Journal of Information Systems and Informatics
Vol. 7, No. 3, September 2025 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v7i3.1180 Published By DRPM-UBD

Enhancing News Similarity with Chunking Strategy and Hyperparameter Setting on Hybrid SBERT - Node2Vec Model

Reza Ananta Permadi Supriyo[1], Urip Teguh Setijohatmo[2], Asri Maspupah[3]

[1,2,3]Computer Engineering and Informatics Department, Bandung State Polytechnic, Bandung, Indonesia
Email: [1]reza.ananta.tif421@polban.ac.id, [2]urip@jtk.polban.ac.id*, [3]asri.maspupah@polban.ac.id

Abstract

The proliferation of online news necessitates accurate article similarity systems to combat information overload, yet models based solely on semantic content often ignore crucial structural context like news source and publication date. This research proposes and evaluates a hybrid embedding model that integrates semantic representations from Sentence-BERT (SBERT) with structural representations from Node2Vec. A series of quantitative experiments were conducted on the challenging, multilingual SPICED dataset to determine the optimal model configuration. Using Mean Squared Error (MSE) for evaluation, the results show that a per-paragraph chunking strategy yielded the best performance. This strategy's effectiveness was validated by the identical performance of an optimal fixed-size chunk (450 characters with a 64 overlap), a value that aligns closely with the dataset's average paragraph length. Furthermore, a community-focused (BFS-like) Node2Vec configuration (p=1.0, q=2.0, l=60) was identified as optimal for the structural component. Significantly, the final hybrid model (MSE = 0.1434) proved superior to both the purely semantic (MSE = 0.1449) and purely structural models (MSE = 0.2512). This study concludes that the fusion of content and context provides the most comprehensive and accurate representation for news similarity detection.

Keywords: News Similarity, SBERT, Node2Vec, Chunking, Graph Embedding

1. INTRODUCTION

Text Similarity is a fundamental task in Natural Language Processing (NLP) that quantitatively measures the degree of semantic relatedness between two or more text documents [1]. The ability to accurately compute this similarity is crucial for a wide range of applications, including information retrieval, document clustering, and, most pertinent to this study, content recommendation systems[2]. In the contemporary digital era, which is characterized by an information explosion on online news portals, robust similarity measurement is necessary to enhance recommendation relevance and combat information overload. Traditional

methods based on word frequency, such as TF-IDF, possess inherent limitations in comprehending the semantic meaning of text [3]. These approaches often fail to address critical linguistic challenges like synonymy (different words with similar meanings) and polysemy (a single word with multiple meanings), leading to less accurate and relevant similarity assessments for users [4].

To address these semantic challenges, modern research has pivoted towards Transformer-based language models such as BERT [5] and its derivative, Sentence-BERT (SBERT) [6], which have demonstrated a profound ability to understand contextual meaning. However, models focusing solely on textual content often overlook crucial meta-structural context. An article's source, category, and publication date provide structural signals that are vital for determining its relationship to other articles. This contextual information allows for the data to be modeled as a graph network, where each article is a node and the shared metadata form the relationships between them. The challenge then becomes how to convert this complex topological structure into a numerical format that a machine learning model can use. The process of encoding nodes into a low-dimensional vector space while preserving this network structure is known as graph embedding [7]. This structural dimension can be effectively captured using graph embedding algorithms like Node2Vec [8], which learn node representations from network topology.

Specifically, this research aims to answer three key questions. First, does the proposed hybrid model demonstrate superior performance compared to models that rely solely on either semantic or structural embeddings alone? Second, which text chunking strategy—such as per-sentence, per-paragraph, or fixed-size segmentation—is most effective for generating a high-quality semantic representation? Finally, how does the optimization of Node2Vec's key hyperparameters (returnFactor, inOutFactor, and walkLength) influence the quality of the structural embedding?

Therefore, this study proposes and evaluates a hybrid model that integrates semantic embeddings from SBERT with structural context embeddings from Node2Vec, inspired by approaches that combine textual and graph-based features for enhanced representation [9]. Instead of using GNN as [9], our work use more on building each component using embedding approach. The primary hypothesis is that a fusion of content (semantics) and context (structure) yields a more comprehensive and superior similarity measurement, as the inclusion of contextual information has been shown to improve performance in news recommendation systems [5]. The main contribution of this paper affected by embedding approaches is the systematic analysis of two critical optimization factors. The first is an investigation into the impact of various text chunking strategies, a crucial preprocessing step for effective dense passage retrieval [9]. The second

contribution is an analysis of Node2Vec hyperparameter optimization, specifically how the biased random walk parameters (returnFactor and inOutFactor) influence the quality of the final structural embedding, a key aspect of the algorithm's design [6]. These in our work are completed with hybrid methods.

The remainder of this paper is organized as follows. Section 2 details the research methodology. Section 3 presents and discusses the experimental results. Finally, Section 4 concludes the paper and suggests directions for future work.

2. METHODS

This research employs a quantitative experimental design to construct and evaluate a hybrid embedding model. The workflow proceeds chronologically from problem definition, analysis of problem domain, set up of experiment, the development of software for experiment, conducting experiment and discuss the results on quantitative evaluation against a ground truth dataset. The overall process is depicted in Figure 1.

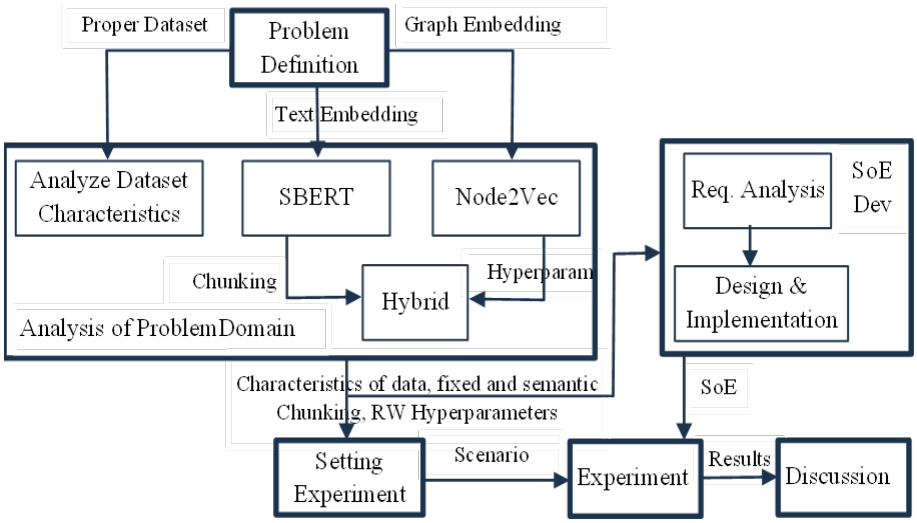

Figure 1. The end-to-end research workflow from problem definition to model evaluation

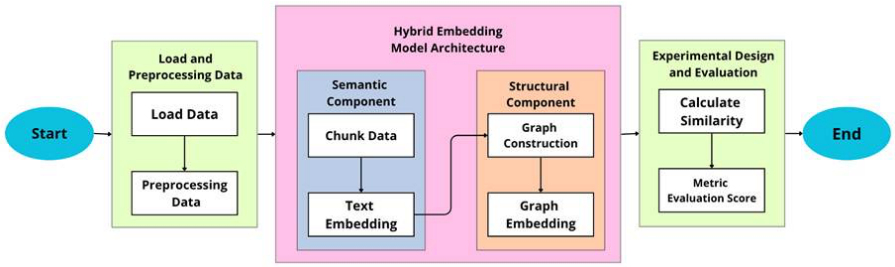

Figure 2. Req. Analysis of SoE

2.1. Problem Definition

News and its related downstream tasks have one of the essential requirements, news similarity. This can be computed using some approaches, for example machine learning as [9] or other approaches like utilizing LLM models for example. This research used one of LLM models which consider embedding methods on domain of news as the case study. News is composed of sentences and therefore measuring the similarity intrinsically can be done by sentence embedding. On the other hand, news has structural attributes that can further improve the precision especially related to the downstream tasks. This research considers attributes related to news recommendation, as publishing stuffs in this case, to include on computing news similarity. The sentence embedding itself has the drawback in which related sentences can be missed. This drawback can be lifted by organizing the sentences to explicitly represented by the graph structure in the form of chunking. So chunking and structural attributes of news perfectly forming to graph structure and therefore yield to graph embedding. To conduct this research dataset representing news is taken from SPICED dataset of Zenodo and its characteristics are subject to be explored. This data characteristics are useful to guide the best experiment setting.

2.2. Sbert

Text Embedding can be solved by many approaches starting from word embedding, sentence embedding and document embedding [10]. This research use utility SentenceTransformer available from Graph GDS of Neo4J as sentence level text embedding. The SentenceTransformer works based on Sentence Bert, Sbert, one of Bert variants.

2.3. Node2Vec

On the other hand, as problem definition revealed, graph embedding is potential embedding candidates too, where in this research, chose Node2Vec available from

Graph GDS of Neo4J. The main mechanisme of Node2Vec is gathering facts in the form of nodes and their relationships through random walk [11]. The behaviour of random walk depends on the hyperparameters that in general leading to more BFS or DFS like representing gathering local and global facts.

2.4. Hybrid

Each method has its own advantages. This invite the idea of how if both methods are combined. On the semantic part needs to manage sentences handling because of the technological limitation, the size of token computed. Incorporating graph to the handling can be potential solution and implemented by cutting text to the set of chunking. This chunking organized sentences can be treated as part of structural news graph along with the news attributes and edges for embedding generation. This approach spesifically explained in the development of sofware of experiment. Overall, the analysis of problem domain produces not only the charateristics of data, various chanking like chunking size, chunking overlap, and random walk hyperParameter but also the algorithm of model behavior. Both outputs are then feed to subsequent two independent processes, the development of software of experiment and experiment planning.

2.5. The development of Software of Experiment

To conduct experiment required software to support experiment, so it is developed based on requirement analysis and followed by design and implementation afterwards.

2.5.1. Load and Preprocessing Data

The data preparation pipeline consists of three main chronological stages: data loading, preprocessing, and content enrichment. First, the data loading stage involves reading the ground truth dataset, sourced from the Zenodo repository[12], into memory. This process utilizes the panda's library to parse the source file (zenodo_release_data.csv) into a structured DataFrame, making the data accessible for programmatic manipulation.

Following this, a preprocessing stage is performed to restructure the raw data for the subsequent steps. This is crucial for creating a clear "working map" of all unique articles that need to be processed. In this research, preprocessing includes splitting the content.pair_id column (e.g., '123_456') to extract the individual IDs for each article in a pair and mapping these unique IDs to their corresponding URLs and titles.

Crucially, the initial dataset does not contain the full text content of the articles, which is essential for semantic analysis. Therefore, an essential data acquisition and enrichment step was performed. A Python script utilizing the newspaper3k library was developed to iterate through the unique URLs identified during preprocessing, scrape the full text content for each article, and compile a comprehensive main corpus. This enriched corpus, containing both the full text and its original metadata, served as the foundation for building the knowledge graph and training the embedding models. The content similarity score from the original ground truth file was subsequently used as the target value for calculating the Mean Squared Error during model evaluation.

The main corpus used for model construction consists of 26,788 unique news articles sourced from the SPICED dataset. This dataset is characterized by its significant linguistic diversity, containing articles in multiple languages including, but not limited to, English, Mandarin, German, Spanish, and Arabic, originating from over 3,938 distinct news sources. This rich and varied dataset provides a robust foundation for training the embedding models and ensures that the findings are tested across a challenging, real-world distribution of news content.

A descriptive statistical analysis revealed a notably consistent structure within this diverse corpus: the average article consists of approximately 16 sentences, with an average sentence length of around 24 words. This statistical profile has significant implications for the experimental design of the chunking strategies. The moderate length of the articles validates the use of aggregation techniques to form a single document embedding. Most importantly, the average paragraph length (empirically found to be around 76 words) provides a strong justification for the parameters tested in the fixed-size chunking experiments. A chunk size of 450 characters, which was found to be optimal, effectively captures a representative, paragraph-level semantic unit. This data-driven approach ensures that the experimental setup for chunking is well-aligned with the intrinsic properties of the corpus.

2.5.2. Hybrid Embedding Model Architecture

The proposed methodology is based on a hybrid architecture that generates two distinct embeddings—one semantic and one structural—which are then fused into a final comprehensive vector. This approach is motivated by research demonstrating that integrating textual features and graph-based representations can lead to more robust and effective models for various NLP tasks, such as text classification[13]. The resulting final vector serves as a rich, unified representation for each news article, and its effectiveness in measuring similarity is subsequently evaluated quantitatively against a human-annotated ground truth dataset.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.5.2.1. Semantic Component (SBERT)

The initial stage of the methodology focuses on processing the textual content to generate the semantic representation (sbertEmbedding) while simultaneously constructing the foundational knowledge graph. First, the text from each article is segmented into smaller units (chunks) [14]. Based on preliminary experiments, a per-paragraph chunking strategy was identified as optimal, as it best preserves the author's thought flow.

During this process, a knowledge graph is created in Neo4j, where each article is represented as a :Document node and each chunk becomes a :Chunk node, linked by a :HAS_CHUNK relationship. The pre-trained SBERT model, paraphrase-multilingual-MiniLM-L12-v2, is then employed to generate a vector embedding for the article's title and each individual chunk. This process leverages the architecture proposed by [6], which is specifically designed to produce semantically meaningful sentence embeddings. The resulting chunk embeddings are stored as a property on each :Chunk node.

Finally, to create the document-level semantic vector, these chunk vectors are aggregated via a weighted average, assigning a higher weight to the title's embedding. This produces a single, content-rich sbertEmbedding vector, which is then stored as a property on the corresponding :Document node. The output of this stage is therefore twofold: a graph populated with documents and their semantically-embedded chunks, and a high-level semantic vector for each document.

2.5.2.2. Structural Component (Node2Vec)

The structural context representation (structuralEmbedding) is derived by enriching the same knowledge graph that was constructed during the semantic component stage. After the graph has been populated with :Document, :Chunk, :Source, and :Date nodes, this stage focuses on modeling the higher-level relationships between documents based on their shared metadata. Contextual relationships such as :SAME_SOURCE and :PUBLISHED_NEAR are created between :Document nodes.

The Node2Vec algorithm, implemented with the Neo4j Graph Data Science (GDS) library, is then executed on this enriched structural graph. It learns a topological representation for each document by simulating biased random walks, controlled by returnFactor (p) and inOutFactor (q) hyperparameters, as proposed by [8]. This process generates the final structuralEmbedding, which captures the position and context of each article within the news ecosystem.

The literature, such as the optimization research by [15], identifies several key hyperparameters that significantly influence Node2Vec's performance, including embedding dimension (d), number of random walks (r), walk length (l), return parameter (p), and in-out parameter (q).

In this study, the experimental focus was specifically directed at the three parameters that control the graph traversal strategy: walkLength (l), returnFactor (p), and inOutFactor (q). The range of values tested for returnFactor (p) and inOutFactor (q) was adapted from the hyperparameter space explored in the study by [15], which systematically evaluated these values to understand their impact on various graph types. Meanwhile, the selection of the range for walkLength was specifically adapted to the news data domain used in this research, where a moderate walk length was tested to find the optimal balance between capturing sufficient structural context and avoiding noise from overly distant exploration.

The parameters d (embedding dimension) and r (number of walks) were not varied and were set to commonly used standard values. This decision was based on computational efficiency and the research's primary focus on analyzing the quality of the traversal strategy rather than the quantity of walks or the representational capacity of the resulting vectors.

2.6. Experimental Design and Evaluation

This step indicates what and how to running of experimental design. A two-stage experimental design was employed to optimize the model. First, a comparative analysis of chunking strategies was conducted. Second, the hyperparameters of the Node2Vec algorithm were tuned. The performance of each model configuration (Semantic-only, Structural Context-only, and Hybrid) was quantitatively evaluated against the ground truth dataset. Cosine Similarity was used to calculate a similarity score between the embedding vectors of each article pair. This predicted score was then compared against the ground truth score using Mean Squared Error (MSE) as the primary metric. MSE was chosen for its sensitivity to large prediction errors, which are particularly detrimental in this context.

2.6.1. Cosine Similarity

Cosine Similarity is a metric used to measure the similarity between two non-zero vectors in a multi-dimensional space. Instead of measuring Euclidean distance, this metric calculates the cosine of the angle between the two vectors, thus focusing on their orientation rather than their magnitude[16]. The resulting score ranges from -1 to 1, where 1 indicates identical orientation (0° angle), 0 indicates orthogonality (90° angle), and -1 indicates opposite orientation.

The similarity between two vectors, A and B, is calculated by taking their dot product and dividing it by the product of their magnitudes (L2 norms), as shown in (1).

$$\text{Similarity}(A,B) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

This normalization by vector magnitude makes the metric insensitive to factors like document length and focuses solely on the semantic orientation of the text embeddings. In this research, Cosine Similarity is the core function used to generate a predicted similarity score from the output embedding vectors of each model configuration (Semantic, Structural, and Hybrid). This score is then used as the predictive value that is subsequently evaluated against the ground truth using the Mean Squared Error (MSE) metric.

2.6.2. Mean Squared Error (MSE)

In addition to correlation, model performance can be evaluated as a regression problem, where the objective is for the model-generated similarity scores to predict the ground truth scores as accurately as possible[17]. The most common metric for measuring the error rate in regression problems is the Mean Squared Error (MSE). MSE measures the average of the squared differences between the predicted and actual values. The error is squared for two primary reasons: first, to eliminate negative values so that errors do not cancel each other out, and second, to give a larger penalty to large errors compared to small ones. The mathematical formula for MSE is as follows.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (2)$$

Where:

n is the total number of data pairs being evaluated.

Y_i is the i-th actual value (in this case, the human_score from the ground

$\hat{Y}_i$ is the i-th value predicted by the model (the resulting cosine similarity score).

Unlike correlation, the MSE value has no upper bound, but its interpretation is straightforward: the lower the MSE value (approaching 0), the better the model's performance, as it signifies a smaller average distance between the prediction and the reality. Given that both the ground truth and the predicted model scores in this research operate within a defined range between -1 and 1, an MSE score close to zero definitively indicates a very low prediction error and excellent model performance. In this study, MSE is used as the primary evaluation metric to assess

how accurately a model can "guess" the similarity score provided by human annotators [18].

3. RESULTS AND DISCUSSION

This section presents and analyzes the quantitative results from the series of experiments conducted. The analysis is structured to correspond with the research questions, beginning with the optimization of the model's individual components, followed by an evaluation of the final hybrid model. To provide a concrete and intuitive understanding of the underlying mechanisms, the discussion is supplemented with a detailed process flow simulation illustrating how the model handles specific cases.

3.1. Process Flow Simulation: From Raw Data to Final Similarity Score

The purpose of this simulation is to trace two different news articles through the entire system to demonstrate how the hybrid model produces a final similarity score.

Step 1: Initial Data (Input)

To illustrate the methodological workflow, this simulation utilizes two representative news articles drawn from the preprocessed dataset. These articles are intentionally chosen because they contain the word "bunga" with entirely different meanings (a case of polysemy).

Table 1. Input Data

ID	Title	Content (Indonesia)	Source	Date
1	Suku Bunga Acuan BI Diperkirakan Naik	"Bank Indonesia (BI) memberikan sinyal kenaikan suku bunga acuan..."	ekonomi.com	2025-07-08
2	Pameran Bunga Mawar di Bandung	"Taman kota Bandung dipenuhi ribuan bunga mawar yang sedang mekar..."	wisatabandung.id	2025-07-08

Step 2: Semantic Embedding Process (SBERT)

This stage aims to generate the sbertEmbedding that represents the content of each article.

2a. Chunking:

The content of each article is segmented into smaller units (chunks), for example, per paragraph or per sentence.

a) Chunk 1-1: "Bank Indonesia (BI) memberikan sinyal kenaikan suku bunga acuan."
b) Chunk 2-1: "Taman kota Bandung dipenuhi ribuan bunga mawar yang sedang mekar."

2b. Tokenization & Input Embedding:

Each chunk (and title) is tokenized into sub-word tokens with special tokens like [CLS] and [SEP] added. Each token is then converted into an initial vector via the summation of Token, Position, and Segment Embeddings[19]. At this point, the base vector for the polysemous word is still identical in both sentences.

2c. Self-Attention Mechanism (Resolving Polysemy):

The input vectors are then processed through the Transformer layers[20].

a) For the keyword in Article 1, the attention mechanism observes its neighbors like "suku" and "acuan." It assigns high attention weights to these words, pulling the final output vector towards a financial concept[21].
b) For the keyword in Article 2, attention observes neighbors like "taman" and "mawar," pulling its output vector towards a botanical concept.

2d. Pooling & Document Aggregation:

After all tokens in a chunk have their final contextual vectors, mean pooling averages them to produce a single vector representing that chunk[6]. Finally, all chunk vectors from a document are aggregated (via a weighted average giving higher importance to the title) to produce the final sbertEmbedding.

Result: V_sbert_1 (a financial vector) and V_sbert_2 (a botanical vector) will be positioned far apart in the vector space.

2e. Semantic Embedding and Graph Construction

After generating the semantic embeddings, a graph is constructed where each document and its corresponding chunks are represented as nodes. Concurrently, a structural context graph is built in Neo4j. For example, two document nodes might not share a :SAME_SOURCE relationship but could be linked by a :PUBLISHED_NEAR relationship if their publication dates are identical.

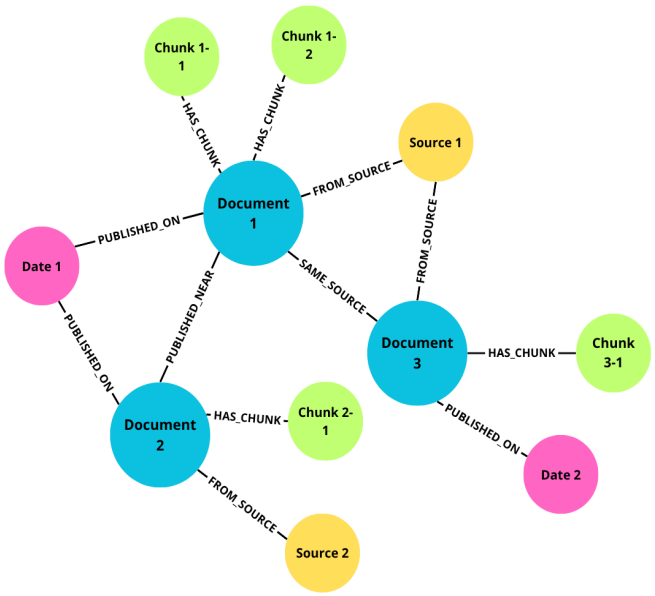

Figure 3. Graph Model

Step 3: Structural Embedding Process (Node2Vec)

This stage runs separately and considers only metadata to generate the structuralEmbedding.

3a. Biased Random Walk:

The Node2Vec algorithm, using its optimized p and q hyperparameters, explores this graph. Because articles 1 and 2 are only weakly connected, they will not frequently co-occur in the same short walks.

3b. Learning & Result:

From these walk patterns, Node2Vec learns that articles 1 and 2 do not belong to the same strong structural context.

> Result: V_struct_1 and V_struct_2 will have dissimilar vectors, reflecting their different metadata contexts.

Step 4: Hybrid Construction and Final Calculation

This final stage fuses the two types of information and produces a final score.

4a. Vector Concatenation: For each article, the two vectors are concatenated.
a) V_hybrid_1 = concatenate(V_sbert_1, V_struct_1)
b) V_hybrid_2 = concatenate(V_sbert_2, V_struct_2)

4b. Cosine Similarity Calculation:
The final step is to calculate the cosine similarity between the two hybrid vectors.
a) Final Score = Cosine_Similarity(V_hybrid_1, V_hybrid_2)

Because the two articles are highly dissimilar both semantically (V_sbert) and structurally (V_struct), their final hybrid vectors will also be very different. The cosine similarity calculation will yield a very low score (approaching 0.0). This accurately proves that the two news articles are indeed not similar, demonstrating the success of the hybrid model in capturing both dimensions.

3.2. Corpus Characteristics Analysis

To better understand the textual properties of the dataset, a descriptive statistical analysis was performed on the main corpus following the data enrichment process. This analysis focused on text length and granularity, which are critical factors for determining an effective chunking strategy. The findings reveal a notable consistency across the articles, as illustrated in Figures [2, 3, 4].

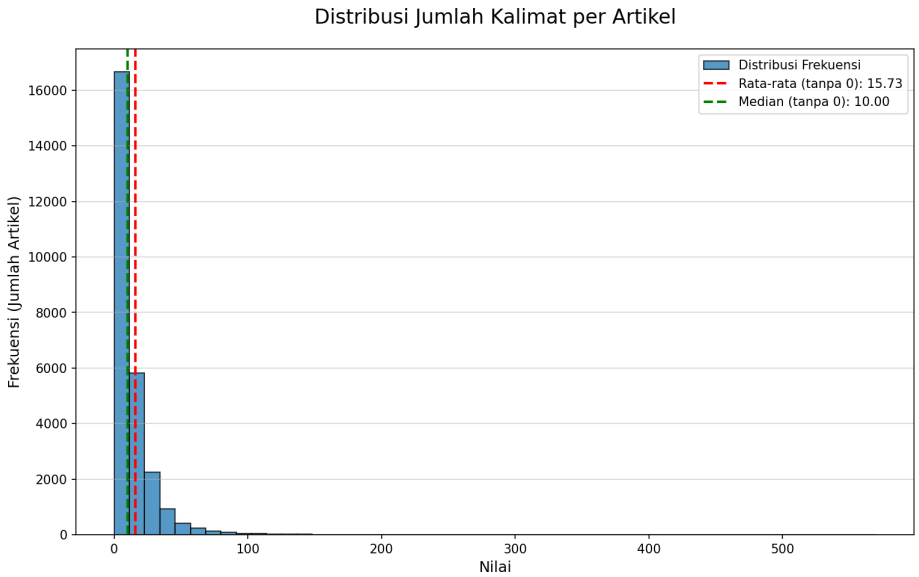

Figure 4. Distribution of Sentences per News Article

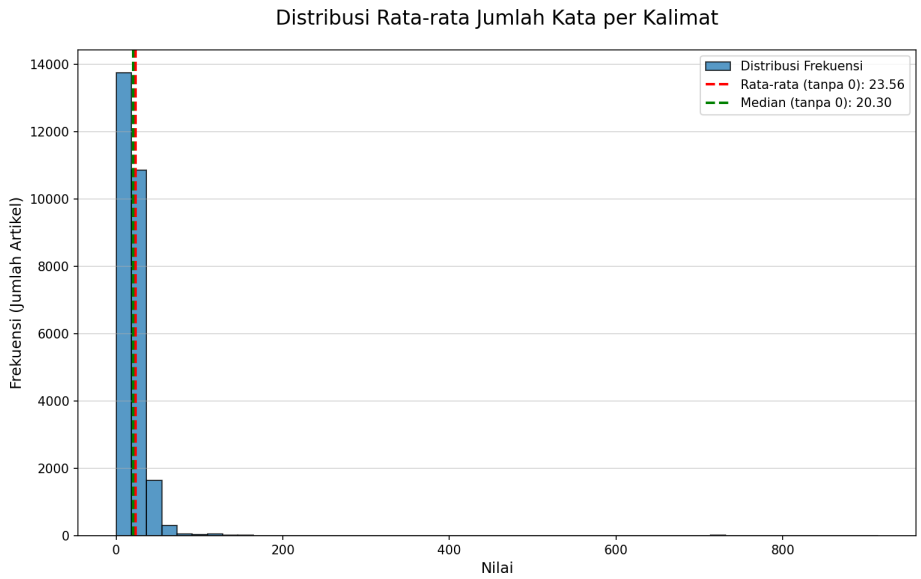

Figure 5. Distribution of Word per Sentence

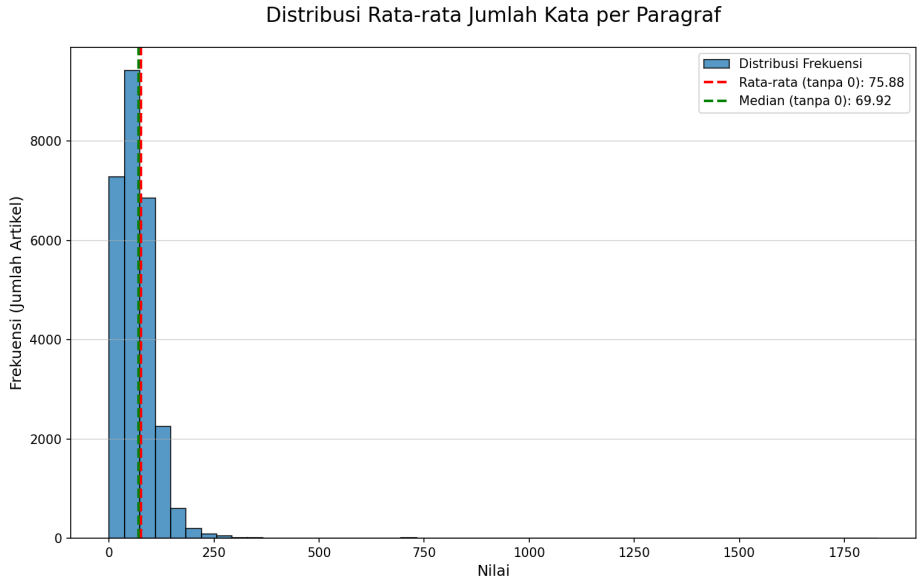

Figure 6. Distribution of Sentence per Paragraph

The key statistics are summarized as follows:

a) The average article consists of approximately 15.73 sentences.
b) The average sentence length is approximately 23.56 words.

c) A synthetic paragraph, defined as a group of four sentences, has an average length of approximately 75.88 words.

These characteristics have significant implications for the research methodology. The moderate length of the articles validates the use of aggregation techniques to form a single document embedding. The average sentence length of ~24 words is well within the input limits of the SBERT model, confirming the feasibility of sentence-level processing.

Most importantly, the average paragraph length provides a strong empirical justification for the parameters tested in the fixed-size chunking experiments. An average paragraph of ~76 words per-paragraph corresponds to a character count that is highly aligned with the chunkSize values of 450 and 512 words explored in this study. This indicates that the experimental setup was well-designed to test chunk sizes that capture a representative, paragraph-level semantic unit, which ultimately proved to be the most effective strategy. Overall, these statistical findings confirm that the dataset possesses a predictable structure, validating the experimental design choices for chunking.

3.3. Experiment 1: Comparative Analysis of Chunking Strategies

The initial experiment was designed to identify the most effective text chunking strategy for generating the semantic embedding (sbertEmbedding). The selection of an appropriate chunking strategy is a crucial preprocessing step in modern dense retrieval systems, as it significantly impacts the balance between contextual completeness and semantic granularity [22]. Therefore, various strategies were tested, and the performance of the final hybrid model was measured using Mean Squared Error (MSE), aligning with research practices that emphasize the importance of passage retrieval optimization [23]. The results are summarized in Table 2.

Table 2. Model Performance by Chunking Strategy

ID	Strategy	Parameters	MSE
1	Fixed-size	size = 512, overlap = 64	0.1599
2	Fixed-size	size = 450, overlap = 64	0.1449
3	Fixed-size	size = 450, overlap = 32	0.1511
4	Fixed-size	size = 256, overlap = 64	0.1504
5	Fixed-size	size = 256, overlap = 32	0.1518
6	Semantic	paragraph	0.1449
7	Semantic	sentence	0.1469

The results indicate that two strategies achieved an identical, superior performance: per-paragraph chunking and fixed-size chunking with a size of 450 characters and an overlap of 64. The per-sentence approach was also highly competitive. This suggests that for this dataset, chunks that capture a complete thought unit (either a natural paragraph or a sufficiently large text block) are optimal. Given its conceptual advantage of respecting the author's original semantic boundaries without arbitrary cuts, the per-paragraph chunking strategy was selected as the standard for all subsequent experiments.

3.4. Experiment 2: Node2Vec Hyperparameter Optimization

After fixing the chunking strategy, the second experiment focused on optimizing the structural component by tuning the Node2Vec hyperparameters. The goal was to find the combination of returnFactor (p), inOutFactor (q), and walkLength (t) that minimized the final model's MSE. The range of values tested for returnFactor and inOutFactor was {0.25, 0.5, 1.0, 2.0, 4.0}. This selection was directly inspired by the hyperparameter space explored in related research, which systematically evaluated these parameters to understand their impact on model performance[15].

Table 3. Performance of Top Node2Vec Hyperparameter Configurations

ID	Parameters			MSE
	returnFactor (p)	inOutFactor (q)	walkLength (t)	
1	1.0	0.25	40	0.1457
2	1.0	0.5	40	0.1452
3	1.0	1.0	40	0.1449
4	1.0	2.0	40	0.1454
5	1.0	4.0	40	0.1441
…	…	…	…	…
78	0.25	1.0	20	0.1441
79	0.5	1.0	20	0.1439
80	2.0	1.0	20	0.1457
81	4.0	1.0	20	0.1444
…	…	…	…	…
101	1.0	0.25	60	0.1438
102	1.0	0.5	60	0.1440
103	1.0	2.0	60	0.1434
…	…	…	…	…
130	4.0	1.0	80	0.452

The findings, summarized in Table 2, identify the optimal configuration as returnFactor = 1.0, inOutFactor = 2.0, and walkLength = 60. A consistent trend observed throughout the experiments was the outperformance of configurations with an inOutFactor > 1. This strongly suggests that a BFS-like exploration, which focuses on the local community structure (i.e., articles from the same source or

date), is the most effective strategy for capturing the meaningful structural context within this news graph.

3.5. Final Model Performance

Using the optimized components from the previous experiments, a final evaluation was performed to compare the performance of the proposed Hybrid model against its constituent parts.

Table 4. Final Performance Comparison of Embedding Models

Model Type	Description	MSE
Structural Context-only (Node2Vec)	Node2Vec Embedding only	0.2512
Semantic-only (SBERT)	SBERT Embedding only	0.1449
Hybrid Model (Node2Vec + SBERT)	SBERT + Node2Vec Combined	0.1434

The results in Table 4 definitively show the superiority of the hybrid approach. The Hybrid model achieved the lowest MSE (0.1434), outperforming the strong baseline set by the SBERT-only model (MSE 0.1449) and the much weaker Structural Context-only model. This supports the primary hypothesis of this research: that integrating both semantic content and structural context provides a more comprehensive and accurate measure of news article similarity. While the SBERT component provides a powerful understanding of what an article is about, the Node2Vec component adds a crucial layer of contextual information—who published it and when—that refines the similarity score to better align with nuanced human judgment. This synergy confirms that for a complex domain like news, context is a vital, non-redundant signal to content.

3.6. Discussion

The experimental results definitively demonstrate the superiority of the proposed hybrid model, which achieved the lowest Mean Squared Error (MSE) of 0.1434. This finding validates the primary hypothesis that integrating semantic content (from SBERT) and structural context (from Node2Vec) provides a more comprehensive and accurate measure of news article similarity. This section will discuss the interpretation of these findings, compare them with related work, and explore their practical implications.

3.6.1. Interpretation of Experimental Findings

The optimization experiments revealed crucial insights into the model's components. The outperformance of the per-paragraph chunking strategy (MSE 0.1449) can be attributed to its ability to preserve the natural semantic boundaries established by the author. Unlike per-sentence chunking, which can be too granular, or fixed-size chunking, which is arbitrary, the per-paragraph method provides SBERT with text segments that are both contextually rich and thematically focused. This is supported by the dataset's characteristics, where the optimal fixed-size chunk of 450 characters closely aligns with the average paragraph length, suggesting this size is ideal for capturing a complete thought unit.

Furthermore, the optimization of Node2Vec hyperparameters showed that an exploration strategy focused on local communities (a BFS-like walk with an inOutFactor $q > 1$) was most effective. This finding implies that for this news domain, the strongest structural signal is the "publisher ecosystem"; articles from the same source form dense, informative communities, and a model that learns to recognize this community structure performs best.

3.6.2. Comparison with Related Work

The superiority of the hybrid model aligns with a significant trend in the literature where combining textual and graph-based features leads to more robust representations. While a direct numerical comparison is challenging due to differences in datasets and tasks, the principle is well-supported. For instance, the work by [22] on Text GCN demonstrated that modeling a text corpus as a single graph can improve classification tasks. Similarly, research in news recommendation, such as that by [7], has shown that enriching graph embeddings with contextual information enhances performance. Our findings contribute to this body of work by empirically demonstrating that even a straightforward concatenation of independently optimized SBERT and Node2Vec embeddings provides a significant performance lift for similarity detection, suggesting that structural context is a vital, non-redundant signal to content.

3.6.3. Practical Implications for News Recommendation Systems

The findings of this research offer several practical implications for the development of real-world news aggregation and recommendation platforms:

1) The results strongly suggest that platforms should move beyond purely content-based similarity. By incorporating structural metadata (source, publication date), a recommendation engine can make more nuanced decisions, such as differentiating between an original breaking news report

and a follow-up analysis from the same publisher, even if their text is highly similar.

2) The community-focused nature of the optimal Node2Vec configuration is particularly valuable. It allows the system to identify articles that belong to the same "news ecosystem" or developing story. This can be used to improve the coherence of recommended articles or, conversely, to increase diversification by avoiding recommendations of multiple, nearly identical articles from a single source.
3) The modular, two-stage approach (semantic embedding followed by structural embedding) provides a practical and scalable blueprint for implementation. The computationally intensive SBERT embedding process can be run as new articles are ingested, while the Node2Vec component can be periodically re-trained to update the structural context of the entire news graph. This balance makes the sophisticated hybrid approach feasible in a production environment.

This study not only validates a specific hybrid model but also provides a strong argument for a more holistic approach to news similarity, where the relationships between articles are considered as important as the content within them.

4. CONCLUSION

This research successfully developed and evaluated a hybrid embedding model for news article similarity, yielding three primary conclusions. First, the proposed hybrid model, which integrates semantic embeddings from SBERT and structural embeddings from Node2Vec, is demonstrably superior to single-component approaches. The final hybrid configuration achieved the lowest Mean Squared Error (MSE) of 0.1434, confirming the hypothesis that combining content and context leads to a more accurate and robust similarity measure. Second, the optimization experiments revealed crucial insights into the optimal configuration for each component. The comparative analysis of preprocessing methods showed that a per-paragraph chunking strategy (along with a fixed-size chunk of 450 characters) was most effective. This aligns with the dataset's characteristic average paragraph length (approximately 76 words per-paragraph or ~450 words in one news article), indicating that this size provides the best balance of rich context and thematic focus for SBERT to process. Finally, hyperparameter optimization of Node2Vec showed that an exploration strategy focused on local communities (a BFS-like walk with an inOutFactor $q > 1$) was most effective at capturing relevant structural signals. This finding implies that the strongest structural signal in this news domain is the "publisher ecosystem"; articles from the same source form dense, informative communities, and a model that learns to recognize this community structure performs best.

These findings collectively demonstrate that a multi-faceted approach, which considers both the semantic content of an article and its structural position within the broader information ecosystem, is crucial for nuanced similarity detection. The methodology presented offers a robust blueprint for developing more effective news recommendation and analysis systems. Finally, several promising directions could build upon these results. At least four paths should be further explored for future works. The First path is the implementation of end-to-end Graph Neural Network (GNN) models [9], which could learn the interaction between content features and graph structure simultaneously compared and contrasted to our work on the same problem domain and environment, yielding further performance analysis. The second path is continuous periodically computation handling, perhaps on daily basis, for the continously news production, means fast recompute requirement. Even more a critical requirement for real-time news environments. So other methods such as FastRP should be considered [24]. The third path is the model's contextual understanding could also be significantly enhanced by enriching the knowledge graph with additional entity nodes (such as :Author or :Location) extracted via Named Entity Recognition (NER) [25]. Furthermore, the fourth is testing the model's generalizability on larger, more diverse datasets or adapting it to other types of content, such as blogs or social media posts, would be a valuable step in validating the broader applicability of this hybrid approach.

REFERENCES

[1] J. Wang and Y. Dong, "Measurement of text similarity: A survey," *Inf.*, vol. 11, no. 9, pp. 1–17, 2020, doi: 10.3390/info11090421.

[2] N. Pradhan, M. Gyanchandani, and R. Wadhvani, "A Review on Text Similarity Technique used in IR and its Application," *Int. J. Comput. Appl.*, vol. 120, no. 9, pp. 29–34, 2015, doi: 10.5120/21257-4109.

[3] D. K. Wardy, I. K. G. D. Putra, and N. K. D. Rusjayanthi, "Clustering Artikel pada Portal Berita Online," *JITTER- J. Ilm. Teknol. dan Komput.*, vol. 3, no. 1, pp. 3–11, 2022.

[4] K. Erk, "Vector Space Models of Word Meaning and Phrase Meaning: A Survey," *Linguist. Lang. Compass*, vol. 6, no. 10, pp. 635–653, 2012, doi: 10.1002/lnco.362.

[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[6] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 3982–3992, 2019, doi: 10.18653/v1/d19-1410.

[7] H.-S. Sheu and S. Li, "Context-aware Graph Embedding for Session-based News Recommendation by," 2020.
[8] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," *HHS Public Access*, pp. 607–612, 2016, doi: 10.1145/2939672.2939754.node2vec.
[9] J. Wu, J. Sun, H. Sun, and G. Sun, "Performance Analysis of Graph Neural Network Frameworks," *Proc. - 2021 IEEE Int. Symp. Perform. Anal. Syst. Software, ISPASS 2021*, pp. 118–127, 2021, doi: 10.1109/ISPASS51385.2021.00029.
[10] L. Wu *et al.*, "Word mover's embedding: From word2vec to document embedding," *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018*, pp. 4524–4534, 2018, doi: 10.18653/v1/d18-1482.
[11] L. Meng and N. Masuda, "Analysis of node2vec random walks on networks: Node2vec random walks on networks," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 476, no. 2243, 2020, doi: 10.1098/rspa.2020.0447.
[12] E. Shushkevich, M. V. Loureiro, L. Mai, S. Derby, and T. K. Wijaya, "SPICED: News Similarity Detection Dataset with Multiple Topics and Complexity Levels," *2024 Jt. Int. Conf. Comput. Linguist. Lang. Resour. Eval. Lr. 2024 - Main Conf. Proc.*, pp. 15181–15190, 2024.
[13] X. Wang, X. He, Y. Cao, M. Liu, and T. S. Chua, "KGAT: Knowledge graph attention network for recommendation," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 950–958, 2019, doi: 10.1145/3292500.3330989.
[14] P. Verma, "S2 Chunking: A Hybrid Framework for Document Segmentation Through Integrated Spatial and Semantic Analysis," pp. 1–10, 2025, [Online]. Available: http://arxiv.org/abs/2501.05485
[15] M. Angelos Goulis Supervisor and dr Clara Stegehuis, "Optimising node2vec in Dynamic Graphs Through Local Retraining," 2024.
[16] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, "Cosine similarity scoring without score normalization techniques," *Odyssey 2010 Speak. Lang. Recognit. Work.*, pp. 71–75, 2010.
[17] J. C. Nacher and T. Akutsu, "Analysis on critical nodes in controlling complex networks using dominating sets," *Proc. - 2013 Int. Conf. Signal-Image Technol. Internet-Based Syst. SITIS 2013*, pp. 649–654, 2013, doi: 10.1109/SITIS.2013.106.
[18] B. J. Goode and D. Datta, "A Geometric Approach to Predicting Bounds of Downstream Model Performance," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1596–1604, 2020, doi: 10.1145/3394486.3403210.
[19] Z. Huang, D. Liang, P. Xu, and B. Xiang, "Multiplicative Position-aware Transformer Models for Language Understanding," no. usually 512, 2021, [Online]. Available: http://arxiv.org/abs/2109.12788
[20] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is All You Need in Speech Separation," Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.13154

[21] T. Mikolov, W. T. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," *Proc. 2nd Work. Comput. Linguist. Lit. CLfL 2013 2013 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL-HLT 2013*, pp. 746–751, 2015.

[22] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019*, pp. 7370–7377, 2019, doi: 10.4000/books.aaccademia.4577.

[23] V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," *EMNLP 2020 - 2020 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 6769–6781, 2020, doi: 10.18653/v1/2020.emnlp-main.550.

[24] H. Chen, S. F. Sultan, Y. Tian, M. Chen, and S. Skiena, "Fast and accurate network embeddings via very sparse random projection," *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 399–408, 2019, doi: 10.1145/3357384.3357879.

[25] T. Chavan and S. Patil, "Named Entity Recognition (Ner) for News Articles," *Int. J. Adv. Res. Eng. Technol.*, vol. 2, no. 1, pp. 103–112, 2024, doi: 10.34218/ijaird.2.1.2024.10.