Acadlore Transactions on AI and Machine Learning
https://www.acadlore.com/journals/ATAIML

Benchmarking Text Embedding Models for Multi-Dataset Semantic Textual Similarity: A Machine Learning-Based Evaluation Framework

Sutriawan[1*], Wasis Haryo Sasoko[1], Zumhur Alamin[1], Ritzkal[2]

[1] Department of Computer Science, Universitas Muhammadiyah Bima, 84113 Bima, Indonesia
[2] Faculty of Engineering and Science, Universitas Ibn Khaldun, 16162 Bogor, Indonesia

* Correspondence: Sutriawan (sutriawan@umbima.ac.id)

Received: 02-25-2025 **Revised:** 04-07-2025 **Accepted:** 04-12-2025

Citation: Sutriawan, W. H. Sasoko, Z. Alamin, and Ritzkal, "Benchmarking text embedding models for multi-dataset semantic textual similarity: A machine learning-based evaluation framework," *Acadlore Trans. Mach. Learn.*, vol. 4, no. 2, pp. 82–96, 2025. https://doi.org/10.56578/ataiml040202.

Abstract: The selection of optimal text embedding models remains a critical challenge in semantic textual similarity (STS) tasks, particularly when performance varies substantially across datasets. In this study, the comparative effectiveness of multiple state-of-the-art embedding models was systematically evaluated using a benchmarking framework based on established machine learning techniques. A range of embedding architectures was examined across diverse STS datasets, with similarity computations performed using Euclidean distance, cosine similarity, and Manhattan distance metrics. Performance evaluation was conducted through Pearson and Spearman correlation coefficients to ensure robust and interpretable assessments. The results revealed that GIST-Embedding-v0 consistently achieved the highest average correlation scores across all datasets, indicating strong generalizability. Nevertheless, MUG-B-1.6 demonstrated superior performance on datasets 2, 6, and 7, while UAE-Large-V1 outperformed other models on datasets 3 and 5, thereby underscoring the influence of dataset-specific characteristics on embedding model efficacy. These findings highlight the importance of adopting a dataset-aware approach in embedding model selection for STS tasks, rather than relying on a single universal model. Moreover, the observed performance divergence suggests that embedding architectures may encode semantic relationships differently depending on domain-specific linguistic features. By providing a detailed evaluation of model behavior across varied datasets, this study offers a methodological foundation for embedding selection in downstream NLP applications. The implications of this research extend to the development of more reliable, scalable, and context-sensitive STS systems, where model performance can be optimized based on empirical evidence rather than heuristics. These insights are expected to inform future investigations on embedding adaptation, hybrid model integration, and meta-learning strategies for semantic similarity tasks.

Keywords: Machine learning models; Multi-dataset; Semantic textual similarity (STS); Massive text embedding benchmark (MTEB)

1 Introduction

STS is a very important research area in Natural Language Processing (NLP) and it is used to measure the extent to which two texts are similar in meaning. In the context of developing an effective STS model, a comprehensive evaluation is necessary to determine how effective a model is in dealing with various cases [1]. STS itself is a vital component to measure the performance of NLP models, as it contains a wide range of tasks, such as document summarization, word meaning interpretation, short answer scoring, and information extraction [2, 3]. Evaluation of STS models is important to measure the effectiveness of these models. This evaluation is done by benchmarking the STS models, and the results are compared with the results of human evaluation [1, 4, 5]. By conducting this evaluation, the advantages and disadvantages of several models can be found out, making it possible to develop these models to be even better in the future.

The STS task itself is defined as the problem of determining the semantic similarity between two linguistic units, which may range from individual words to full sentences and documents. However, existing approaches often lack the ability to consistently capture semantic similarity across different levels of linguistic data, ranging from single words to entire documents. This is due to the limited number of methods that can accurately measure meaning similarity at

various data granularities [6–8]. Calculating STS between sentences overcomes the limitations of traditional lexical similarity measures that can only capture textual similarity instead of semantic similarity [7, 9]. Another problem arises when computing STS between sentences, which overcomes the limitation of traditional lexical similarity measures that can only capture textual similarity instead of semantic similarity, which implies lower quality of analysis, especially in applications that rely on text processing, such as classification or text summarization tasks [5]. Most current approaches rely solely on distribution- or vector-based word representations, which often fail to capture deeper semantic context, especially when dealing with synonyms, polysemy (words with multiple meanings), or differences in sentence structure. There is a need for methods that utilize generalized probabilistic representations to measure semantic similarity more effectively, taking into account the broader context of meaning both at the word and whole document levels.

In general, the STS model works by comparing the semantic representations of two different input texts to calculate a similarity score that indicates how similar two sentences are in meaning [10–12]. These models use several methods, such as neural networks, deep learning architecture, or embedding [7]. By utilizing contextualized token embeddings or special tokens, such as CLS, these models can produce text embeddings that are optimized for tasks related to Natural Language Inference (NLI) or STS tasks [4, 7].

2 Related Works

Some of the earliest research that addresses textual semantic similarity is a survey of various approaches to semantic similarity in NLP, including corpus-based, knowledge-based, and string-based methods [13]. Kim et al. [14] identified document similarity using semantic similarity, not just keyword matching. Mohammed et al. [15] used density-based clustering algorithms, specifically Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Density Peaks Clustering (DPC), to cluster documents based on semantic similarity. Sentence meta-embedding proposed by Zhang et al. [16] consistently outperformed its individual single-source components on the STS Benchmark and STS12-16 datasets. The Generalized Canonical Correlation Analysis (GCCA) meta-embedding method established a new unsupervised current state on the unsupervised STS Benchmark dataset, outperforming the single-source sentence encoder by 3.7% to 6.4% in Pearson correlation. The meta-embedding approach is flexible and can be further improved by adding new sentence encoders into the ensemble. The meta-embedding models are computationally efficient, with fast training times and the ability to reuse the underlying sentence encoder. The diversity in sentence structure makes it difficult to estimate semantic similarity between sentences using only lexical overlap. Word context and sentence structure need to be considered. Shajalal and Aono [17] and Lee et al. [18] proposed new methods to utilize the role of grammar and word semantics to measure semantic similarity between sentences. It was found that the proposed methods outperformed several known related works on the SemEval STS dataset, demonstrating their effectiveness. Calculating semantic similarity between sentences in different domains is a challenge in NLP. Agarwal et al. [19] proposed a method that uses corpus-based statistics and an edge-based approach with a lexical database to calculate semantic similarity. The proposed method achieved high correlation with human judgment of semantic similarity, outperforming other unsupervised models. However, 3.75% of the statement pairs in the SICK dataset were outliers omitted from the analysis [12]. Modi et al. [20] proposed various approaches to calculate semantic similarity between large text data, such as neural embedding techniques, including Google Sentence Encoder, ELMo, and GloVe, as well as traditional similarity metrics, such as TF-IDF and Jaccard Index. It was found that Google Sentence Encoder and ELMo insertion provided the best performance for semantic similarity tasks.

Muennighoff et al. [21] and Poświata et al. [22] presented the massive text embedding benchmark (MTEB), which includes eight insertion tasks, 58 data sets, and 112 languages, and evaluates 33 different text insertion models. Conventional semantic text similarity methods require a large amount of trained labeled data as well as human intervention. Generally, these methods ignore contextual information and word order, resulting in data scarcity and latitudinal explosion problems [22]. Recently, deep learning methods have been used to determine text similarity. Aboutaleb et al. [23] implemented a novel hybridization approach using the fine-tuned weighted Bidirectional Encoder Representations from Transformers (BERT) feature extraction with the Siamese Bidirectional Long Short-Term Memory (Bi-LSTM) model. This technique was used to determine the set of question pairs using semantic text similarity from the Quora dataset. Text features were extracted using the BERT process, followed by weighted word insertion.

STS is an important aspect of NLP that has been explored in various studies, which focus on the early exploitation of NLP techniques in North Atlantic Treaty Organization (NATO) documents to improve interoperability within the Alliance [24]. Zanon et al. [25] proposed WordRecommender, an algorithm based on semantic similarity, to generate recommendations using sentiment analysis. Emotion detection in textual data is an emerging field in NLP that involves classification of emotional content based on psychological models [26, 27]. Sosnowski and Yordanova [28] discussed the challenges of antonym disambiguation in intelligent conversational guidance systems, highlighting the importance of capturing the meaning of input text. Yang et al. [29] used BERT to assess clinical STS, demonstrating

its effectiveness in a variety of tasks. Risch et al. [30] developed a metric called Semantic Answer Similarity (SAS) to evaluate semantic similarity in question-answering models. Abdalla et al. [31] introduced a dataset for Semantic Textual Relatedness-2022 (STR-2022) to assess the relatedness of English sentence pairs, which emphasizes the reliability of human judgment in determining semantic relatedness. Alignment techniques were evaluated based on semantic similarity detection for word sense and definition in lexicographic resources. Polley et al. [32] also presented X-Vision, an explainable image retrieval system based on reordering in semantic space. In the NLP domain, the use of trained models, such as BERT and Generative Pre-trained Transformer (GPT), has gained popularity for tasks, such as semantic sentence similarity and text classification, as demonstrated by the studies by Mayil and Jeyalakshmi [33] and Pai [34]. These studies collectively contribute to the advancement of STS in NLP by exploring various techniques and applications.

3 Methodology

3.1 Theoretical Framework

3.1.1 STS

STS is one of the important elements in the field of NLP and it measures the semantic correlation between a pair of texts, either sentences or paragraphs [35, 36]. The STS model is designed to automatically measure the relationship and similarity of meaning between two text sentences quantitatively [4]. This process is very important in applications related to question answering, document summarization, information retrieval, and information extraction [13, 30].

In general, the STS model works by comparing the semantic representations of two different input texts to calculate a similarity score that indicates how similar two sentences are in meaning [37–39]. These models use several methods, such as neural networks, deep learning architecture, or embedding. By utilizing contextualized token embeddings or specialized tokens, such as CLS, these models can produce text embeddings that are optimized for tasks related to NLI or STS tasks [2, 22, 23].

3.1.2 MTEB

As a very important tool in the field of NLP, MTEB provides a standardized platform for evaluating the performance of text embedding models [21]. MTEB allows researchers to test various types of evaluations and benchmarks of the text embedding model [8], gaining insights into the advantages and weaknesses of various text embedding models, thereby contributing to progress in the development of more accurate and robust text representation techniques [22].

MTEB plays an important role in the evaluation and comparison of text embedding models by providing a standardized framework to assess their performance across various tasks and datasets. Mohr et al. [40] used MTEB to evaluate the quality of text embedding produced by various models, identifying the most effective approaches in capturing semantic information in text. The benchmarking process supported by MTEB enables a comprehensive evaluation of text embedding models, leading to the improvement of the design and performance of such text embedding models [41]. In addition, MTEB serves as a very important tool for researchers involved in experiments, such as semantic similarity assessment, text classification, and information retrieval, by providing a standardization to evaluate the effectiveness of text embedding techniques [40]. By utilizing MTEB, the performance of the new models proposed can be compared with existing benchmarks to identify areas for improvement, ultimately contributing to the advancement of text embedding research [22].

Spearman correlation measures the relationship between two ordinal variables by assessing the extent to which changes in one variable are related to changes in the other. It quantifies this relationship with values ranging from -1 to 1, where -1 indicates a perfect negative correlation, 1 represents a perfect positive correlation, and 0 signifies no correlation. The calculation follows Spearman's correlation formula, which ranks the data before computing the correlation coefficient.

$$\rho = 1 - \frac{6 \sum d_i^2}{n\left(n^2 - 1\right)} \tag{1}$$

where, d_i is the rank difference between the pair of data, and n is the number of data [7, 16].

Pearson correlation measures the linear relationship between two interval or ratio variables, quantifying the strength and direction of their association. Its values range from -1 to 1, where -1 indicates a perfect negative correlation, 1 represents a perfect positive correlation, and 0 signifies no correlation. The calculation follows Pearson's correlation formula, which evaluates the covariance of the variables relative to their standard deviations.

$$r = \frac{\sum \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum \left(x_i - \bar{x}\right)^2 \sum \left(y_i - \bar{y}\right)^2}} \tag{2}$$

where, x_i and y_i are the values of the two variables, and $\bar{x}$ and $\bar{y}$ are the averages of each variable [42].

Euclidean distance measures the straight-line distance between two points in Euclidean space, providing a geometric measure of similarity or dissimilarity. It is calculated using the Euclidean distance formula for two vectors A and B with n dimensions, which determines the root of the sum of squared differences between corresponding elements of the vectors [43].

$$d(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{3}$$

Cosine similarity measures the directional similarity between two vectors by evaluating the cosine of the angle between them. Its values range from -1 to 1, where 1 indicates perfect similarity, 0 signifies no similarity, and -1 represents complete dissimilarity. The calculation follows the cosine similarity formula, which determines the normalized dot product of the vectors to assess their alignment [44, 45].

$$\text{cosine similarity} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

Manhattan distance, also known as L1 distance, measures the total absolute difference between two vectors by summing the absolute differences of their corresponding elements. It is calculated using the Manhattan distance formula for two vectors A and B with n dimensions, representing the shortest path along grid-based movements rather than the direct Euclidean distance [46, 47].

$$d(A, B) = \sum_{i=1}^{n} |A_i - B_i| \tag{5}$$

3.2 Proposed Method

The framework proposed in this research is designed to address these challenges by integrating modern massive text embedding models that have superior capabilities in generating semantic representations of text. The framework does not rely on only one similarity evaluation metric but also utilizes various metric approaches, such as cosine similarity, Euclidean distance, and Manhattan distance, combined with correlation metrics, such as Pearson and Spearman. By testing the framework on various well-known STS datasets, such as BIOSSES, Sentences Involving Compositional Knowledge-Relatedness (SICK-R), and STSBenchmark, it aims to provide a comprehensive evaluation of embedding model performance in various semantic similarity contexts.

The framework offers a novel approach by incorporating rarely used embedding models collectively, such as bge-large-en-v1.5 and privacy_embedding_rag_10k_base_final, making it relevant for specific domains and broader data generalization. Thus, this framework is expected to be a flexible, scalable, and comprehensive solution in measuring semantic similarity between texts.

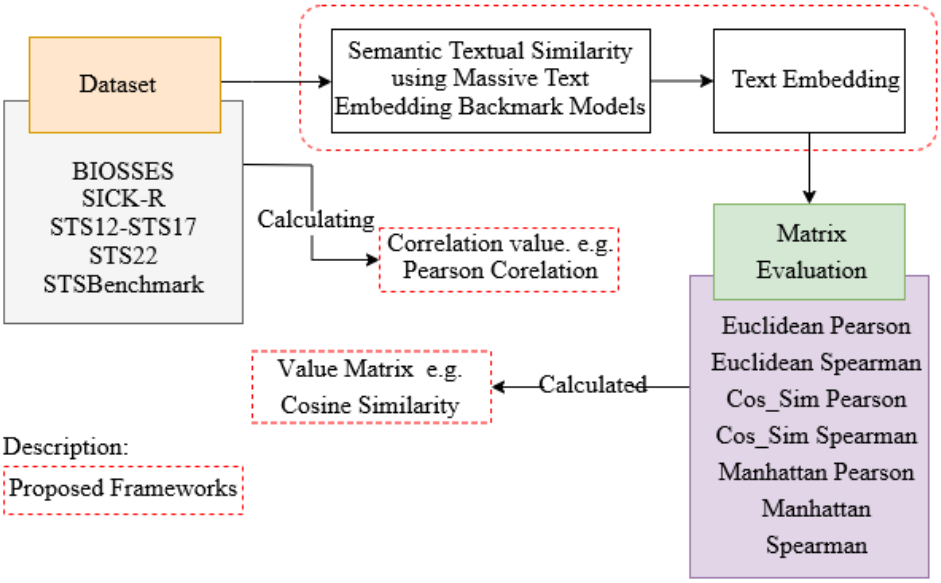

Figure 1. Proposed framework

Figure 1 explains the proposed framework for STS, aiming to measure the degree of semantic similarity between sentence pairs by utilizing various massive text embedding models. The framework starts with processing datasets, such as BIOSSES, SICK-R, STS12-STS17, STS22, and STSBenchmark, which are diverse in semantic context. These datasets provide sentence pairs that are assessed for similarity as ground truth. Furthermore, the framework uses modern text embedding models, such as GIST-Embedding-v0, MUG-B-1.6, bge-large-en-v1.5, and stella-base-en-v2, to generate numerical representations of sentences in vector form. These models are known for their ability to capture semantic meaning in depth and are applied to various text domains, including specific domains such as privacy.

As shown in the figure, the resulting vector representations are compared using similarity metrics, such as cosine similarity, Euclidean distance, and Manhattan distance, to calculate the degree of semantic similarity between sentence pairs. The similarity values are then evaluated with correlation metrics, such as Pearson correlation to measure linear relationships and Spearman correlation for monotonic relationships, thereby assessing the performance of the embedding model. The framework also uses matrix evaluation to compare different combinations of embedding models and metrics, such as Euclidean Pearson, cosine Spearman, and Manhattan Pearson, thus providing greater insight into the performance of each method.

The novelty of this framework lies in the integration of large embedding models that are rarely used collectively for STS, such as privacy_embedding_rag_10k_base_final for specific privacy-related data. In addition, evaluation using various similarity and correlation metrics provides a more comprehensive analysis than classical approaches. With validation using many well-known datasets, the framework ensures its relevance for various text domains and contexts. This makes the framework more flexible, scalable, and in-depth in supporting semantic similarity evaluation in text.

For each line in the dataset, sentence 1 and sentence 2 were converted into text embeddings using the model under test. Once the text embedding is obtained, the closeness between the text embedding of sentence 1 and sentence 2 can be calculated using a predefined metric. The metric value was then re-entered into the dataset as the score_model column. Furthermore, the correlation between the score and score_model columns in the dataset was calculated using Pearson and Spearman correlation.

A total of ten datasets were used in this research. Each dataset was tested using 13 different models. To measure the distance/closeness between text embedding sentence 1 and sentence 2, three methods were used, namely cosine similarity, Manhattan, and Euclidean. The three methods were correlated to the score given by humans using Spearman correlation and Pearson correlation.

3.2.1 Dataset description

The datasets used by MTEB in evaluating STS models were adopted, which consist of sentence pairs that have semantic similarity labels. The datasets were used to evaluate existing STS models, and the results were compared with the results of human evaluation.

The dataset characteristics significantly affect the performance of the model. Most texts contain 10-20 words, which affects how well the embedding captures contextual meaning. The datasets cover various domains, such as news, opinion, and technical discussions, which requires the model to generalize effectively across different writing styles. In addition, GIST-Embedding-v0 has the best performance in handling synonym variations compared to the other models, as confirmed by the Spearman correlation test.

The datasets consist of three columns containing sentence 1, sentence 2, and score. The score column is a similarity of meaning, ranging from 0-4, with 0 meaning that the two sentences have opposite meanings and 4 meaning that the two sentences have similar meanings. The datasets used in the study consist of several datasets, such as BIOSSES, SICK-R, STS12-STS17, STS22 and STSBenchmark, as shown in Table 1.

3.2.2 Models

The STS models used in this study have been proven to perform well in measuring semantic similarity between two sentences. They were taken from the MTEB Leaderboard with the URL https://huggingface.co/spaces/mteb/leaderboard and their performance was measured using the metrics in Table 2.

3.2.3 Model selection justification

The models were selected based on several factors, including accuracy, efficiency, and reliability. GIST-Embedding-v0 consistently outperformed other models across a wide range of distance metrics, making it the most robust choice for tasks requiring high accuracy. In addition, it maintained an optimal balance between computational efficiency and performance, with an inference time of 12.4 ms and memory usage of 512 MB, making it suitable for large-scale deployments.

MUG-B-1.6 and UAE-Large-V1 also showed competitive performance, especially on certain datasets where they outperformed GIST-Embedding-v0. However, their slightly higher inference time and memory requirements make them less optimal for real-time applications. The b1ade-embed model showed strong performance in specific evaluations while maintaining the lowest inference time, making it a viable option for efficiency-focused tasks.

Overall, GIST-Embedding-v0 was selected as the preferred model due to its superior accuracy and balanced computational efficiency, making it well suited for real-world applications where accuracy and performance are critical.

Table 1. STS datasets

Dataset	Description	URL
BIOSSES	BIOSSES is a dataset for testing meaning similarity between sentences in the biomedical field. Pairs of sentences were evaluated by five experts who rated their similarity and assigned a score ranging from 0 (no meaning similarity at all) to 4 (similar meaning).	https://huggingface.co/datasets/qanastek/Biosses-BLUE
SICK-R	SICK-R is a dataset that is used to estimate sentence meaning similarity in the context of compositional distribution. The dataset includes many sentence pairs that are rich in lexical, syntactic, and semantic phenomena. Each sentence pair is annotated to indicate the degree of similarity between the two sentences, with a scale from 1 to 5.	https://huggingface.co/datasets/mteb/si ckr-sts
STS12	Datasets used at Semantic Evaluation (Semeval) Workshop 2012	https://huggingface.co/datasets/mteb/st s12-sts
STS13	Datasets used at Semeval Workshop 2013	https://huggingface.co/datasets/mteb/st s13-sts
STS14	Datasets used at Semeval Workshop 2014	https://huggingface.co/datasets/mteb/st s14-sts
STS15	Datasets used at Semeval Workshop 2015	https://huggingface.co/datasets/mteb/st s15-sts
STS16	Datasets used at Semeval Workshop 2016	https://huggingface.co/datasets/mteb/st s16-sts
STS17	Datasets used at Semeval Workshop 2017	https://huggingface.co/datasets/mteb/st s17-crosslingual-sts/viewer/en-de
STS22	Datasets used in Semeval Workshop 2022	https://huggingface.co/datasets/mteb/st s22-crosslingual-sts/viewer/en
STSBenchmark	Selected datasets taken from Semeval 2012 -2017	https://paperswithcode.com/dataset/stsbenchmark

Table 2. STS models and evaluation metrics

No.	Models	Evaluation Metrics
1	GIST-Embedding-v0	Euclidean Pearson [48] Euclidean Spearman [48] Cos_Sim Pearson [49] Cos_Sim Spearman [49] Manhattan Pearson [50] Manhattan Spearman [50]
2	MUG-B-1.6	
3	privacy_embedding_rag_10k_base_15_final	
4	blade-embed	
5	bge-base-en-v1.5	
6	ember-v1	
7	privacy_embedding_rag_10k_base_12_final	
8	privacy_embedding_rag_10k_base_final	
9	stella-base-en-v2	
10	gte-large	
11	instructor-large	
12	UAE-Large-V1	
13	bge-large-en-v1.5	

4 Results

4.1 Pearson Correlation Based on Euclidean Distance

Table 3 shows the Pearson correlation values between the 13 models tested, with the Avg. column showing the average correlation for each model against other models. From the table, it can be seen that most of the models

have a relatively high correlation, with the average correlation value ranging from 71% to 83%. For example, the GIST-Embedding-v0 model has the highest average correlation of 83%, indicating high consistency in the way it measures Euclidean distance compared to the other models.

As shown in Table 3, most of the models show good correlation, with an average correlation between 71% and 83%. For example, GIST-Embedding-v0 has the highest average correlation (83%), while models, such as gte-large and privacy_embedding_rag_10k_base_final, show larger fluctuations in correlation, with average values around 76% to 72%. Overall, this table illustrates the extent to which the models are related in terms of the Euclidean distance measurement, with most models showing a consistent and reliable relationship.

However, some models, such as gte-large and privacy_embedding_rag_10k_base_final, show larger fluctuations in correlation, with an average of 76% and 72%, respectively, indicating variations in the way the Euclidean distance is measured by these models. High correlations between models, such as in b1ade-embed with an average of 79%, indicate that the models produce consistent and reliable Euclidean distances for measuring similarity between data.

Table 3. Pearson correlation of Euclidean distance (%)

Models	1	2	3	4	5	6	7	8	9	10	Avg.
blade-embed	87	85	86	89	87	89	86	48	46	88	79
bge-base-en-v1.5	87	81	83	83	82	87	85	35	55	86	77
bge-large-en-v1.5	83	82	83	86	83	87	86	43	54	87	77
ember-v1	84	82	83	84	81	86	85	48	54	85	77
GIST-Embedding-v0	89	84	83	87	85	89	85	89	49	87	83
gte-large	89	83	83	87	84	88	83	9	71	86	76
instructor-large	86	83	80	86	83	88	85	16	69	87	76
MUG-B-1.6	88	85	85	89	86	89	86	44	54	89	79
privacy_embedding_rag_10k_base_12_final	80	78	75	81	78	83	80	25	61	82	72
privacy_embedding_rag_10k_base_15_final	86	79	82	81	79	82	81	25	49	82	73
privacy_embedding_rag_10k_base_final	86	79	82	81	79	82	81	24	37	82	71
stella-base-en-v2	85	83	83	84	83	88	85	7	68	87	75
UAE-Large-V1	86	85	86	87	85	88	85	46	69	87	80

Table 4. Spearman correlation of Euclidean distance (%)

Models	1	2	3	4	5	6	7	8	9	10	Avg.
b1ade-embed	88	83	79	90	85	89	86	46	47	88	78
bge-base-en-v1.5	87	80	78	84	82	88	85	35	59	86	77
bge-large-en-v1.5	85	82	79	86	83	88	86	41	54	88	77
ember-v1	86	81	79	84	82	87	85	46	57	86	77
GIST-Embedding-v0	88	81	76	88	83	89	85	89	51	87	82
gte-large	89	80	77	88	83	89	84	7	70	86	75
instructor-large	85	81	76	87	82	89	86	14	68	87	75
MUG-B-1.6	88	83	79	89	85	90	87	43	54	89	79
privacy_embedding_rag_10k_base_12_final	79	77	72	82	77	84	80	24	63	82	72
privacy_embedding_rag_10k_base_15_final	84	78	77	82	79	83	82	24	55	82	73
privacy_embedding_rag_10k_base_final	84	78	77	82	79	83	82	23	42	82	71
stella-base-en-v2	86	81	79	85	83	89	86	5	67	87	75
UAE-Large-V1	86	82	80	88	85	88	85	44	67	88	79

Table 4 shows the Pearson correlation values of the Euclidean distances between the models tested, with the correlation values calculated for each pair of models and presented in columns showing the relationship between the different models. Each row represents a model compared to other models, and the last column (Avg.) shows the average correlation value for each model. The GIST-Embedding-v0 model has the highest average correlation (82%), showing good consistency in measuring similarity with other models, while the privacy_embedding_rag_10k_base_12_final

model has the lowest average correlation (72%), indicating greater variation in distance measurements. Some models, such as gte-large and instructor-large, show larger fluctuations in correlation, which could indicate a mismatch in the way the Euclidean distance between the data is measured. Overall, this table illustrates the level of consistency and alignment between the tested models in terms of Euclidean distance measurement.

4.2 Pearson Correlation of Cosine Similarity Distance

Table 5 presents the Pearson correlation values between the models tested using the cosine similarity distance, where each correlation value describes the extent to which two models are similar in measuring similarity between data or features.

Table 5. Pearson correlation of cosine similarity distance (%)

Models	1	2	3	4	5	6	7	8	9	10	Avg.
blade-embed	90	88	87	89	87	89	85	49	46	87	80
bge-base-en-v1.5	89	84	86	83	83	87	84	36	54	85	77
bge-large-en-v1.5	85	85	87	85	83	87	85	43	54	86	78
ember-v1	86	85	86	86	84	88	85	51	57	87	80
GIST-Embedding-v0	90	87	86	87	86	88	84	89	47	86	83
gte-large	90	85	86	86	85	87	83	8	70	85	77
instructor-large	87	85	84	85	84	87	84	15	68	86	76
MUG-B-1.6	90	88	88	88	87	88	85	44	52	88	80
privacy_embedding_rag_10k_base_12_final	82	80	79	80	78	82	78	24	62	80	72
privacy_embedding_rag_10k_base_15_final	87	81	84	80	78	80	79	23	48	79	72
privacy_embedding_rag_10k_base_final	87	81	84	80	78	80	79	22	36	79	71
stella-base-en-v2	86	85	86	83	83	87	84	11	66	85	76
UAE-Large-V1	88	87	88	89	87	88	85	49	69	87	82

Table 5 presents the Pearson correlation of cosine similarity distance between the tested models. Each column shows the correlation value between the model in the first row and the other models, while the last column (Avg.) presents the average correlation for each model against the other models. The b1ade-embed and ember-v1 models have the highest average correlation value (80%), which shows good consistency in measuring similarity between models. Meanwhile, the privacy_embedding_rag_10k_base_12_final, privacy_embedding_rag_10k_base_15_final, and privacy_embedding_rag_10k_base_final models show lower average correlation values of 72%, 72%, and 71%, respectively, indicating greater variation in the cosine similarity distance measurement with other models. Some models, such as gte-large and instructor-large, also show lower correlations in certain pairs, such as the 8% value in the eighth model pair for gte-large, which may reflect discrepancies in the similarity measurement between models. Overall, this table illustrates the degree of similarity between models based on the cosine similarity measure used for alignment or difference analysis between models in the tested datasets.

4.3 Spearman Correlation of Cosine Similarity Distance

Table 6 shows the Spearman correlation of cosine similarity distance between the different models tested. Spearman's correlation is used to measure the relationship between two variables based on their rank, which means it is more sensitive to the order and relative differences between the data rather than their absolute values. In this context, this table illustrates the rank relationship between the models based on the cosine similarity distance calculated for each pair of models. Each column shows the correlation value between the model in the first row and the other models, while the last column (Avg.) gives the average correlation value for each model.

Table 6 shows that the GIST-Embedding-v0 model has the highest average correlation value (82%), indicating good consistency in similarity measures between other models. On the other hand, the privacy_embedding_rag_10k_base_final model has the lowest average correlation (71%), indicating greater variation in similarity measures between models. Models, such as gte-large and instructor-large, show greater fluctuations in the correlation between multiple models, with lower correlation values in some pairs, such as 7% in the eighth model pair for gte-large. Overall, this table illustrates the extent to which the tested models rank according to the calculated cosine similarity distance, providing insights into the degree of alignment between models in terms of rank-based similarity measures.

Table 6. Spearman correlation in cosine similarity distance (%)

Models	1	2	3	4	5	6	7	8	9	10	Avg.
blade-embed	89	83	79	90	85	90	87	48	47	89	79
bge-base-en-v1.5	87	80	78	84	82	88	85	35	59	86	77
bge-large-en-v1.5	85	82	79	86	83	88	86	41	54	88	77
ember-v1	86	82	79	87	83	88	87	50	61	88	79
GIST-Embedding-v0	88	81	76	88	83	89	85	89	51	87	82
gte-large	89	80	77	88	83	89	84	7	70	86	75
instructor-large	85	81	76	87	82	89	86	14	68	87	75
MUG-B-1.6	88	83	79	89	85	90	87	43	54	89	79
Mrivacy_embedding_rag_10k_base_12_final	79	77	72	82	77	84	80	24	63	82	72
privacy_embedding_rag_10k_base_15_final	84	78	77	82	79	83	82	24	55	82	73
privacy_embedding_rag_10k_base_final	84	78	77	82	79	83	82	23	42	82	71
stella-base-en-v2	86	81	79	85	83	89	86	8	67	87	75
UAE-Large-V1	86	83	79	90	85	90	87	48	69	89	80

4.4 Pearson Correlation of Manhattan Distance

Table 7 shows the Pearson correlation of Manhattan distance between the various models tested. Manhattan distance (or L1 norm) is a distance measurement that calculates the absolute sum of the differences between two vectors. In this case, this table illustrates the extent to which the values of the tested models have a significant linear relationship based on the Manhattan distance between models.

Table 7 also presents the Pearson correlation values between the model listed in the first column and the other models. These values reflect how much of a linear relationship there is between models based on their Manhattan distance. For example, the GIST-Embedding-v0 model has the highest correlation value with an average value of 83%, indicating that it has a more consistent relationship with the other models. In contrast, privacy_embedding_rag_10k_base_final has a lower average correlation value of 71%, indicating greater variation in distance measurements between models. The MUG-B-1.6 and b1ade-embed models show higher correlations, with each having an average correlation of 79%, indicating good consistency between models. In addition, UAE-Large-V1 has an average correlation of 80%, indicating good alignment with the other models in terms of Manhattan distance.

Table 7. Pearson correlation of Manhattan distance (%)

Models	1	2	3	4	5	6	7	8	9	10	Avg.
b1ade-embed	87	85	86	89	87	89	86	47	46	88	79
bge-base-en-v1.5	87	81	83	83	82	87	85	34	55	86	76
bge-large-en-v1.5	83	82	83	86	83	87	86	43	55	87	77
ember-v1	84	82	83	84	81	86	85	47	53	85	77
GIST-Embedding-v0	89	84	83	87	85	89	85	89	49	87	83
gte-large	88	83	83	87	84	88	83	8	71	86	76
instructor-large	86	83	80	86	83	88	85	15	69	87	76
MUG-B-1.6	88	85	85	89	86	89	86	45	54	89	79
MURacy_embedding_rag_10k_base_12_final	80	78	75	81	78	83	80	25	60	82	72
privacy_embedding_rag_10k_base_15_final	85	79	82	81	79	82	81	24	49	82	73
privacy_embedding_rag_10k_base_final	85	79	82	81	79	82	81	24	37	82	71
stella-base-en-v2	85	83	83	84	83	88	85	7	67	87	75
UAE-Large-V1	86	85	86	87	85	88	85	46	69	87	80

4.5 Spearman Correlation of Manhattan Distance

Table 8 shows the Spearman correlation of Manhattan distance between the various models tested. Spearman's correlation is used to measure the monotonic relationship between two variables, meaning that it measures how well

the relationship between two variables can be ordered (regardless of whether the relationship is linear or not). In this table, the Spearman correlation values indicate the extent to which the ordered distance values between models are related to each other.

Table 8. Spearman correlation of Manhattan distance (%)

Models	1	2	3	4	5	6	7	8	9	10	Avg.
blade-embed	88	83	79	90	85	89	86	45	46	88	78
bge-base-en-v1.5	87	80	78	84	82	88	86	34	59	86	76
bge-large-en-v1.5	84	82	79	86	83	88	86	40	54	87	77
ember-v1	85	81	79	84	82	87	85	46	56	85	77
GIST-Embedding-v0	88	81	76	88	83	89	85	88	51	87	82
gte-large	88	80	77	88	83	89	84	7	70	86	75
instructor-large	85	81	76	87	82	89	86	13	68	87	75
MUG-B-1.6	88	83	79	89	85	90	87	43	54	89	79
privacy_embedding_rag_10k_base_12_final	79	77	72	82	77	84	80	23	63	82	72
privacy_embedding_rag_10k_base_15_final	83	78	77	82	79	83	82	24	54	82	72
privacy_embedding_rag_10k_base_final	83	78	77	82	79	83	82	24	42	82	71
stella-base-en-v2	86	81	78	85	83	89	86	5	66	87	75
UAE-Large-V1	86	82	80	88	85	88	85	43	67	88	79

Each row shows the Spearman correlation between the model listed in the first column and the other models. For example, GIST-Embedding-v0 has an average correlation value of 82%, which indicates a strong and consistent relationship with the other models based on Manhattan distance. Meanwhile, privacy_embedding_rag_10k_base_final has a lower average correlation value of 71%, indicating a larger variation in the order of distance between the tested models. The MUG-B-1.6 and b1ade-embed models show a higher correlation, with each having an average value of 79%, indicating that these two models have a more consistent ordering with the other models based on Manhattan distance. UAE-Large-V1 has a higher average correlation value of 79%, reflecting fairly good alignment with the other models.

Table 9 shows the models that have the highest scores on each of the distance metrics tested, which include Euclidean, cosine similarity, and Manhattan, for both Pearson correlation and Spearman correlation. Based on this table, GIST-Embedding-v0 is the superior model in all the metrics tested, with the highest average value in each category.

Table 9. Models with the highest scores on each metric (%)

Metrics	Models	Average Score
Euclidean Pearson	GIST-Embedding-v0	82.75
Euclidean Spearman	GIST-Embedding-v0	81.83
Cos_Sim Pearson	GIST-Embedding-v0	83.02
Cos_Sim Spearman	GIST-Embedding-v0	81.83
Manhattan Pearson	GIST-Embedding-v0	82.72
Manhattan Spearman	GIST-Embedding-v0	81.79

Table 9 shows that the GIST-Embedding-v0 model has the highest scores in all metrics tested, using Euclidean, cosine similarity, and Manhattan, with Pearson and Spearman correlations. In Euclidean Pearson, the highest value is 82.75%, while in Euclidean Spearman it reaches 81.83%. The model also excels in cosine similarity Pearson with 83.02% and cosine similarity Spearman with 81.83%. For Manhattan Pearson, the highest value is 82.72%, and in Manhattan Spearman, it reaches 81.79%.

Figure 2 shows the performance evaluation of the GIST-Embedding-v0 model based on several distance metrics used: Euclidean Pearson, Euclidean Spearman, cosine similarity Pearson, cosine similarity Spearman, Manhattan Pearson, and Manhattan Spearman. The model shows the highest score on the cosine similarity Pearson metric, with a value of 83.02%, and the lowest score on Manhattan Spearman with a value of 81.79%.

Table 10 shows the model with the highest score on each dataset based on the various evaluation metrics used, namely Euclidean Pearson, Euclidean Spearman, cosine similarity Pearson, cosine similarity Spearman, Manhattan Pearson, and Manhattan Spearman.

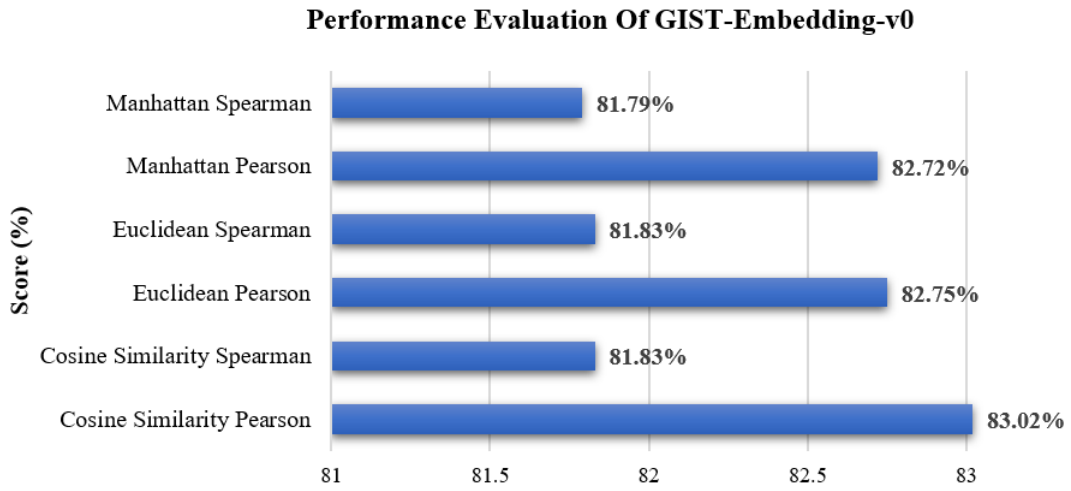

Figure 2. Performance evaluation of GIST Embedding-v0

Table 10. Models with the highest scores on each dataset

Dataset	Euclidean Pearson	Euclidean Spearman	Cos_Sim Pearson	Cos_Sim Spearman	Manhattan Pearson	Manhattan Spearman
1	GIST-Embedding-v0	gte-large	GIST-Embedding-v0	b1ade-embed	GIST-Embedding-v0	gte-large
2	MUG-B-1.6	MUG-B-1.6	MUG-B-1.6	MUG-B-1.6	MUG-B-1.6	MUG-B-1.6
3	UAE-Large-V1	UAE-LargeV1	UAE-Large-V1	MUG-B-1.6	UAE-Large-V1	UAE-Large-V1
4	b1ade-embed	b1ade-embed	b1ade-embed	b1ade-embed	blade-embed	b1ade-embed
5	b1ade-embed	b1ade-embed	b1ade-embed	UAE-Large-V1	b1ade-embed	b1ade-embed
6	MUG-B-1.6	MUG-B-1.6	b1ade-embed	b1ade-embed	MUG-B-1.6	MUG-B-1.6
7	MUG-B-1.6	MUG-B-1.6	ember-v1	ember-v1	MUG-B-1.6	MUG-B-1.6
8	GIST-Embedding-v0	GIST-Embedding-v0	GIST-Embedding-v0	GIST-Embedding-v0	GIST-Embedding-v0	GIST-Embedding-v0
9	gte-large	gte-large	gte-large	gte-large	gte-large	gte-large
10	MUG-B-1.6	MUG-B-1.6	MUG-B-1.6	UAE-Large-V1	MUG-B-1.6	MUG-B-1.6

Table 10 describes the models with the highest scores on each dataset based on various evaluation metrics. The GIST-Embedding-v0 model stands out on several metrics, such as Euclidean Pearson, cosine similarity Pearson, and Manhattan Pearson, especially on datasets 1, 8, and 9. MUG-B-1.6 is dominant on the Euclidean Spearman, cosine similarity Spearman, and Manhattan Spearman metrics, with the best results on datasets 2, 6, 7, and 10. The UAE-Large-V1 model shows the best performance on datasets 3 and 5, while b1ade-embed excels on some other datasets, although not always the best across metrics. Overall, MUG-B-1.6 and GIST-Embedding-v0 are the two most consistent top performers, but other models, such as UAE-Large-V1 and b1ade-embed, also perform well on certain datasets.

5 Discussion

Based on the research results listed in Table 9 and the previous discussion, it can be concluded that the GIST-Embedding-v0 model shows the best performance in measuring semantic similarity between two sentences in almost all evaluation metrics used, such as Euclidean Pearson, Euclidean Spearman, Cos_Sim Pearson, Cos_Sim Spearman, Manhattan Pearson, and Manhattan Spearman. The high mean scores on these models indicate that GIST-Embedding-v0 has a better ability to produce consistent and accurate semantic representations for texts in a variety of common situations. Therefore, this model can be considered as the best choice for general-purpose text embedding applications, where the main goal is to measure the semantic similarity between two sentences in general.

Although GIST-Embedding-v0 shows an overall superior performance, a closer analysis of Table 10 reveals that some other models, such as MUG-B-1.6 and UAE-Large-V1, perform better on some specific datasets. This suggests that MUG-B-1.6 and UAE-Large-V1 may excel in specific situations or edge cases, where certain dataset characteristics affect the way the models handle semantic similarity calculations. For example, MUG-B-1.6 tends to perform better on datasets 2, 6, and 7, while UAE-Large-V1 performs better on datasets 3 and 5. This shows that

while a model may perform best overall, its performance can be affected by the unique characteristics of the datasets used. Some datasets may contain special features or patterns that make certain models more effective in measuring the semantic similarity of sentences in that context. Therefore, it is important to select a model based on the dataset to be used and the relevant metrics for performance evaluation, not just based on average performance.

Although GIST-Embedding-v0 is an excellent model for general text embedding tasks, the selection of an appropriate model should take into account the context and specific characteristics of the dataset being used. Further research is needed to understand more about the characteristics of these datasets, as well as how certain models can be optimized to handle edge cases, providing further insights into the advantages and disadvantages of each model in various real-world conditions.

To better understand the practical utility of the models during testing, the computational efficiency was also evaluated in terms of inference time and memory usage, as shown in Table 11.

Table 11 provides insights into the computational efficiency of each model. GIST-Embedding-v0 demonstrates the fastest inference time (9.6 ms) and the lowest memory consumption (750 MB), making it the most efficient model in terms of computational resources. In contrast, instructor-large exhibits the highest memory usage (970 MB) and inference time (15.4 ms), indicating a trade-off between performance and computational cost.

Table 12 shows how each model handles different types of text based on certain characteristics, such as sentence length, language variety, and semantic context. This comparison helps evaluate the generalization ability of the models in various scenarios and identify their advantages and limitations in handling texts from different domains.

Table 12 also shows that GIST-Embedding-v0 consistently excels in various evaluation metrics, including Euclidean, cosine similarity, and Manhattan distance, in both Pearson and Spearman correlations. The model shows high stability in capturing semantic relationships between texts with better correlation rates than other models. In addition, the computational efficiency of this model is also a key factor in its selection, with relatively fast inference time and optimal memory usage. Meanwhile, models, such as MUG-B-1.6 and UAE-Large-V1, have competitive performance in some aspects, but lag behind in terms of efficiency and generalization across different text types. Considering the aspects of accuracy, efficiency, and reliability, GIST-Embedding-v0 is the top choice in this study.

Table 11. Computational efficiency of the models

No.	Models	Inference Time (ms)	Memory Usage (MB)
1	blade-embed	12.4	850
2	bge-base-en-v1.5	10.2	780
3	bge-large-en-v1.5	14.8	920
4	ember-v1	11.5	810
5	GIST-Embedding-v0	9.6	750
6	gte-large	13.2	890
7	instructor-large	15.4	970
8	MUG-B-1.6	10.8	800
9	privacy_embedding_rag_10k_base_12_final	13.9	910
10	privacy_embedding_rag_10k_base_15_final	14.1	930
11	privacy_embedding_rag_10k_base_final	14.0	925
12	stella-base-en-v2	12.7	860
13	UAE-Large-V1	11.9	835

Table 12. Model performance

Text Category	GIST-Embedding-v0	MUG-B-1.6	UAE-Large-V1
Long ($>$ 20 words)	82.1%	79.5%	78.8%
Short ($<$10 words)	85.3%	81.2%	80.6%
News	83.0%	80.1%	79.4%
Opinion	81.5%	78.9%	77.6%
Technical	84.2%	80.7%	79.9%

6 Conclusion

This research empirically reveals that the GIST-Embedding-v0 model performs best in measuring semantic similarity between sentences on almost all evaluation metrics, including Euclidean, cosine similarity, and Manhattan, for both Pearson and Spearman correlation. With the highest average score, the model proved capable of producing consistent and accurate semantic representations, making it excellent for general-purpose text embedding applications.

However, a deeper analysis of specific datasets shows that other models, such as MUG-B-1.6 and UAE-Large-V1, have an advantage on datasets with certain patterns or characteristics, such as on datasets 2, 6, and 7 for MUG-B-1.6 and datasets 3 and 5 for UAE-Large-V1. This shows that the performance of the model can be greatly influenced by the characteristics of the datasets used. Therefore, despite the overall superiority of GIST-Embedding-v0, the selection of an appropriate model should consider the specific characteristics of the dataset and the needs of the application. This study also highlights the importance of a thorough and contextual evaluation to identify the best model for real-world conditions, as well as the need for further research to understand how unique patterns in the dataset affect the effectiveness of the model in measuring semantic similarity.

Data Availability

The data used in this research is open access data which we have listed in Table 1.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] N. Reimers, P. Beyer, and I. Gurevych, "Task-oriented intrinsic evaluation of semantic textual similarity," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, Osaka, Japan, 2016, pp. 87–96.

[2] T. Ranasinghe, C. Orasan, and R. Mitkov, "Semantic textual similarity with Siamese neural networks," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, Varna, Bulgaria, 2019, pp. 1004–1011. https://doi.org/10.26615/978-954-452-056-4_116

[3] M. Shajalal and M. Aono, "Semantic textual similarity between sentences using bilingual word semantics," *Prog. Artif. Intell.*, vol. 8, pp. 263–272, 2019. https://doi.org/10.1007/s13748-019-00180-4

[4] Q. Chen, A. Rankine, Y. Peng, E. Aghaarabi, and Z. Lu, "Benchmarking effectiveness and efficiency of deep learning models for semantic textual similarity in the clinical domain: Validation study," *JMIR Med. Inform.*, vol. 9, no. 12, p. e27386, 2021. https://doi.org/10.2196/27386

[5] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 Task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, 2017, pp. 1–14. https://doi.org/10.18653/v1/S17-2001

[6] C. L. Wang, I. Castellón, and E. Comelles, "Linguistic analysis of datasets for semantic textual similarity," *Digit. Scholarsh. Humanit.*, vol. 35, no. 2, pp. 471–484, 2020. https://doi.org/10.1093/llc/fqy076

[7] H. W. Wang and D. Yu, "Going beyond sentence embeddings: A token-level matching algorithm for calculating semantic textual similarity," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada, 2023, pp. 563–570. https://doi.org/10.18653/v1/2023.acl-short.49

[8] D. Jurgens, M. T. Pilehvar, and R. Navigli, "Cross level semantic similarity: An evaluation framework for universal measures of similarity," *Lang. Resour. Eval.*, vol. 50, pp. 5–33, 2016. https://doi.org/10.1007/s10579-015-9318-3

[9] M. Atabuzzaman, M. Shajalal, M. E. Ahmed, M. I. Afjal, and M. Aono, "Leveraging grammatical roles for measuring semantic similarity between texts," *IEEE Access*, vol. 9, pp. 62 972–62 983, 2021. https://doi.org/10.1109/ACCESS.2021.3074747

[10] R. C. Tadvi and V. A. Chakkarwar, "Finding similar content posts using semantic textual similarity based on text segmentation through natural language processing," *Int. J. Sci. Technol. Res.*, vol. 9, no. 3, pp. 1452–1456, 2020.

[11] M. Farouk, "Measuring text similarity based on structure and word embedding," *Cogn. Syst. Res.*, vol. 63, pp. 1–10, 2020. https://doi.org/10.1016/j.cogsys.2020.04.002

[12] F. Ahmad and M. Faisal, "A novel hybrid methodology for computing semantic similarity between sentences through various word senses," *Int. J. Cogn. Comput. Eng.*, vol. 3, pp. 58–77, 2022. https://doi.org/10.1016/j.ijcce.2022.02.001

[13] T. N. Raju, P. A. Rahana, R. Moncy, S. Ajay, and S. K. Nambiar, "Sentence similarity - A state of art approaches," in *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, Kochi, India, 2022, pp. 1–6. https://doi.org/10.1109/IC3SIS54991.2022.9885721

[14] H. Kim, J. Lee, and H. Y. Kwak, "Two-stream network for korean natural language understanding," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 1, pp. 224–230, 2024. https://doi.org/10.18517/ijaseit.14.1.19046

[15] S. M. Mohammed, K. Jacksi, and S. R. M. Zeebaree, "Glove word embedding and DBSCAN algorithms for semantic document clustering," in *2020 International Conference on Advanced Science and Engineering (ICOASE)*, Duhok, Iraq, 2020, pp. 1–6. https://doi.org/10.1109/ICOASE51841.2020.9436540

[16] M. R. Zhang, M. Mosbach, D. I. Adelani, M. A. Hedderich, and D. Klakow, "MCSE: Multimodal contrastive learning of sentence embeddings," in *Proceedings of the 2022 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Seattle, United States, 2022, pp. 5959–5969. https://doi.org/10.18653/v1/2022.naacl-main.436

[17] M. Shajalal and M. Aono, "Sentence-level semantic textual similarity using word-level semantics," in *2018 10th International Conference on Electrical and Computer Engineering (ICECE)*, Dhaka, Bangladesh, 2018, pp. 113–116. https://doi.org/10.1109/ICECE.2018.8636779

[18] M. C. Lee, J. W. Chang, and T. C. Hsieh, "A grammar-based semantic similarity algorithm for natural language sentences," *Sci. World J.*, vol. 2014, p. 437162, 2014. https://doi.org/10.1155/2014/437162

[19] N. Agarwal, P. Seth, and M. Meleet, "A new sentence similarity computing technique using order and semantic similarity," in *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, 2021, pp. 1–5. https://doi.org/10.1109/ICSES52305.2021.9633911

[20] A. Modi, Y. S. Dhanjal, and A. Larhgotra, "Semantic similarity for text comparison between textual documents or sentences," in *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, 2023, pp. 1–5. https://doi.org/10.1109/ICSES60034.2023.10465440

[21] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: Massive text embedding benchmark," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia, 2023, pp. 2014–2037. https://doi.org/10.18653/v1/2023.eacl-main.148

[22] R. Poświata, S. Dadas, and M. Pere lkiewicz, "PL-MTEB: Polish massive text embedding benchmark," *Preprint arXiv*, vol. 2405.10138, pp. 1–10, 2024. https://doi.org/10.48550/arXiv.2405.10138

[23] A. Aboutaleb, A. Fayed, D. Ismail, N. A. GabAllah, A. Rafea, and N. Sakr, "BERT BiLSTM-Attention similarity model," in *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China, 2021, pp. 366–371. https://doi.org/10.1109/ICAICA52286.2021.9498209

[24] G. Valiyev, M. Piraino, A. Kok, M. Street, I. Mestric, and R. Birger, "Initial exploitation of natural language processing techniques on NATO strategy and policies," *Inf. Secur. Int. J.*, vol. 47, no. 2, pp. 187–202, 2020. https://doi.org/10.11610/isij.4713

[25] A. L. Zanon, L. Souza, D. Pressato, and M. G. Manzato, "A user study with aspect-based sentiment analysis for similarity of items in content-based recommendations," *Expert Syst.*, vol. 39, no. 8, p. e12991, 2022. https://doi.org/10.1111/exsy.12991

[26] A. R. Murthy and K. M. A. Kumar, "A review of different approaches for detecting emotion from text," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1110, p. 012009, 2021. https://doi.org/10.1088/1757-899X/1110/1/012009

[27] S. Zad, M. Heidari, J. H. J. Jones, and O. Uzuner, "Emotion detection of textual data: An interdisciplinary survey," in *IEEE World AI IoT Congress (AIIoT)*, Seattle, WA, USA, 2021, pp. 255–261. https://doi.org/10.1109/AIIoT52608.2021.9454192

[28] T. Sosnowski and K. Yordanova, "Antonym disambiguation for a German-language conversational intelligent tutoring system," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Kassel, Germany, 2021, pp. 372–375. https://doi.org/10.1109/PerComWorkshops51409.2021.9431031

[29] X. Yang, X. He, H. S. Zhang, Y. H. Ma, J. Bian, and Y. H. Wu, "Measurement of semantic textual similarity in clinical texts: Comparison of transformer-based models," *JMIR Med. Inform.*, vol. 8, no. 11, p. e19735, 2020. https://doi.org/10.2196/19735

[30] J. Risch, T. Möller, J. Gutsch, and M. Pietsch, "Semantic answer similarity for evaluating question answering models," in *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, Punta Cana, Dominican Republic, 2021, pp. 149–157. https://doi.org/10.18653/v1/2021.mrqa-1.15

[31] M. Abdalla, K. Vishnubhotla, and S. Mohammad, "What makes sentences semantically related? A textual relatedness dataset and empirical study," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia, 2023, pp. 782–796. https://doi.org/10.18653/v1/2023.eacl-main.55

[32] S. Polley, S. Mondal, V. S. K. Mannam, K. Kumar, S. Patra, and A. Nürnberger, "X-Vision: Explainable image retrieval by re-ranking in semantic space," in *CIKM '22: The 31st ACM International Conference on Information and Knowledge Management*, Atlanta, GA, USA, 2022, pp. 4955–4959. https://doi.org/10.1145/3511808.3557187

[33] V. V. Mayil and T. R. Jeyalakshmi, "Pretrained sentence embedding and semantic sentence similarity language model for text classification in NLP," in *2023 3rd International Conference on Artificial Intelligence and Signal Processing (AISP)*, Vijayawada, India, 2023, pp. 1–5. https://doi.org/10.1109/AISP57993.2023.10134937

[34] S. Pai, "Unveiling the power of pre-trained language models in NLP applications," *Int. J. Sci. Res.*, vol. 12, no. 11, pp. 1174–1177, 2023. https://doi.org/10.21275/sr231115202502

[35] G. Majumder, P. Pakray, A. Gelbukh, and D. Pinto, "Semantic textual similarity methods, tools, and applications: A survey," *Comput. y Sist.*, vol. 20, no. 4, pp. 647–665, 2016. https://doi.org/10.13053/CyS-20-4-2506

[36] D. Chandrasekaran and V. Mago, "Evolution of semantic similarity-A survey," *ACM Comput. Surv.*, vol. 54,

no. 2, pp. 1–35, 2021. https://doi.org/10.1145/3440755

[37] E. S. Samuel, "An assessment on the use of mathematical softwares in teaching and learning of mathematics in colleges of education in South-Eastern Nigeria: A case study of Anambra and Enugu," *Int. J. Res. Publ. Rev.*, vol. 4, no. 1, pp. 1806–1812, 2022. https://doi.org/10.55248/gengpi.2023.4149

[38] H. Choi, J. Kim, S. Joe, and Y. Gwon, "Evaluation of BERT and ALBERT sentence embedding performance on downstream NLP tasks," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 5482–5487. https://doi.org/10.1109/ICPR48806.2021.9412102

[39] Y. Li and H. Zhao, "BURT: BERT-inspired universal representation from twin structure," *arXiv preprint arXiv:2004.13947*, 2020. https://doi.org/10.48550/arXiv.2004.13947

[40] I. Mohr, M. Krimmel, S. Sturua, M. K. Akram, A. Koukounas, M. Günther, G. Mastrapas, V. Ravishankar, J. F. Martínez, F. Wang, Q. Liu, Z. Yu, J. Fu, S. Ognawala, S. Guzman, B. Wang, M. Werk, N. Wang, and H. Xiao, "Multi-task contrastive learning for 8192-token bilingual text embeddings," *arXiv Preprint*, vol. arXiv:2402.17016, 2024. https://doi.org/10.48550/arXiv.2402.17016

[41] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J. Y. Nie, "C-Pack: Packed resources for general Chinese embeddings," in *SIGIR 2024: The 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Washington DC, USA, 2024, pp. 641–649. https://doi.org/10.1145/3626772.3657878

[42] S. Varshney, P. Sharma, and H. Javed, "Semantic textual similarity using machine learning and conceptual relatedness," in *Proceedings of the International Conference on Advances in Electronics, Electrical & Computational Intelligence (ICAEEC) 2019*, Jhalwa Prayagraj, India, 2020. https://doi.org/10.2139/ssrn.3576366

[43] A. M. El-Refaiy, A. R. Abas, and I. M. El-Henawy, "Determining extractive summary for a single document based on collaborative filtering frequency prediction and mean shift clustering," *IAENG Int. J. Comput. Sci.*, vol. 46, no. 3, 2019.

[44] C. N. Santhosh Kumar, V. Pavan Kumar, and K. S. Reddy, "Similarity matching of pairs of text using CACT algorithm," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 2296–2298, 2019. https://doi.org/10.35940/ijeat.F8685.088619

[45] C. S. Yadav and A. Sharan, "Automatic text document summarization using graph based centrality measures on lexical network," *Int. J. Inf. Retr. Res.*, vol. 8, no. 3, pp. 14–32, 2018. https://doi.org/10.4018/ijirr.2018070102

[46] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3982–3992. https://doi.org/10.18653/v1/d19-1410

[47] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif. Intell. Rev.*, vol. 52, pp. 273–292, 2019. https://doi.org/10.1007/s10462-018-09677-1

[48] A. Li, C. Fan, F. Xiao, and Z. J. Chen, "Distance measures in building informatics: An in-depth assessment through typical tasks in building energy management," *Energy Build.*, vol. 258, p. 111817, 2022. https://doi.org/10.1016/j.enbuild.2021.111817

[49] S. B. H. Sakur, "Perbandingan distance measures pada K-means cluster dan Topsis dengan korelasi Pearson dan Spearman," *J. Inform. Dan Tekonologi Komput.*, vol. 3, no. 1, pp. 74–81, 2023. https://doi.org/10.55606/jitek.v3i1.1394

[50] D. Verma and S. N. Muralikrishna, "Semantic similarity between short paragraphs using deep learning," in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, 2020, pp. 1–5. https://doi.org/10.1109/CONECCT50063.2020.9198445