pairs, which were stored in a list representing each document. The resulting sentences were then stripped of extra whitespace and stopwords before being used for TF-IDF keyword extraction and subsequent semantic similarity computation. This approach was selected to ensure that the preprocessing output supports more accurate contextual similarity measurement. The result of this stage is a set of cleaned terms along with their grammatical categories such as verbs, nouns, adjectives, and others. This

step is intended to ensure that the extracted terms more closely reflect the contextual meaning of the text.

Table 3 presents the results of the preprocessing stage. The clean column represents the cleaned and standardized text, which is a combination of the article title and its description. Meanwhile, the POS maps column contains the processed text along with the corresponding part-of-speech labels after applying POS tagging and lemmatization.

Table 3. Data Before and After Preprocessing

idx	text	clean	clean pos maps
0	Accenture plc (NYSE:ACN) Shares Acquired by Cetera Advisor Networks LLC. Cetera Advisor Networks LLC raised its stake in shares of Accenture plc (NYSE:ACN – Free Report) by 5.3% during the second quarter, according to its most recent 13F filing with the Securities and Exchange Commission. The institutional investor owned 45,780 sh…	accenture plc nyse acn share acquire cetera advisor network llc cetera advisor network llc raise stake share accenture plc nyse acn free report second quarter accord recent f filing security exchange commission institutional investor sh	{'accenture': 'n', 'plc': 'n', 'nyse': 'a', 'acn': 'a', 'share': 'n', 'acquire': 'v', 'cetera': 'n', 'advisor': 'n', 'network': 'n', 'llc': 'r', 'raise': 'v', 'stake': 'n', 'free': 'a', 'report': 'n', 'second': 'a', 'quarter': 'n', 'accord': 'v', 'recent': 'a', 'f': 'n', 'filing': 'n', 'security': 'n', 'exchange': 'n', 'commission': 'n', 'institutional': 'a', 'investor': 'n', 'sh': 'n'}
1	…	…	…

2.3. Keyword Extraction Using TF-IDF

At this stage, TF-IDF is used to identify the most representative terms within each document. The method assigns a numerical weight to each word based on its frequency in a document and its rarity across the entire corpus, allowing distinctive terms to be prioritized[17]. The mathematical formulation of TF-IDF is presented as shown in Equation 1.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

$$TF.IDF_{t,d} = tf_{t,d} \times \log\left(\frac{N}{df_t}\right) \quad (1)$$

Equation 1 defines the components of the TF-IDF calculation. Here, $tf_{t,d}$is the frequency of term $t$in a given document d, $N$is the total number of documents in the corpus, and df_trepresents the number of documents in which the term $t$appears. A total of five keywords were extracted from each document based on the highest TF-IDF scores. The selection of five terms follows the common practice in academic research, where at least five keywords are typically included to represent the core content of a paper. For illustration, ten English-language samples were taken from the dataset used in this study. After the preprocessing stage, the cleaned data were obtained as presented in Table 3, resulting in a total of 210 unique terms.From the extracted terms, the next step is to compute the Document Frequency (DF) and the Inverse Document Frequency (IDF) values. These metrics are used to determine the significance of each term across the document co llection. Table 4 presents an example of the calculated DF and IDF values for a subset of the processed data.

Table 4. IDF Computation

Term	DF	IDF
change	3	= log(10/3) = 0,52288
around, business, climate firm, global, look, russia say, war, Wednesday, would	2	= log(10/2) = 0,69897
Kata lain	1	= log(10/1) = 1

In the official documentation of Scikit-learn, the TF-IDF computation used in the TfidfVectorizer applies smoothing by adding a value of 1 to avoid division by zero during the IDF calculation process. With this modification, the IDF formula becomes:

As an example, the following calculation illustrates the IDF value for the term *"change"*.

IDF change = log ((10+1)/ (7+1)) + 1= 2,0116

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

The TF-IDF value for a specific term in a given document is obtained by multiplying its TF and IDF values. For example, if the term "change" appears once in a particular document, then its TF value equals 1. So,

$$TF.IDF = 1 \times 2{,}0116 = 2{,}0116$$

As a result, the document keywords were generated using the TF-IDF extraction process by selecting the top five terms with the highest TF-IDF scores from each document.

Algorithm 1. TF-IDF–based Keyword Extraction

Input: Preprocessed document set $D = \{d_1, d_2, \dots, d_n\}$

Output: Keyword set $K = \{K(d_1), K(d_2), \dots, K(d_n)\}$

Steps:

1. Construct the TF-IDF matrix M from the document set D.
2. Initialize an empty result container $K = [\,]$.
3. For each document vector v_i in matrix M:
    1. Convert v_i into COO sparse representation.
    2. Extract $(term\_index, tfidf\_score)$ pairs from the vector.
    3. Sort the extracted pairs in descending order based on $tfidf\_score$.
    4. Select the Top-N highest-scoring terms (N = 5 in this study).
    5. Map selected indices to actual feature names (terms).
    6. Store selected terms and their scores in a dictionary $K(d_i)$.
    7. Append $K(d_i)$ to the result container K.
4. Return K as the final keyword extraction result.

Algorithm 1 outlines the procedure for extracting representative keywords from each document using TF-IDF. The TF-IDF matrix is first generated, and each document vector is transformed into a sparse format to efficiently access non-zero term weights. The terms are then sorted in descending order based on TF-IDF scores, and the Top-N highest-scoring terms (N = 5) are selected as keywords for each document. The selected terms are converted into their textual form and stored in a dictionary, producing a concise keyword set used for subsequent semantic enrichment and clustering.

Table 5. Keyword Extraction Across the Datasets

D_1	D_2	D_3	D_n	D_{10}
linkedin 0.411	start 0.454	sponsor 0.397	...	climate 0.393
user 0.205	close 0.454	war 0.338		report 0.308
tailor 0.205	world 0.227	ukrainian 0.199		impact 0.308
summarize 0.205	wework 0.227	ukraine 0.199		adaptation
seeker 0.205	understand	swiss 0.199		0.308
	0.227			accelerate 0.30

Table 5 shows the top five keyword results extracted using the TF-IDF method for several sample documents. Each keyword is accompanied by its TF-IDF weight, representing the level of importance of the term within its respective document. For example, in document D1, the term "linkedin" has the highest TF-IDF score (0.411), indicating that it plays a dominant role in describing the document's content. Tie handling for equal TF-IDF values is implemented by sorting index–score pairs in descending order of TF-IDF weights. When multiple terms share the same score, they are further sorted by their feature index in ascending order. This approach ensures deterministic and consistent keyword selection, as terms with identical weights are prioritized based on their index position in the feature space.

2.4. Similarity Matrix Using WordNet

After each document has its keywords extracted, the process continues with similarity computation. The initial stage of similarity measurement is string-based similarity [18]; if the strings are identical, the similarity score is assigned the maximum value. The next step involves semantic similarity measurement based on the WordNet ontology. The WordNet dataset was downloaded from the NLTK website. This similarity calculation employs the Wu-Palmer method , which considers similarity based on the depth of the hierarchy of the synsets of two words within the WordNet taxonomy [19].

$$\text{sim}_{\text{wup}}(s1,s2)=2\times\frac{\text{depth}(\text{lcs}(s1,\ s2))}{(\text{depth}(s1)+\text{depth}(s2))} \tag{2}$$

Equation (2) involves synsets from WordNet, where $s_1$and $s_2$correspond to individual synsets, and $depth(lcs)$indicates the depth of their Least Common Subsumer. LCS is the

most general concept shared by the two words whose similarity is being measured. For example, the Wu-Palmer similarity calculation can be illustrated using the words article and country from the sample dataset.

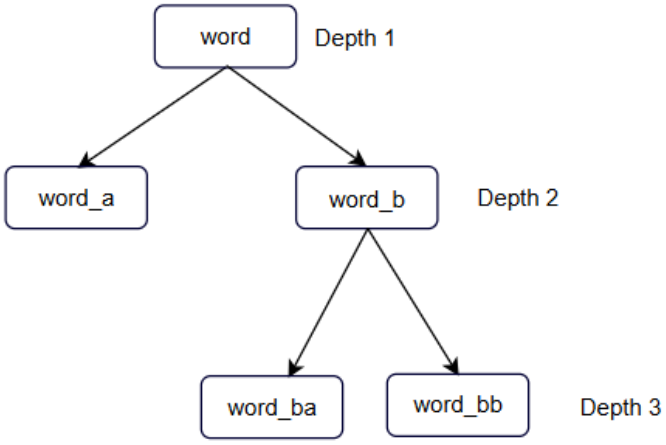


Figure 2. Illustration of the LCS Concept in Wu-Palmer Similarity Calculation

WS4J indicates that the LCS of these two words is at depth 3, while the depth of the word article is 8 and that of country is 9. Therefore,

$$Sim_{wup}\,(article, country) = (2 \times 3)/(8+9) = 0.3529$$

This similarity calculation is performed between each term obtained and the top five keywords in each document, selecting the first term from the POS tagging process according to its word type. The mean and standard deviation are then computed. These values will later serve as the feature extraction results for each document. In this study, out-of-vocabulary (OOV) terms that are not present in WordNet are not removed during preprocessing. Instead, they are retained, and their similarity values are computed using a deterministic rule: an OOV term receives a similarity value of 1 if it is identical to the corresponding keyword, and 0 otherwise. If a feature column results in zero-constant values across all documents, the column is subsequently removed from the similarity matrix, as it provides no discriminative information. This procedure also contributes to automatic dimensionality reduction.

2.5. Clustering Process

The result of the previous process is a similarity matrix, obtained by calculating the highest similarity value between the terms and the keywords of each document. This matrix is then used as input for the clustering process. The grouping process employs the K-Means Clustering method. This model is widely used due to its simplicity and ease of implementation. Despite having several limitations, some studies, such as [14], have shown that it can achieve good performance. The roots of the K-means algorithm can be found in Lloyd's 1957 Bell Labs work, in which he proposed the technique for use in pulse-code modulation [20] The basic steps of the K-Means Clustering process are as follows:

1) Determining the number of k-clusters, for example, in this stage, the number of clusters is set to k = 2
2) Initializing the initial centroids, typically, random numbers are used. The randomness in initialization can lead to variations in clustering results each time the process is run. Therefore, in the K-Means implementation from sklearn, *random*_state = 42 and n_*init* = 10 are set. Multiple trials on the sample dataset produced consistent clustering results.
3) Calculating the distance between centroids and each data point. Euclidean Distance is the most commonly used measure for this calculation.

$$d(i,j)=\sqrt{\sum_{p=1}^{n}(x_{ip}-x_{jp})^2} \quad (3)$$

Equation (3) defines the Euclidean distance, which measures the straight-line distance between two points $i$and $j$in an n-dimensional feature space. Here, $x_{ip}$and x_{jp}denote the values of the p-th feature for points $i$and j, respectively. For example, there are two datasets illustrated as follows.

A = [2, 4, 5, 6, 7, 8, 3, 5, 9, 1]

B = [3, 5, 2, 6, 8, 7, 4, 5, 10, 2]

From the example, we can initialize (n = 10), where A represents the first dataset, and B represents the first cluster centroid.

$$(A,B) = \sqrt{\sum_{p=1}^{10}(A_p-B_p)^2}$$

$$= \sqrt{(2-3)^2+(4-5)^2+(5-2)^2+(6-6)^2+(7-8)^2+\ldots+(1-2)^2}$$

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

$= \sqrt{1+1+9+0+1+1+1+0+1+1}$

$= \sqrt{16}$

$= 4$

From the calculation above, the distance between data point A and centroid B is 4.

4) Grouping each input data point based on the shortest distance to each centroid. For example, if there are initially two cluster centroids, with B as the first cluster centroid and C as the second cluster centroid, the distances are calculated as follows:

$d(A, B) = 4$

$d(A, C) = 5$

The first clustering result is $\min(d(A, B), d(A, C)) = 4$. Therefore, dataset A is assigned to the cluster of centroid B, or cluster 1.

5) Update new centroid

$$C_{ij} = \frac{1}{N_i} \sum_{k=0}^{N_i} x_{kj} \quad (4)$$

In Equation (4), the cluster centroid C_{ij} is updated as the mean of all points assigned to the i-th cluster. Here, N_i denotes the total number of points in the cluster, and x_{kj} refers to the j-th feature of the k-th point within the cluster. For example, if the members of cluster 1 are A and C, then for each data point in datasets A and C, compute the mean for each feature or dimension coordinate:

New Centroid = [(2+3)/2, (4+5)/2, (5+2)/2, ..., (1+2)/2]

New Centroid = [2.5, 4.5, 3.5, 6, 7.5, 7.5 3.5, 5, 9.5, 1.5]

6) Repeat the iteration from steps 3–5 until convergence. Convergence is usually achieved when the cluster assignments for the dataset no longer change, or when changes are not significant. The determination of the number of clusters to achieve optimal grouping can be performed using the elbow method[21]. This method identifies the appropriate number of (k)-clusters by observing the point on the graph where the curve begins to level off.

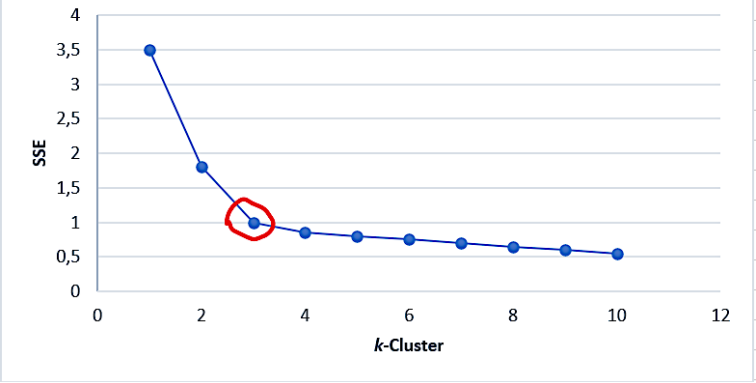


Figure 3. Illustration Result Elbow Method

This method typically uses the calculation of the Sum of Squared Errors (SSE), which is compared against progressively increasing values of k.

The SSE formula is as shown in Equation 5.

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij}^{p} \, dist(o_i, c_j)^2 \quad (5)$$

Equation (5) defines the Sum of Squared Errors (SSE), which measures the overall dispersion of data points relative to their cluster centroids. In this equation, o_i is the i-th data point, c_j is the centroid of the j-th cluster, and w_{ij}^{p} indicates the weighted membership of the point to the cluster. The double summation runs over all points and clusters to quantify the total clustering error. For example, consider SSE calculation with the number of clusters $n = 2$. For cluster 1 with centroid $[1,2]$ and three members A = [2, 4], B = [3, 2], and C = [1, 6]:

$$dist\,(A, \text{cluster 1}) = (2-1)^2 + (4-2)^2 = 5$$
$$dist\,(B, \text{cluster 1}) = (3-1)^2 + (2-2)^2 = 4$$
$$dist\,(C, \text{cluster 1}) = (1-1)^2 + (6-2)^2 = 16$$
$$\text{SSE Cluster 1} = 5 + 4 + 16 = 25$$

Similarly, SSE is calculated for Cluster 2, Cluster 3, and so on, resulting in the following:

$$\text{SSE Cluster 2} = 12$$
$$\text{SSE Cluster 3} = 10$$

The total SSE is obtained by summing the SSEs of all clusters:

$$\text{SSE} = \text{SSE Cluster 1} + \text{SSE Cluster 2} + \text{SSE Cluster 3} = 25 + 12 + 10 = 47$$

SSE is calculated every time the clustering process is executed. The smaller the SSE, the closer the data points are to their cluster centroids. As the number of clusters $k$increases, the SSE generally decreases. When SSE changes become insignificant after a certain number of clusters, the elbow method is applied to determine the optimal cluster number as shown in Figure 3. Once the appropriate number of clusters is found, the clustering result corresponding to this number is selected. The following is an example of clustering results using the dataset in Table 2 with $n = 2$ clusters, determined according to the elbow method using SSE calculation.

Table 6. Cluster Result of n=2

	Text	Cluster
0	LinkedIn hits 1 billion members, adds AI featu…	0
1	WeWork to start closing some offices around th…	0
2	Ukraine brands Swiss food giant 'sponsor of wa…	0
3	Bill Ackman says it's 'pathetic' that law firm…	0
4	Netanyahu is focused on his own political 'sur…	0
5	Carlsberg would rather lose its Russian busine…	1
6	What has changed for voters since the 2020 ele…	0
7	How a climate model can illustrate and explain…	0
8	Scholars reveal improved human greenspace expo…	0
9	Report: As climate impacts accelerate, finance…	1

2.6. Method Evaluation

The evaluation model used in this study is the Silhouette Coefficient, which measures how close an object is to its own cluster compared to other neighboring clusters. It was first introduced by Rousseeuw in 1987 [22]. The coefficient ranges from -1 to 1, with higher values indicating better clustering results [23]. The Silhouette Coefficient is defined as follows:

$$s(i) = \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad (6)$$

Equation (6) defines the silhouette coefficient, $s(i)$, which measures how well object $i$fits within its assigned cluster. Here, $a_i$represents the average dissimilarity of $i$with all other objects in the same cluster A, while $b_i$is the smallest average dissimilarity between $i$and the objects in any other cluster $C(A \neq C)$. The coefficient ranges from –1 to 1, with higher values indicating better clustering.

3. RESULTS AND DISCUSSION

3.1. TF-IDF for Keyword Extraction

During the preprocessing stage, some data were found to be unsuitable for further processing, so data filtering was performed. In this stage, data cleaning was carried out twice, taking into account the condition of the dataset. From the available data, the researcher selected 1,000 records for experimentation across 20 categories to facilitate the selection and determination of the percentage of data to be used.

Table 7. TF-IDF Keyword Extraction Results

Text	Clean Text	Keywords
WeWork's inevitable retreat is here. WeWork has started to close some of its coworking spaces as it reportedly prepares to file for bankruptcy.	wework inevitable retreat wework start close coworking space reportedly prepare file bankruptcy	wework 0.575 coworking 0.335 inevitable 0.329 retreat 0.299 bankruptcy 0.275

POS tagging was performed prior to lemmatization to obtain the base form of words as expected in context. As shown in Table 7, five keywords were extracted from each document. The TF-IDF process was used to identify the top five terms in each news article, which serve as the representative features of the text.

3.2. Enhancement with WordNet Semantic Information

The feature extraction calculation using WordNet involves taking the highest similarity value between all terms and the keywords of each document obtained from the previous

process. The similarity score is selected from the first position of each word type and used to construct the similarity matrix.

Table 8. Construction of the Similarity Matrix with Semantic Information

indeks dok.	aai	abc	abduct	ability	able	...	zillow
0	00	-0.0618	-0.0494	-0.0988	-0.0618	...	00
1	00	-0.0327	-0.0246	-0.0490	-0.0295	...	00
2	00	-0.0395	-0.0274	-0.0656	-0.0345	...	00
3	00	-0.0547	-0.0215	-0.0473	-0.0253	...	00
4	00	0.0074	0.0174	0.0130	0.0261	...	00

As shown in Table 8, the similarity matrix was constructed with a total of 6,509 terms, compared against the top five keywords from each document. The values were derived from the mean minus the standard deviation. Some columns in the similarity matrix contained zero-constant features due to the calculations. These columns were removed [24] because they did not carry meaningful information, reducing the feature dimension to 5,490. After generating the document similarity matrix, a Min–Max Scaling normalization procedure was applied to transform the values into the range [0,1][25]. This step was necessary because the similarity matrix contained negative values, which may cause bias in distance calculations and degrade clustering performance. Figure 4 shows the heatmap of the similarity matrix, demonstrating the variation in similarity values across documents. By scaling all features to a uniform range, the contribution of each dimension becomes balanced, and the K-Means algorithm can compute Euclidean distances more effectively, reducing distortion caused by differing value scales.

Figure 5 illustrates the distribution comparison of similarity scores before and after semantic enrichment. The baseline exhibits a highly concentrated distribution near zero, indicating a limited ability to capture contextual similarity between documents, as most document pairs are assigned very low similarity values. In contrast, the proposed semantic-enhanced representation produces a broader and more dispersed distribution spanning approximately −0.1 to 0.25, reflecting increased variability and greater sensitivity in distinguishing semantic relationships across documents.

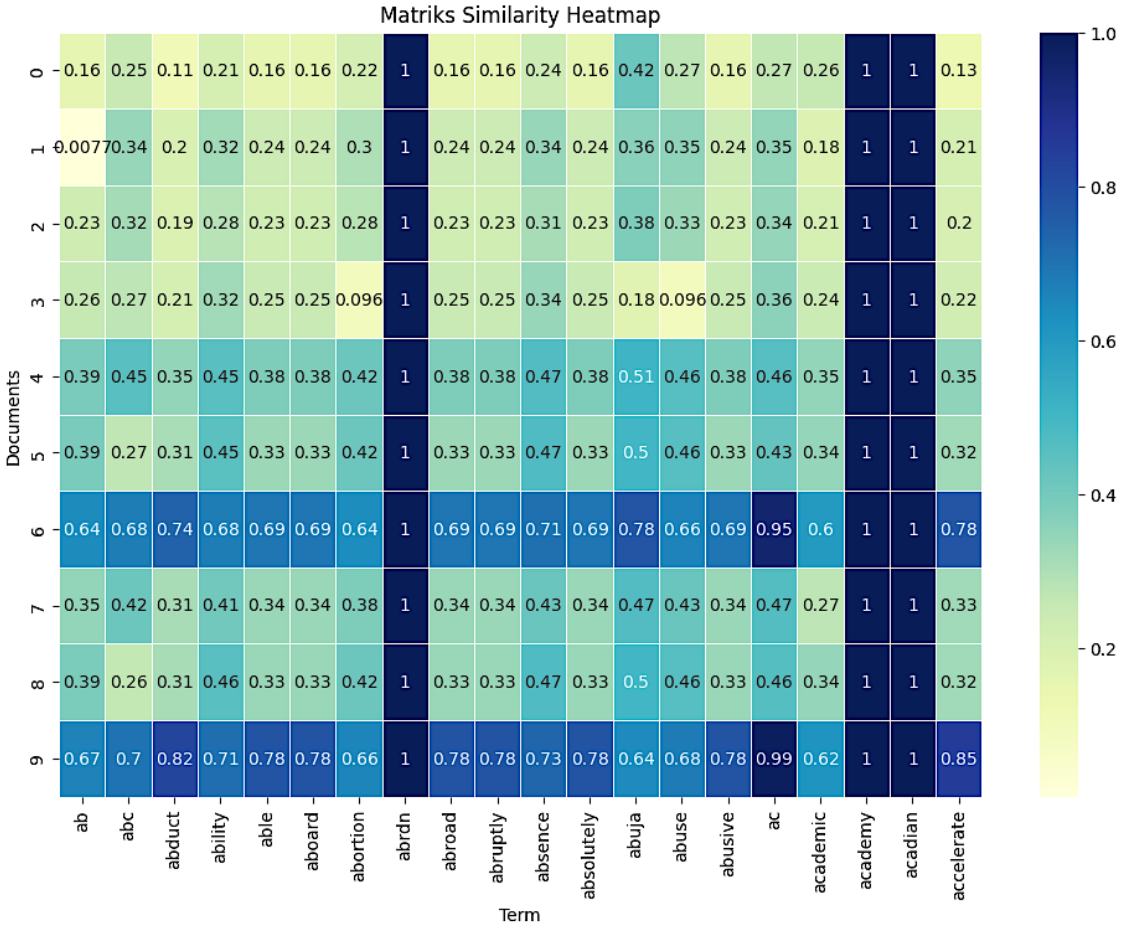


Figure 4. Heatmap of Similarity Matrix after Scalling

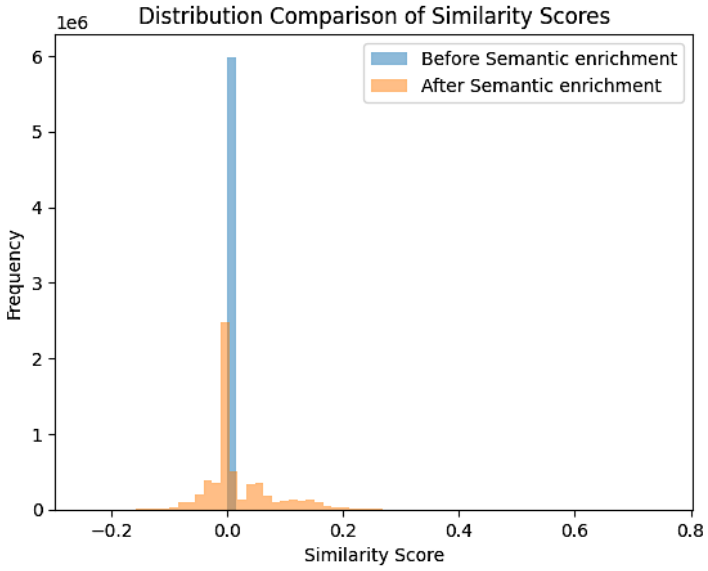
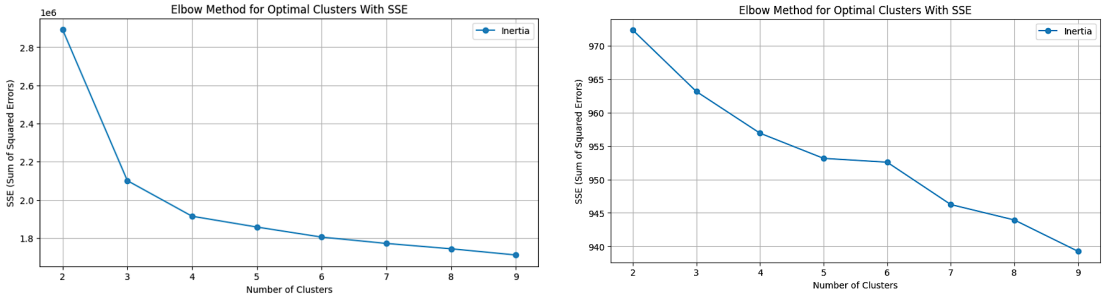


Figure 5. Comparison Before and After Semantic Enrichment

3.3. Clustering Results Using K-Means

The similarity matrix obtained from the previous process, which incorporates semantic information from WordNet, was used as input for the clustering process. Based on experiments using the elbow method, Figure 6 shows that the optimal clustering point occurs at 3 clusters. For comparison, using the same dataset composition, the syntactic TF-IDF extraction method achieved its optimal elbow point at 5 clusters.



(a) (b)

Figure 6. Determination of the Number of Clusters Using the Elbow Method
(a) Proposed Method (b) TF-IDF

From the 20 data categories tested, the resulting number of clusters did not match the existing categories. This discrepancy is caused by misalignment between the dataset's inherent categories and the assigned clustering. The following are examples of data grouped within a single cluster.

Table 9. Sample Corpus with misaligned categories

corpus	category	cluster
What's Israel's Hannibal Directive? A former IDF soldier tells all. The controversial policy to avoid capture of Israeli soldiers isn't formally in place now. But echoes persist in Gaza.	Architecture	0
Israeli-Palestinian Peace Camp Shaken But Determined. The Israel-Palestinian peace camp has long promoted dialogue against hatred and bloodshed but the passions inflamed by the deadliest Gaza war yet pose entirely new challenges for the movement.	Climate	0

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

corpus	category	cluster
Iran warns US, Israel of 'harsh consequences' if war crimes continue in Gaza. Iran's foreign minister has warned the United States and Israel that they will face "harsh consequences" if they fail to permanently stop war crimes in the Gaza Strip committed during the genocidal war on the besieged Palestinian territory.	Food	0

Although the categories differ, the text descriptions discuss Israel and Gaza. Consequently, the computation automatically groups them into a single cluster. This task does not utilize a confusion matrix due to these conditions inherent in the problem. The following are the most frequent words appearing in clusters 1 through 3.

Table 10. Frequently Occurring Words in Each Cluster

Cluster	top words	Frequency
0	[market, new, say, report, gaza]	[57, 44, 38, 30, 27]
1	[report, share, free, inc, nyse]	[339, 322, 225, 219, 203]
2	[report, share, new, market, say]	[94, 60, 53, 53, 49]

From Table 10, it can be observed that some words appear in multiple clusters. This indicates that the representation with added semantic information does not group documents based on a single dominant keyword, but rather on the overall contextual closeness of meaning. The characteristics of similarity within a cluster can also be assessed from the average similarity values and their standard deviations.

Table 11. Mean Similarity and Standard Deviation from each cluster

Cluster	Number of doc.	Mean Sim.	Std.
0	203	0.993	0.0051
1	507	0.985	0.0085
2	290	0.995	0.0025

From Table 11, it can be seen that the cluster separation demonstrates good quality. Clusters 0 and 2 show an average similarity above 0.99 with relatively low standard

deviations, indicating a high degree of homogeneity in context. Cluster 1, on the other hand, exhibits the lowest average similarity with a comparatively higher standard deviation, suggesting slightly more variability than the other two clusters. This indicates that the clustering process successfully groups documents with strong contextual relationships, even when they belong to different categories, as shown in Table 9.

Experiments were also conducted using several scenarios with 1,000 data points to observe the diversity of results. First Scenario (S1) tested the model with 20 categories. The relatively balanced category structure in this scenario allows evaluation of clustering performance under an evenly distributed dataset. Second Scenario (S2) maintained a balanced quantity but with 10 categories, aiming to assess model performance under higher data density. Third Scenario (S3) used an imbalanced data distribution with proportions of 30: 30: 20: 10: 10 to evaluate how the model performs under conditions resembling real-world datasets. The experimental results for all scenarios are presented with the optimal number of clusters for each scenario, which is k = 3.

Table 12. Mean Similarity and Standard Deviation from all scenario

Cluster	Number of doc.			Mean Sim.			Std		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
0	203	212	141	0.993	0.992	0.992	0.0051	0.0051	0.0054
1	507	501	629	0.987	0.986	0.989	0.0085	0.0085	0.0068
2	290	287	230	0.995	0.995	0.996	0.0025	0.0025	0.0022

From Table 12, it can be observed that the average similarity and standard deviation across the three scenarios with different data proportions remain relatively consistent. This test also presents results from the baseline method using pure TF-IDF for comparison.

Table 13. Mean Similarity and Standard Deviation from Baseline Method

Cluster	Number of doc.			Mean sim.			Std.		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
0	85	526	402	0.059	0.088	0.008	0.084	0.023	0.027
1	81	117	283	0.096	0.016	0.101	0.081	0.029	0.055

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Cluster	Number of doc.			Mean sim.			Std.		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
2	664	77	74	0.007	0.119	0.086	0.02	0.076	0.072
3	138	169	72	0.132	0.123	0.211	0.064	0.059	0.088
4	32	111	169	0.193	0.072	0.074	0.107	0.092	0.070

From Table 13, it can be observed that the maximum average similarity for the text representation using the baseline TF-IDF mode is 0.211, with the smallest standard deviation being 0.02. The optimal number of clusters, (k = 5), was determined from the average across the three scenarios.

Table 14. Silhouette Score from Proposed Method

k-cluster	Silhouette Score Proposed Method		
	S1	S2	S3
2	0,352083	0,35625	0,374306
3	0,350694	0,341667	0,334028
4	0,298611	0,289583	0,278472
5	0,259028	0,252083	0,256944
6	0,244444	0,24375	0,236111
7	0,240972	0,240972	0,235417
8	0,172917	0,245139	0,184722
9	0,172222	0,179167	0,184722

Table 14 shows that at k=3, the Silhouette Scores remain relatively high across all sessions, indicating that the clustering structure is still well-formed and coherent. While the scores slightly decrease compared to k=2, the clusters at k=3 achieve a balance between intra-cluster cohesion and inter-cluster separation, consistent with the optimal cluster number suggested by the SSE by Elbow method. Furthermore, the minimal variation across sessions S1, S2, S3 demonstrates the stability of the clustering results across different trials.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Table 15. Silhouette Score from Baseline Method

k-cluster	Silhouette Score Baseline		
	S1	S2	S3
2	0.009	0.009	0.013
3	0.009	0.011	0.016
4	0.011	0.012	0.020
5	0.011	0.012	0.022
6	0.011	0.015	0.024
7	0.013	0.016	0.017
8	0.013	0.012	0.025
9	0.010	0.010	0.018

Table 15 shows that the TF-IDF baseline exhibits very low Silhouette Scores (<0.03) across all cluster numbers. This indicates that the resulting clusters are poorly separated, with documents tending to be evenly distributed and lacking well-defined groupings. Moreover, even as the number of clusters increases, the Silhouette Scores do not show any significant improvement, highlighting the limited effectiveness of the TF-IDF representation in capturing meaningful cluster structures.

3.4. Comparison of Model Testing Results

Next, a comparison of the testing results was conducted using the TF-IDF text representation method. The comparison used the same dataset, but without the removal of zero-constant feature columns.

Table 14. Comparison of Silhouette Score

k-cluster	Silhouette Score					
	Proposed Method			TF-IDF		
	S1	S2	S3	S1	S2	S3
2	0.507	0.513	0.539	0.009	0.009	0.013
3	0.505	0.492	0.481	0.009	0.011	0.016
4	0.430	0.417	0.401	0.011	0.012	0.020
5	0.373	0.363	0.370	0.011	0.012	0.022
6	0.352	0.351	0.340	0.011	0.015	0.024

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

k-cluster	Silhouette Score					
	Proposed Method			TF-IDF		
	S1	S2	S3	S1	S2	S3
7	0.347	0.347	0.339	0.013	0.016	0.017
8	0.249	0.353	0.266	0.013	0.012	0.025
9	0.248	0.258	0.266	0.010	0.010	0.018

The comparison results in Table 14 show the clustering performance using different representations. The first is the proposed method, which incorporates semantic information from WordNet, and the second uses purely syntactic information from TF-IDF.

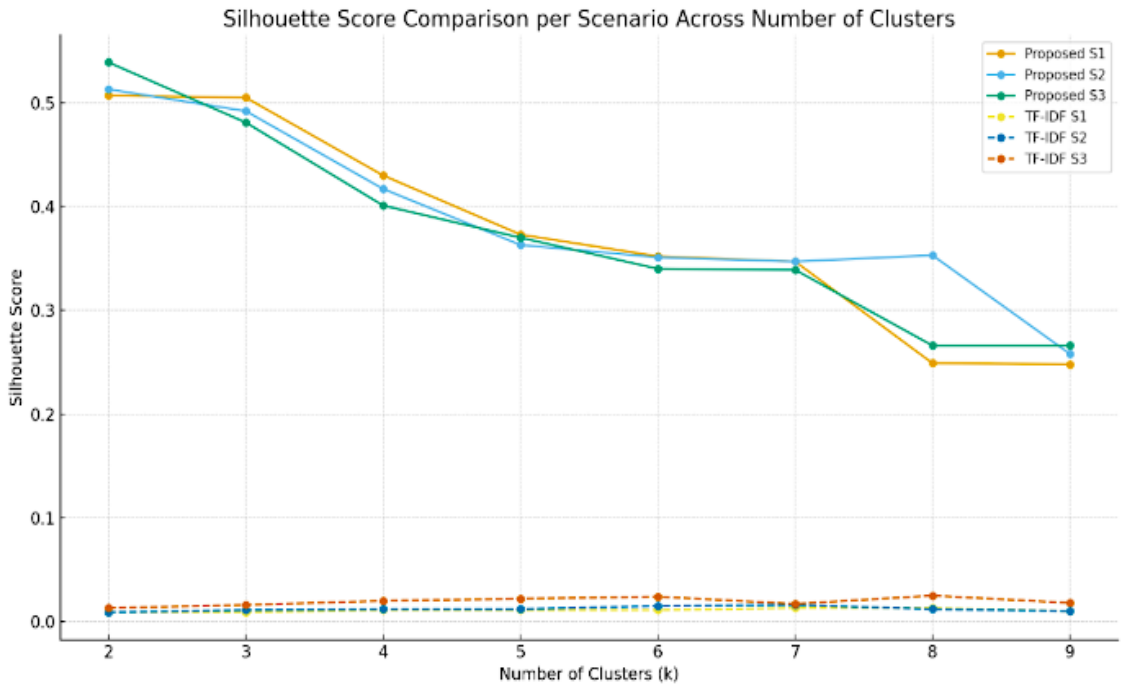


Figure 7. Silhouette Line Score

3.5. Discussion

The addition of semantic information improved the Silhouette Scores even with fewer clusters, indicating that contextual enrichment allows the algorithm to better distinguish between documents compared to frequency-based methods. This highlights the importance of incorporating semantic knowledge to capture meaningful relationships in textual data.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

In contrast, the TF-IDF baseline exhibits very low Silhouette Scores (<0.03) across all k values, showing that frequency-based representations fail to form well-separated clusters. Even as the number of clusters increases, the scores remain relatively stable, demonstrating the limited capability of TF-IDF in capturing contextual similarity.

Figure 7 illustrates that the proposed method significantly outperforms TF-IDF. Silhouette Scores for the proposed method range from 0.26 to 0.52, with optimal performance at k=2 and k=3. This demonstrates that the method produces compact and coherent clusters, whereas TF-IDF fails to generate meaningful separations.

The decreasing trend of Silhouette Scores with increasing k suggests that the data naturally supports only 2–3 clusters. As shown in Table 9, although the categories differ, the text descriptions in one cluster consistently discuss Israel and Gaza. Consequently, the algorithm automatically groups them into a single cluster, reflecting meaningful contextual similarity. Due to this inherent characteristic, a confusion matrix is not suitable for evaluation, and therefore this study employs internal evaluation using Silhouette Scores to assess clustering performance.

Overall, the proposed method not only forms better-separated and more compact clusters but also shows stability across different sessions. These results confirm that semantic enrichment is essential for improving clustering performance, particularly when traditional frequency-based approaches are insufficient to capture document-level contextual relationships.

4. CONCLUSION

From various experiments, it can be concluded that incorporating WordNet semantic information enhances the performance of text clustering. It is evident that in all tests, the proposed model achieved higher performance values than the comparison method. Even at the optimal number of clusters for the baseline method, the proposed method still outperformed it. A key limitation of the proposed method is its relatively high computational complexity, particularly during the semantic enrichment and similarity matrix computation phases. This may restrict scalability when applied to very large datasets and should be considered in practical implementations. Although the addition

of semantic information significantly improves clustering quality, it comes at the cost of increased computation time.

For future research, optimization strategies or approximation methods should be explored to balance high semantic accuracy with computational efficiency, while the text representation could be further enriched by incorporating advanced models such as BERT, Word2Vec, or similar approaches, thereby enhancing the semantic information captured and ensuring the method remains feasible for large-scale applications.

REFERENCES

[1] D. B. Bisandu, R. Prasad, and M. M. Liman, "Clustering news articles using efficient similarity measure and N-grams," *Int. J. Knowl. Eng. Data Min.*, vol. 5, no. 4, p. 333, 2018, doi: 10.1504/IJKEDM.2018.095525.

[2] N. Disayiram and R. A. H. M. Rupasingha, "A comparative study of clustering english news articles using clustering algorithms," in *2022 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, Colombo, Sri Lanka: IEEE, Sept. 2022, pp. 108–113. doi: 10.1109/SCSE56529.2022.9905210.

[3] C. Bouras and V. Tsogkas, "A clustering technique for news articles using WordNet," *Knowl.-Based Syst.*, vol. 36, pp. 115–128, Dec. 2012, doi: 10.1016/j.knosys.2012.06.015.

[4] A. El-Hamdouchi, "Comparison of hierarchic agglomerative clustering methods for document retrieval," *Comput. J.*, vol. 32, no. 3, pp. 220–227, Mar. 1989, doi: 10.1093/comjnl/32.3.220.

[5] A. Subakti, H. Murfi, and N. Hariadi, "The performance of BERT as data representation of text clustering," *J. Big Data*, vol. 9, no. 1, p. 15, Dec. 2022, doi: 10.1186/s40537-022-00564-9.

[6] Z. Chen, C. Mi, S. Duo, J. He, and Y. Zhou, "ClusTop: An unsupervised and integrated text clustering and topic extraction framework," Jan. 03, 2023, *arXiv*: arXiv:2301.00818. doi: 10.48550/arXiv.2301.00818.

[7] H. T. A. Simanjuntak, P. E. P. Silaban, J. K. S. Manurung, and V. H. Sormin, "Klasterisasi berita bahasa indonesia dengan menggunakan k-means dan word embedding," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 10, no. 3, pp. 641–652, July 2023, doi: 10.25126/jtiik.20231026468.

[8] S.-W. Kim and J.-M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, p. 30, Dec. 2019, doi: 10.1186/s13673-019-0192-7.

[9] E. Kurniawan and N. Hendrastuty, "Penerapan algoritma k-means untuk melakukan klasterisasi pada peringkasan teks," *J. Inform. Teknol. Dan Sains Jinteks*, vol. 6, no. 3, pp. 514–520, Aug. 2024, doi: 10.51401/jinteks.v6i3.4435.

[10] Aubaidan, "Comparative study of k-means and k-means++ clustering algorithms on crime domain," *J. Comput. Sci.*, vol. 10, no. 7, pp. 1197–1206, July 2014, doi: 10.3844/jcssp.2014.1197.1206.

[11] L. M. Abualigah, A. T. Khader, and M. A. Al-Betar, "Multi-objectives-based text clustering technique using K-mean algorithm," in *2016 7th International Conference on Computer Science and Information Technology (CSIT)*, Amman, Jordan: IEEE, July 2016, pp. 1–6. doi: 10.1109/CSIT.2016.7549464.

[12] J. Ravi and S. Kulkarni, "Text embedding techniques for efficient clustering of twitter data," *Evol. Intell.*, vol. 16, no. 5, pp. 1667–1677, Oct. 2023, doi: 10.1007/s12065-023-00825-3.

[13] K. K. Saravanakumar, M. Ballesteros, M. K. Chandrasekaran, and K. McKeown, "Event-driven news stream clustering using entity-aware contextual embeddings," Jan. 26, 2021, *arXiv*: arXiv:2101.11059. doi: 10.48550/arXiv.2101.11059.

[14] S. Yeasmin, N. Afrin, and M. R. Huq, "Transformer-based text clustering for newspaper articles," in *machine intelligence and emerging technologies*, vol. 490, Md. S. Satu, M. A. Moni, M. S. Kaiser, and M. S. Arefin, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 490. , Cham: Springer Nature Switzerland, 2023, pp. 443–457. doi: 10.1007/978-3-031-34619-4_35.

[15] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, "A semantic approach for text clustering using WordNet and lexical chains," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2264–2275, Mar. 2015, doi: 10.1016/j.eswa.2014.10.023.

[16] Kumar Saksham, "Global News Dataset." Kaggle. doi: 10.34740/KAGGLE/DSV/7105651.

[17] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to information retrieval, 1st ed. Cambridge University Press, 2008. doi: 10.1017/CBO9780511809071.

[18] G. U. Abriani and M. A. Yaqin, "Implementasi metode semantic similarity untuk pengukuran kemiripan makna antar kalimat," *Ilk. J. Comput. Sci. Appl. Inform.*, vol. 1, no. 2, pp. 47–57, Dec. 2019, doi: 10.28926/ilkomnika.v1i2.15.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[19] B. Montolalu and S. Rochimah, "Deteksi konflik leksikal pada diagram kelas menggunakan modifikasi graf dan similaritas wordnet," *Syst. Inf. Syst. Inform. J.*, vol. 3, no. 1, pp. 1–8, Aug. 2017, doi: 10.29080/systemic.v3i1.187.

[20] A. Géron, Hands-On machine learning with scikit-learn, keras, and tensorflow: concepts, tools, and techniques to build intelligent systems, 2nd ed. Sebastopol: O'Reilly, 2019.

[21] F. Malik, S. Khan, A. Rizwan, G. Atteia, and N. A. Samee, "A novel hybrid clustering approach based on black hole algorithm for document clustering," *IEEE Access*, vol. 10, pp. 97310–97326, 2022, doi: 10.1109/ACCESS.2022.3202017.

[22] J. Han and M. Kamber, *Data mining: concepts and techniques*, 3rd ed. Burlington, MA: Elsevier, 2012.

[23] M. J. P. Canon, L. L. Maceda, and C. Y. Sy, "Clustering with enhanced word embeddings for contextual analysis in academic texts," *Int. J. Eng. Trends Technol.*, vol. 72, no. 6, pp. 170–177, June 2024, doi: 10.14445/22315381/IJETT-V72I6P118.

[24] S. Das and U. Mert Cakmak, *Hands-On Automated Machine Learning*. Sciendo, 2018. doi: 10.0000/9781788622288.

[25] C. C. Aggarwal, *Data Mining: The Textbook*. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-14142-8.