Electronics and Information Technologies, 2025, 30, 43–58
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.89 + 004.738.5

SEMANTIC SIMILARITY ANALYSIS USING TRANSFORMER-BASED SENTENCE EMBEDDINGS

Bohdan Pavlyshenko*, *Mykola Stasiuk
Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine

Pavlyshenko B. M., Stasiuk M. I. (2025). Semantic Similarity Analysis Using Transformer-Based Sentence Embeddings. *Electronics and Information Technologies*, 30, 43–58.
https://doi.org/10.30970/eli.30.4

ABSTRACT

Background. Transformer-based models have become central to natural language processing, demonstrating state-of-the-art performance in semantic similarity assessment, a task critical for various applications. These models capture detailed relationships between text, advancing the ability to gauge semantic relatedness.

Materials and Methods. The performance of sentence embedding models, including *all-mpnet-base-v2*, *all-MiniLM-L6-v2*, *paraphrase-multilingual-mpnet-base-v2*, *bge-base-en-v1.5*, *all-roberta-large-v1*, *all-distilroberta-v1*, *LaBSE*, *paraphrase-MiniLM-L3-v2*, *bge-large-en-v1.5*, was assessed across different dataset sizes with two datasets. The following preprocessing steps were applied to the datasets: lowercasing, removing stop words, cleaning from special symbols and numbers, and lemmatization. Cosine similarity scores with negative values, indicating semantic dissimilarity, were treated as equivalent to a human-annotated similarity score of 0, and non-negative cosine similarity values were scaled to the 0-5 range. Metrics such as R^2, MSE, RMSE, MAE, Spearman's Correlation Coefficient, and Kendall's Tau were used for evaluation.

Results and Discussion. Models' performance generally improves with increased data. Evaluation of sentence embedding models revealed performance variations. *all-roberta-large-v1* showed strong accuracy with high R^2values and low errors. *BAAI/bge-large-en-v1.5* excelled in capturing semantic relationships, demonstrating high Spearman's and Kendall's Tau coefficients. *all-MiniLM-L6-v2* demonstrated the fastest embedding generation. *BAAI/bge-base-en-v1.5* presented the lowest accuracy. Processing times generally increase with data size.

Conclusion. This study highlights a trade-off between accuracy and efficiency in sentence embedding. Model selection depends on balancing these factors to align with specific application needs. In cases when requiring high accuracy should favor *all-roberta-large-v1*, while those prioritizing speed would benefit from *all-MiniLM-L6-v2*. *BAAI/bge-large-en-v1.5* is most suitable for tasks demanding semantic understanding of text details.

Keywords: semantic similarity, sentence embeddings, transformers.

INTRODUCTION

As digital information grows at an accelerating pace, the ability to understand the meaning behind text has become fundamental. From search engines to nuanced interactions with virtual assistants, the need for machines to comprehend semantic content is becoming increasingly important. However, while computers excel at processing raw textual data, catching the hidden nuances of natural language remains a complex challenge. The core of this challenge lies in making it possible for machines to move beyond simple word matching and to understand the underlying meaning of text instead.

Researchers have developed text embeddings [1], a powerful technique that transforms words and sentences into dense numerical vectors to address this challenge. Unlike simple word representations, these vectors capture the semantic relationships between text elements, allowing computers to understand context and meaning. This process has evolved significantly, moving from earlier methods like Word2Vec [2] and GloVe [3], which generated static word embeddings, to more advanced contextual embeddings such as those produced by transformers [4] and transformer-based models like BERT [5] and its variants. These advancements have paved the way for the development of various application areas for text embeddings, ranging from fundamental tasks like text classification [6] to more sophisticated techniques such as data augmentation using large language models [7]. The benefit of converting text to vectors is that mathematical operations such as calculating distance can be performed on them, thus enabling computers to quantify semantic relationships [8, 9].

A crucial application of text embeddings is the semantic similarity measurement, quantifying the degree to which two pieces of text share a related meaning. This capability enables a wide range of Natural Language Processing tasks. For instance, accurate semantic similarity in information retrieval allows search engines to identify documents conceptually relevant to a user's query, even if they do not contain the exact words [10]. Similarly, question-answering systems enable matching questions to appropriate answers by understanding their underlying meaning [11]. Other applications include plagiarism detection, where subtle similarities in phrasing can be identified, and document clustering, where related documents are grouped based on their semantic content [12]. The performance of all these applications is directly tied to the quality and accuracy of the text embeddings used to calculate semantic similarity.

Building on these advancements, the broader landscape of Natural Language Processing has recently been transformed by powerful Large Language Models (LLMs) such as Llama 3/4, Grok, Qwen, or Mistral. While these general-purpose LLMs demonstrate impressive capabilities in broad generative and complex reasoning tasks, it is crucial to differentiate them from specialized sentence embedding models. The models primarily evaluated in this study are specifically engineered and optimized for generating high-quality, dense vector representations of text, making them inherently more efficient and effective for the semantic similarity tasks under investigation.

While numerous sentences embedding models have been introduced recently, direct comparisons of their performance and efficiency are not always prevalent. This article therefore, undertook a comparative investigation of various sentence embedding models to address the challenge of accurately quantifying semantic similarity between texts. Specifically, the output of these models, when paired with cosine similarity, was analyzed to estimate the degree of semantic relatedness between different text snippets. Furthermore, to provide a practical perspective on the efficiency of these models, the time taken by each model to generate embeddings and subsequently compute the similarity scores was also measured and compared. This dual approach offered insights into the effectiveness and computational cost of employing different sentence embedding techniques for text similarity tasks.

Obtained insights are not merely theoretical; they directly inform the practical selection of models for diverse real-world applications. For instance, in scenarios demanding high-throughput processing with limited computational resources, such as real-time conversational AI, where models provide rapid, contextually-aware responses by semantically matching user input to relevant intents, or large-scale document indexing, where documents are transformed into searchable embeddings to enable finding content by meaning and efficient content organization, understanding a model's efficiency becomes paramount. Conversely, for applications where strong predictive capability and nuanced semantic understanding are critical, such as precise information retrieval for identifying the most relevant passages to a user's query or complex question-answering systems that must accurately extract answers from vast knowledge bases, model effectiveness takes precedence. The quality of these generated embeddings is therefore paramount for the success of such systems. This study systematically quantifies these crucial trade-offs, providing an empirical basis for developers and researchers to choose the most suitable sentence embedding model among those utilized, taking into account considerations such as efficiency and predictive performance.

MATERIALS AND METHODS

In experiments, two datasets, publicly available on the Hugging Face Portal and designed explicitly for semantic similarity tasks, were utilized: sts-companion [13] and mteb-stsbenchmark-sts [14]. These datasets contain pairs of sentences along with a corresponding human-annotated semantic similarity score. Both datasets use similarity scores, which lie on a scale of 0 to 5 and represent varying degrees of semantic relatedness.

One of the datasets consists of 5289 records, and the other 8628. Both datasets were tested with initial subsets of 100, 500, 1000, 2000, 4000, and 5000 records. The larger dataset was also tested with 8000, and the full dataset size, while the smaller dataset was tested with the full dataset size. This approach allowed us to assess how model effectiveness varies with an increasing number of records.

To evaluate the effectiveness of different sentence embedding models for text similarity comparison, we employed the following models, all of which are available on the Hugging Face portal:

- sentence-transformers/all-MiniLM-L6-v2: this model is a compact and efficient Transformer-based architecture. It is designed to produce effective sentence embeddings with relatively small model size, making it suitable for applications where computational resources are limited or speed is crucial;
- *sentence-transformers/paraphrase-multilingual-mpnet-base-v2*: this model leverages the MPNet architecture [15] and has been trained on a large multilingual corpus. It is fine-tuned explicitly for paraphrase identification and semantic similarity across various languages, making it a strong choice for cross-lingual text understanding tasks;
- *BAAI/bge-base-en-v1.5*: BAAI developed this model based on the transformer architecture and trained it on a substantial English language dataset. It is recognized for achieving high performance on various text embedding benchmarks, demonstrating its ability to capture nuanced semantic relationships in English text;
- *sentence-transformers/all-roberta-large-v1*: this model utilizes the RoBERTa architecture [16], a powerful transformer variant known for its strong language understanding capabilities. Being a "large" model, it has a greater capacity to learn complex semantic representations, often leading to high accuracy in sentence embedding tasks;
- *sentence-transformers/all-distilroberta-v1*: this model is a distilled version of RoBERTa, meaning it retains much of the performance of its larger counterpart while having a smaller size and faster inference speed. It offers a good balance between accuracy and efficiency for generating sentence embeddings;
- *sentence-transformers/LaBSE*: standing for Language-Agnostic BERT Sentence Embeddings [17], LaBSE is explicitly designed to produce embeddings comparable across many different languages. This makes it highly effective for multilingual semantic similarity tasks and cross-lingual information retrieval;
- *sentence-transformers/paraphrase-MiniLM-L3-v2*: like the *all-MiniLM-L6-v2* model, this is another efficient transformer-based model focused on generating sentence embeddings. The "paraphrase" in its name indicates that it has been fine-tuned explicitly on paraphrase datasets, making it well-suited for tasks involving semantic equivalence;

- *sentence-transformers/all-mpnet-base-v2:* is based on the MPNet architecture, which is known for its effectiveness in capturing semantic relationships between sentences. It is designed to produce high-quality sentence embeddings and demonstrates a good balance of accuracy and computational efficiency. This model is suitable for a wide range of semantic similarity tasks.
- *BAAI/bge-large-en-v1.5*: is the larger variant of BAAI's *bge-base-en-v1.5* model for English. With more parameters, it aims to present state-of-the-art performance on English language text embedding tasks. It offers higher accuracy than its base counterpart at the cost of increased computational resources.

For this study, all models were evaluated exclusively on English language texts, irrespective of their inherent multilingual capabilities.

Fig. 1 illustrates the characteristics of the datasets used in this study. The top row displays the distribution of similarity scores for the mteb/stsbenchmark-sts (left) and sts-companion (right) datasets. The x-axis represents the similarity score, while the y-axis represents the frequency of occurrence. The distributions reveal differences in the two datasets' range and concentration of similarity scores. While both datasets have many texts with similar lengths, their score distributions differ. The mteb/stsbenchmark-sts dataset has a more uniform distribution of scores, whereas the sts-companion dataset shows a higher concentration of high-value scores.

Figure 1's middle and bottom rows show the distributions of sentence lengths for the first and second sentences in the comparison pairs, respectively. The x-axis represents the sentence length (number of tokens), and the y-axis represents the frequency of occurrence. These plots provide information about the variability in sentence lengths within each dataset. Notably, sts-companion contains a broader range of sentence lengths than mteb/stsbenchmark-sts, potentially impacting sentence embedding models' computational demands and performance.

To quantitatively evaluate the performance of each sentence embedding model in the semantic similarity task, we employed four commonly used regression metrics:

- **Mean Squared Error (MSE):** calculates the average of the squared differences between the predicted and the actual similarity scores. Lower MSE values indicate better model performance.
- **Root Mean Squared Error (RMSE)**: the RMSE is the square root of the MSE. It measures the average magnitude of the errors in the same units as the target variable, making it more interpretable than MSE.
- **Mean Absolute Error (MAE)**: calculates the average absolute differences between the predicted and actual similarity scores. Like RMSE, MAE is expressed in the units of the target variable and provides a robust measure of error, less sensitive to outliers than MSE or RMSE.
- $\boldsymbol{R^2}$**:** the coefficient of determination measures the proportion of the variance in the human-annotated similarity scores explained by the predicted scores. R^2 values range from 0 to 1, with higher values indicating a better fit of the model to the data.
- **Spearman's Rank Correlation Coefficient**: assesses the strength and direction of the monotonic association between the predicted and actual similarity values. It quantifies how well the ranking of predicted ratings aligns with human-annotated scores. Spearman's correlation spans from –1 to 1, where figures approaching 1 denote a strong positive monotonic link, figures nearing –1 suggest a strong negative monotonic link, and values close to 0 imply a weak or nonexistent monotonic link.
- **Kendall's Tau**: measures the ordinal association between the predicted and actual similarity scores. It evaluates the degree to which the predicted ordering agrees with the human-annotated ordering. Kendall's Tau figures range from –1 to 1, with results approaching 1 indicating strong agreement in the order, results nearing –1 indicating strong disagreement, and results near zero indicating weak or no agreement.

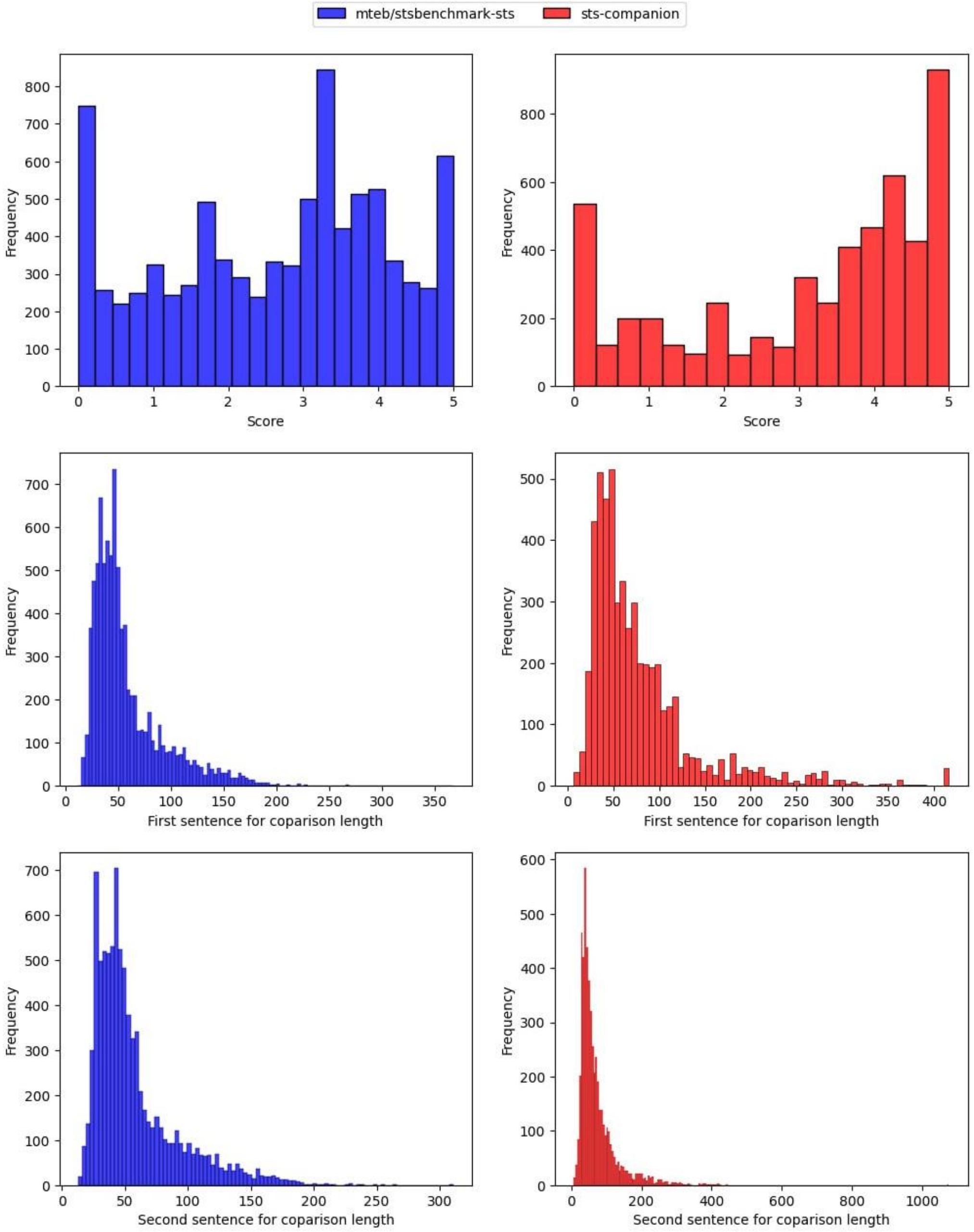

Fig. 1. Distribution of similarity scores and sentence lengths in the mteb/stsbenchmark-sts and sts-companion datasets.

These metrics collectively provide a comprehensive assessment of the sentence embedding models' performance in predicting semantic similarity. Additionally, we measured the time each model took to embed the text and calculated the average of these times to compare the computational efficiency of the models.

The experimental procedure involved the following steps.

1. The sentence pairs from each dataset were preprocessed to ensure consistency. The preprocessing included: lowercasing, removing stop words, cleaning from special symbols and numbers, and lemmatization;

2. Each sentence in the paired data was encoded into a vector embedding using the selected sentence embedding models.
3. The semantic similarity between the resulting vector embeddings for each sentence pair was calculated using cosine similarity. It is important to note that cosine similarity yields a score between –1 and 1. For this study, negative cosine similarity values, indicating semantic dissimilarity, were treated as equivalent to a human-annotated similarity score of 0, and only non-negative cosine similarity values were scaled to the 0-5 range.
4. The scaled cosine similarity scores were compared against the ground truth similarity scores to assess the effectiveness of each sentence embedding model in capturing semantic relationships.

RESULTS AND DISCUSSION

In this study, we made a simplifying assumption regarding the interpretation of negative cosine similarity values. Specifically, any sentence pair with negative cosine similarity calculated from the model-generated embeddings was assigned a scaled similarity score corresponding to the lowest possible human-annotated score - 0. This decision was made to align the model output with the lower bound of the human scoring scale and to focus the analysis primarily on the model's ability to capture positive semantic similarity. This approach is often adopted as a pragmatic choice when mapping model output to human-annotated scales that typically do not assign granular scores below semantic unrelatedness, i.e., below zero similarity. It simplifies the evaluation framework by focusing on the strength of positive semantic relationships, which is often the primary objective in tasks such as information retrieval and question answering.

However, it is important to acknowledge that this approach has limitations. We lose some of the information provided by the model's embedding space by treating all negative cosine similarity values as equivalent. A highly negative cosine similarity suggests a more substantial degree of semantic dissimilarity, potentially even indicating some semantic opposition, than a slightly negative value. Our simplification collapses these distinctions. Furthermore, this approach might mask subtle differences in how the models represent semantic dissimilarity. Some models might consistently produce more negative values for unrelated or contrasting sentences, and this behavior is not captured in our scaled scores. Future work could explore alternative scaling methods, such as mapping the full range of cosine similarity to the 0 - 5 scale, to better preserve the information contained in negative similarity values and provide a more detailed comparison of model performance.

Table 1 presents the times taken by each model to generate text embeddings for text from the mteb-stsbenchmark-sts dataset. The *paraphrase-MiniLM-L3-v2* model demonstrates the most efficient performance, with embedding times around 2 ms for smaller subsets, reaching only 3.7 ms for the whole dataset. The next best performance was exhibited by *all-MiniLM-L6-v2* and *all-distilroberta-v1* models; their times are approximately 3 ms for smaller subsets. However, *all-distilroberta-v1* climbs to 18.6 ms for the complete dataset, while *all-MiniLM-L6-v2* attains 5.6 ms for the most extensive dataset. Similar performance for smaller subsets was shown by *bge-base-en-v1.5* and *paraphrase-multilingual-mpnet-base-v2* – around 4–5 ms; for the 8628-record dataset, times grow to 34.8 ms and 38.8 ms, respectively. The *all-mpnet-base-v2* model has times of roughly 5–7 ms for small subsets and 35.1 ms for the largest. The *LaBSE* model's time expands to 36.3 ms for the whole dataset, compared to about 5–6 ms for smaller ones. In contrast, *all-roberta-large-v1* and *bge-large-en-v1.5* are the slowest; for the 8628-record dataset, times surge to 142.7 ms and 162.6 ms, respectively. While most models maintain relatively stable embedding times for smaller subsets (100 to 2000 records*), all-roberta-large-v1* and *bge-large-en-v1.5* display a substantial jump for the whole dataset.

Table 1. **Times taken by every model to create a vector representation of strings from the *mteb-stsbenchmark-sts* dataset.**

	Time, ms							
Records number **Model**	100	500	1000	2000	4000	5000	8000	8628
all-mpnet-base-v2	7.3	5.3	5.3	5.3	5.3	5.2	33.9	35.1
all-MiniLM-L6-v2	3.1	3.0	3.0	3.0	3.1	3.1	4.5	5.6
paraphrase-multilingual-mpnet-base-v2	4.7	4.9	4.9	4.8	5.0	4.7	38.5	38.8
bge-base-en-v1.5	4.7	4.7	4.7	4.6	4.9	4.6	33.4	34.8
all-roberta-large-v1	9.7	8.4	8.2	8.3	56.2	8.4	141.4	142.7
all-distilroberta-v1	3.3	3.0	3.1	3.0	17.5	3.0	18.3	18.6
LaBSE	6.2	4.8	4.7	4.8	32.3	4.6	35.6	36.3
paraphrase-MiniLM-L3-v2	2.9	2.0	2.1	2.2	3.2	2.2	3.0	3.7
bge-large-en-v1.5	10.1	8.3	8.3	8.4	123.0	8.3	132.8	162.6

For most models, the time to embed a given number of records remains relatively stable across the smaller subset sizes (100 to 2000 records). This suggests that the embedding process scales consistently for these models within this range. However, the time taken for some models noticeably expands when processing the larger subsets, particularly the full dataset. For example, the embedding time for *all-roberta-large-v1* surges to 142.7 ms for the 8628-record dataset, compared to approximately 8.3 ms for smaller subsets. Similarly, *bge-large-en-v1.5* also exhibits a substantial rise, reaching 162.6 ms for the most extensive dataset.

In contrast, models like *all-MiniLM-L6-v2* and *paraphrase-MiniLM-L3-v2* show the smallest increase in processing time when moving to the full dataset, suggesting more consistent scaling performance. *LaBSE* also jumps to 36.3 ms for the most extensive dataset. This suggests that the computational cost of processing the full dataset introduces additional overhead for some models but not all, highlighting differences in how efficiently these models handle larger inputs.

Table 2 shows the performance of each model in terms of embedding time. The *paraphrase-MiniLM-L3-v2* model demonstrates the most efficient performance, consistently achieving the fastest embedding times across all dataset sizes, with times around 2 ms. In contrast, the *bge-large-en-v1.5* model is the slowest, with embedding times growing with the dataset size and reaching 187.2 ms for the largest dataset of 5289 records. This significant rise suggests that its computational demands scale more substantially with larger inputs. Several other models also show relatively fast performance across the smaller dataset sizes, with times generally below 6 ms, including *all-mpnet-base-v2*, *paraphrase-multilingual-mpnet-base-v2*, and *bge-base-en-v1.5*. However, their embedding times display a noticeable climb for the largest dataset. Specifically, *all-roberta-large-v1* and *LaBSE* attain a moderate expansion, reaching 50.1 ms and 45.7 ms, respectively, for the 5289-record dataset. *all-distilroberta-v1* also shows a jump at the largest dataset size, reaching 22.6 ms.

Most models exhibit relatively consistent embedding times across varying dataset sizes. This suggests that the processing demands remain stable for these models. However, some models experience increased processing duration when handling the largest dataset of 5289 records. For instance, the embedding time for *all-roberta-large-v1*

Table 2. **Times taken by every model to create a vector representation of strings from the *sts-companion* dataset.**

	Time, ms						
Records number / **Model**	100	500	1000	2000	4000	5000	5289
all-mpnet-base-v2	6.8	5.3	5.4	5.2	5.2	5.3	5.3
all-MiniLM-L6-v2	3.2	3.1	3.0	3.1	3.1	3.1	3.1
paraphrase-multilingual-mpnet-base-v2	4.8	5.0	4.9	5.0	4.8	4.9	4.8
bge-base-en-v1.5	5.7	4.8	4.7	4.8	4.7	4.8	4.7
all-roberta-large-v1	10.5	8.9	8.8	8.7	8.6	8.6	50.1
all-distilroberta-v1	3.0	3.1	3.0	3.1	3.1	3.0	22.6
LaBSE	6.0	5.0	4.8	4.7	4.8	4.8	45.7
paraphrase-MiniLM-L3-v2	2.4	2.1	2.2	2.2	2.2	2.2	3.0
bge-large-en-v1.5	10.5	9.1	8.8	8.6	8.5	8.7	187.2

rises to 50.1 ms for the 5289-record dataset, compared to approximately 8.6 ms for smaller subsets. Similarly, *LaBSE* increases to 45.7 ms for the most extensive dataset. *bge-large-en-v1.5* displays the most significant increase, reaching 187.2 ms for the largest dataset. Conversely, *all-MiniLM-L6-v2* and *paraphrase-MiniLM-L3-v2* demonstrate the most stable processing times across different dataset sizes, indicating more efficient scaling. *all-distilroberta-v1* also shows a slight increase, reaching 22.6 ms for the largest dataset. These results indicate that the computational load associated with processing the complete dataset introduces additional overhead for specific models, revealing differences in how effectively they manage larger inputs.

Table 2 shows the performance of each model in terms of embedding time. The *paraphrase-MiniLM-L3-v2* model demonstrates the most efficient performance, consistently achieving the fastest embedding times across all dataset sizes, with times around 2 ms. In contrast, the *bge-large-en-v1.5* model is the slowest, with embedding times growing with the dataset size and reaching 187.2 ms for the largest dataset of 5289 records. This significant rise suggests that its computational demands scale more substantially with larger inputs. Several other models also show relatively fast performance across the smaller dataset sizes, with times generally below 6 ms, including *all-mpnet-base-v2*, *paraphrase-multilingual-mpnet-base-v2*, and *bge-base-en-v1.5*. However, their embedding times display a noticeable climb for the largest dataset. Specifically, *all-roberta-large-v1* and *LaBSE* attain a moderate expansion, reaching 50.1 ms and 45.7 ms, respectively, for the 5289-record dataset. *all-distilroberta-v1* also shows a jump at the largest dataset size, reaching 22.6 ms.

Most models exhibit relatively consistent embedding times across varying dataset sizes. This suggests that the processing demand remains stable for these models. However, some models experience increased processing duration when handling the largest dataset of 5289 records. For instance, the embedding time for *all-roberta-large-v1*

rises to 50.1 ms for the 5289-record dataset, compared to approximately 8.6 ms for smaller subsets. Similarly, *LaBSE* increases to 45.7 ms for the most extensive dataset. *bge-large-en-v1.5* displays the most significant increase, reaching 187.2 ms for the largest dataset. Conversely, *all-MiniLM-L6-v2* and *paraphrase-MiniLM-L3-v2* demonstrate the most stable processing times across different dataset sizes, indicating more efficient scaling. *all-*

distilroberta-v1 also shows a slight increase, reaching 22.6 ms for the largest dataset. These results indicate that the computational load associated with processing the complete dataset introduces additional overhead for specific models, revealing differences in how effectively they manage larger inputs.

A notable similarity across both tables is the consistent performance of specific models, particularly *paraphrase-MiniLM-L3-v2* and *all-MiniLM-L6-v2*. These models yield rapid embedding times for smaller datasets, indicating their efficiency in processing text when the input size is relatively constrained. Furthermore, the models that present the most substantial increase in processing time when handling the larger input are also consistent across both tables, namely *bge-large-en-v1.5*. Regardless of the specific dataset, nearly all models demonstrate an increase in processing time when handling the most significant input, suggesting that scaling to accommodate large datasets introduces computational challenges for these embedding methods.

However, key differences in model's performance were observed between the two tables. The magnitude of the increase in processing time for the most extensive datasets varies considerably. For example, *bge-large-en-v1.5* shows a substantial jump in processing time in both cases, highlighting its less efficient scaling. In contrast, *all-MiniLM-L6-v2* and *paraphrase-MiniLM-L3-v2* maintain more stable and lower processing times even for the largest datasets, indicating better scalability. Additionally, the relative ranking of some models shifts between the tables, with models like *all-roberta-large-v1* and *LaBSE* indicating a more pronounced performance decrease for the largest dataset in Table 1 compared to Table 2, suggesting that dataset characteristics can influence model efficiency.

To investigate a potential correlation with observed increases in processing times near the end of the dataset, Figure 2 reveals sentence length distributions between the two datasets. The "mteb/stsbenchmark-sts" dataset exhibits a broader range of sentence lengths for both the first and second sentences. The distribution is concentrated towards shorter lengths, with two peaks: in the middle and another near the end, indicating the presence of some longer sentences. In contrast, the "sts-companion" dataset shows a strong bias towards very short lengths for the first sentences, while the second sentences, though also predominantly short, display greater variability and include some exceptionally long sentences.

Despite these differences in sentence length distribution, it is unlikely that sentence length alone is the primary driver of the observed increase in processing time for the larger datasets in Tables 1 and 2. While transformer-based models can show increased computational cost with more extended sequences, the relatively modest differences in average sentence length, especially when considering the substantial jump in processing time for some models with the complete datasets, suggest that factors such as batching overhead, memory limitations, or model architecture are more influential in the observed performance scaling.

Figure 3 presents a comparative analysis of various sentence embedding models across different evaluation metrics as the number of training records increases. Generally, for smaller subsets of the mteb-stsbenchmark-sts dataset, the performance of most models appears relatively stable across all evaluation metrics. However, as the dataset size increases towards the full 8628 records, the distinctions between model performances become more apparent. For the error rates (MSE, RMSE, and MAE), some models reveal a degradation in effectiveness, showing increased error with larger datasets. Interestingly, a few models seem to maintain or even slightly improve their error metrics as the data size grows. In contrast to the varied trends in error metrics, Spearman's correlation coefficient and Kendall's Tau tend to follow a somewhat similar overall trend across most models: performance generally increases and then plateaus or slightly decreases as the dataset size expands, suggesting a common pattern in how well these models capture semantic ranking with more data.

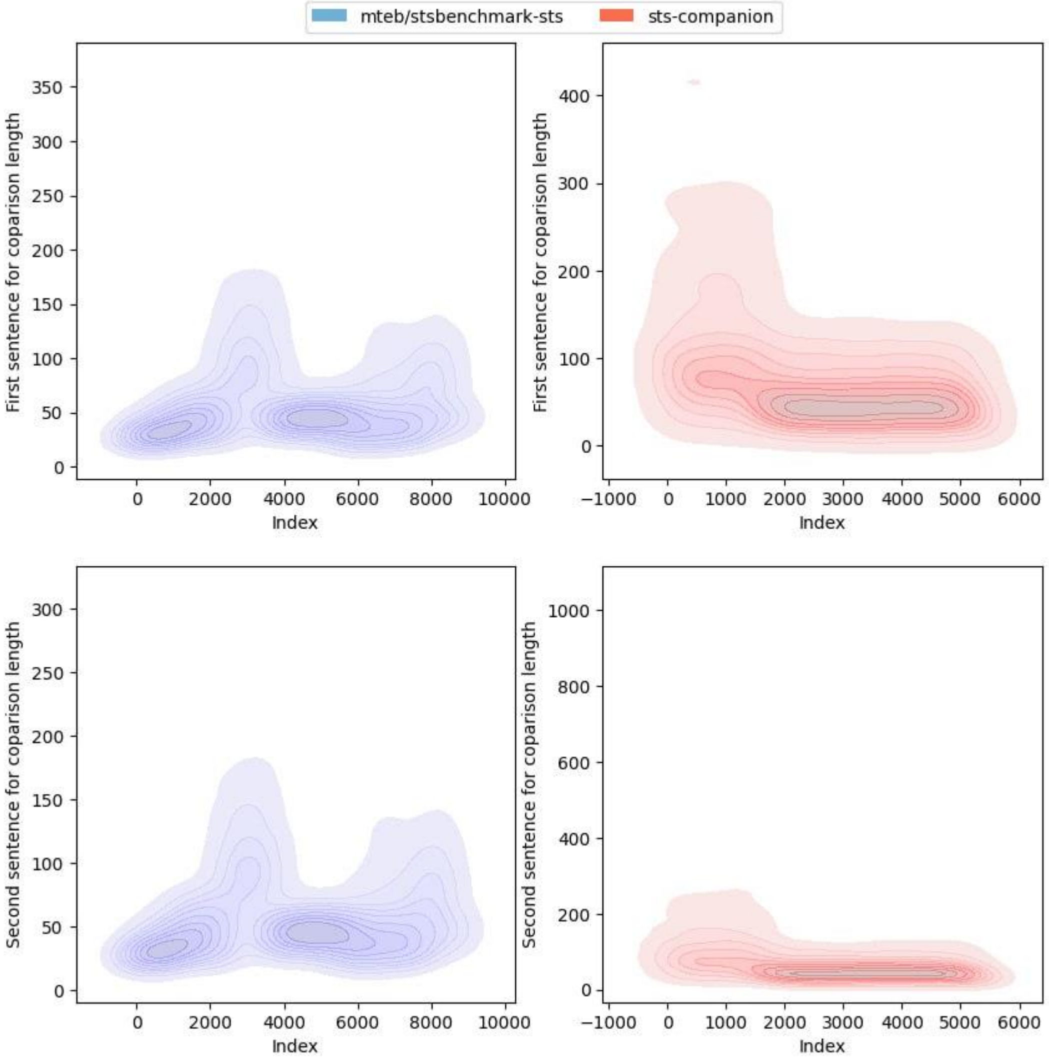

Fig. 2. Sentence lengths in mteb-stsbenchmark-sts and sts-companion.

The *sentence-transformers/all-roberta-large-v1* model appears to be the top performer, consistently demonstrating high R^2 values and low error rates, indicating a strong ability to model the underlying data. Following closely is *BAAI/bge-base-en-v1.5*, which also shows competitive results in terms of R^2 and error metrics, and excels in capturing data relationships as evidenced by its high Spearman's and Kendall's Tau correlation coefficients. The *all-mpnet-base-v2* model generally occupies the third position, showing stable and competitive performance across most metrics. In contrast, the *sentence-transformers/paraphrase-MiniLM-L3-v2* model consistently lags behind the others, showing the lowest R^2 values and the highest error rates throughout the evaluated data range, suggesting it may be less effective in capturing the semantic nuances within this dataset.

Based on the evaluation metrics presented in Figure 3, the *sentence-transformers-/LaBSE* model consistently demonstrates the weakest performance across a range of dataset sizes. Evidenced by the highest MSE and RMSE, *sentence-transformers/LaBSE* attains the largest prediction errors compared to other models. Furthermore, its MAE also remains notably high. In terms of correlation, the *sentence-transformers/LaBSE* model

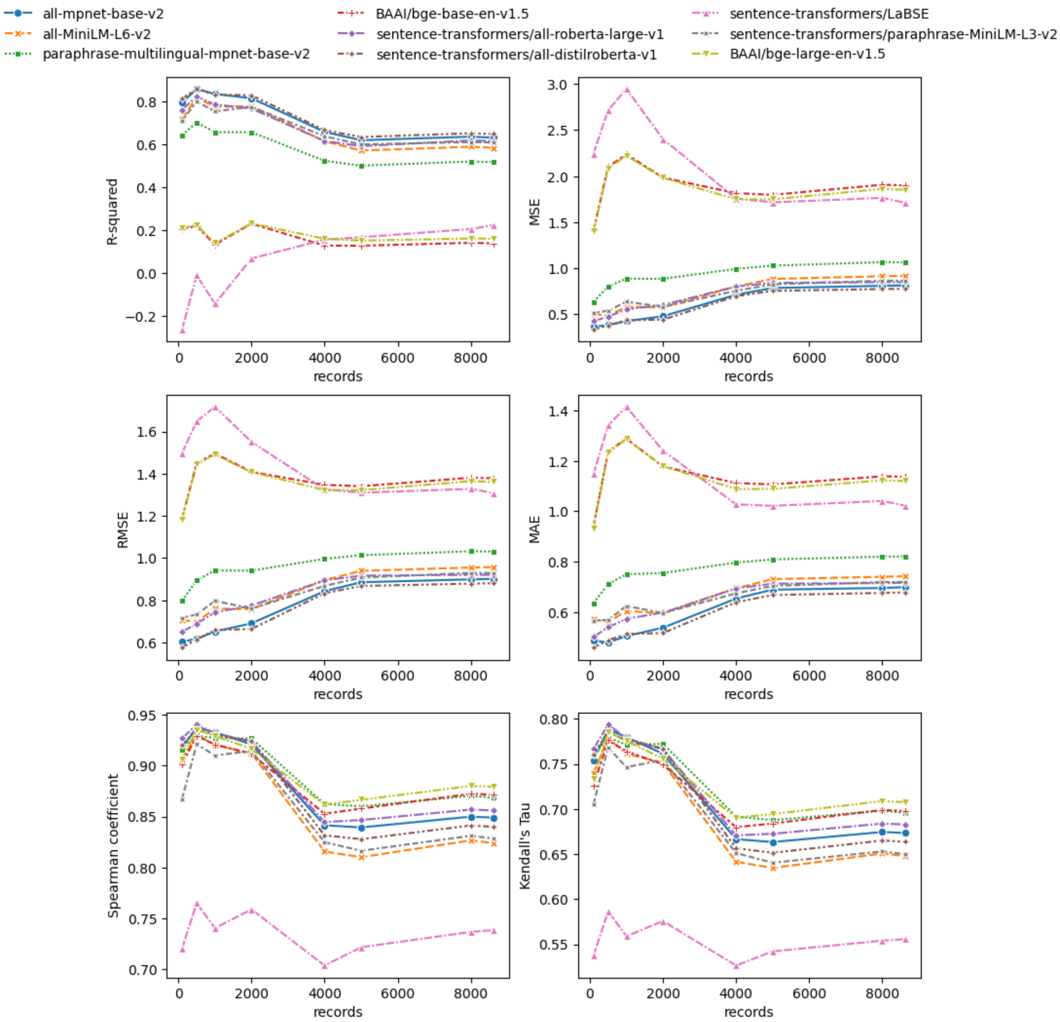

Fig. 3. Performance metrics for sentence embedding models across varying mteb-stsbenchmark-sts dataset sizes.

yields the lowest Spearman’s and Kendall's Tau coefficients, indicating a poor ability to capture both the monotonic and ordinal relationships between predicted and actual semantic similarities. While its R^2 values show some improvement with larger datasets, its overall performance across the majority of the key evaluation metrics positions *sentence-transformers/LaBSE* as the least effective model among those compared in this analysis.

Considering the time taken for vector representation in Table 1 alongside the performance metrics in Figure 3 reveals interesting trade-offs. The *all-MiniLM-L6-v2* model stands out as remarkably fast, consistently achieving the lowest processing times across all dataset sizes. However, its performance, while generally competitive, doesn't place it at the top tier, often showing lower R^2 and higher error rates compared to models like *all-roberta-large-v1* and *bge-base-en-v1.5*. Conversely, models like *sentence-transformers/ all-roberta-large-v1* and *BAAI/bge-large-en-v1.5*, which demonstrated strong performance in the evaluation metrics, tend to be significantly slower, particularly as the dataset size increases, as seen in the higher time values in Table 1. The *sentence-transformers/ paraphrase-MiniLM-L3-v2*, while being relatively fast, also shows the weakest performance

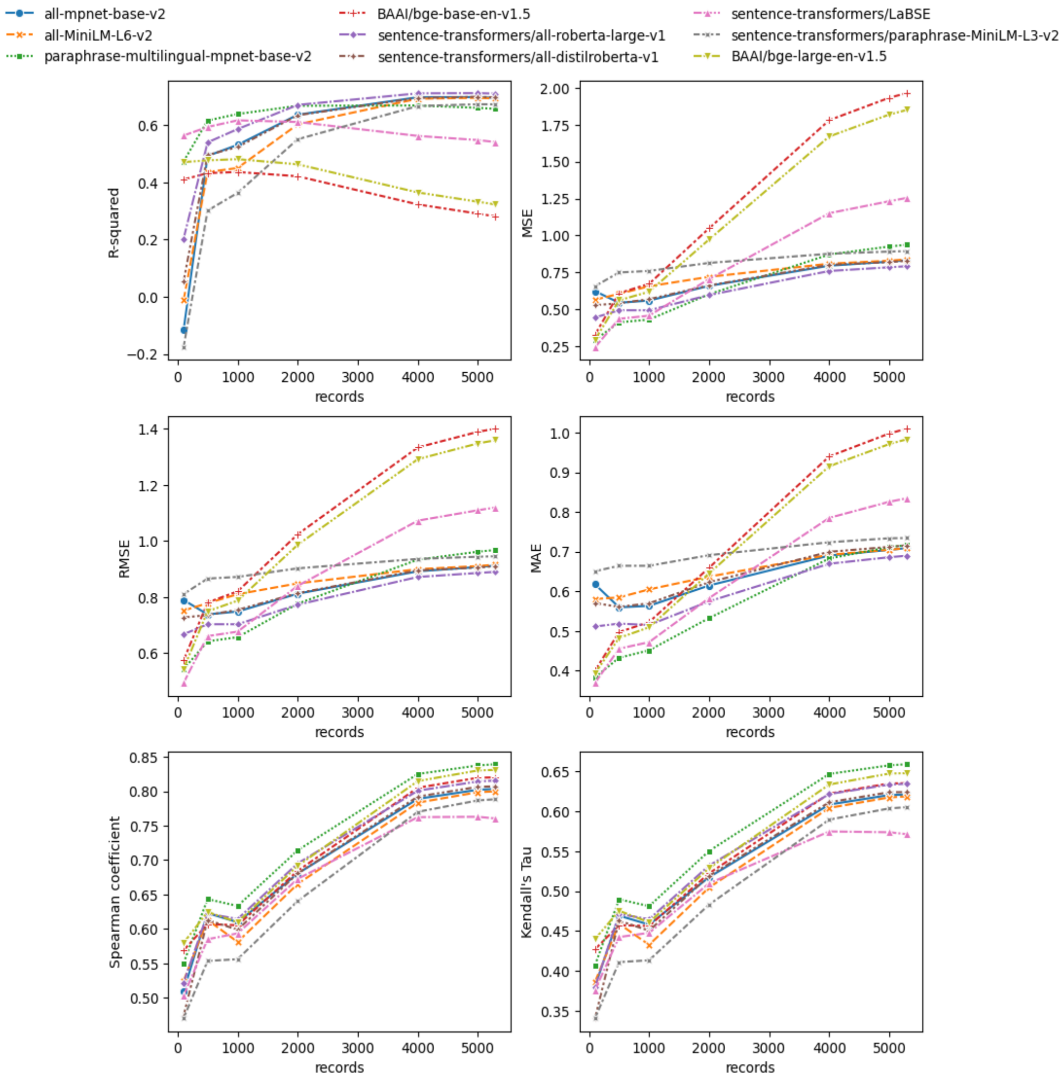

Fig. 4. Performance metrics for sentence embedding models across varying sts-companion dataset sizes.

across the metrics, making it a less appealing choice when considering both efficiency and accuracy.

Figure 4 presents a performance comparison of various sentence embedding models across several evaluation metrics as the number of training records increases. Generally, the models show the most significant performance changes within the initial range of data points, tending to plateau beyond approximately 1000-2000 records. Across the metrics, distinct performance characteristics emerge for different models. The *sentence-transformers/all-roberta-large-v1* model stands out for attaining high R^2 values and low error rates, indicating a strong ability to fit the data and produce accurate predictions. However, when considering the correlation metrics, *paraphrase-mulrilingual-mpnet-base-v2* demonstrates superior performance, consistently presenting the highest Spearman's and Kendall's Tau coefficients, suggesting a stronger ability to capture the monotonic and ordinal relationships within the data. In contrast, the *BAAI/bge-base-en-v1.5* model consistently exhibits the lowest R^2 values and the highest error rates, along with generally lower correlation coefficients, indicating it is the least effective in capturing the underlying relationships within this dataset based on these metrics. Therefore, the choice of the "best" model depends on the specific priorities of the application, whether it emphasizes overall prediction accuracy or the preservation of data rankings.

Based on the evaluation metrics in Figure 4, the top-performing models present a trade-off. For achieving high prediction accuracy, as indicated by R^2 and error metrics, *sentence-transformers/all-roberta-large-v1* appears to be the most effective. However, for tasks where the preservation of data relationships is paramount, the *paraphrase-mulrilingual-mpnet-base-v2* model demonstrates superior performance in terms of Spearman’s and Kendall's Tau coefficients. Consequently, the choice between these two as the "best" depends on the specific application priorities. The *all-mpnet-base-v2* model generally is the third-best option, showing a consistent and competitive performance across all the evaluated metrics.

Focusing solely on Figure 4, the *BAAI/bge-base-en-v1.5* model demonstrates the weakest performance in terms of prediction accuracy, showing the lowest R^2 values and the highest error rates. When considering the correlation metrics, the situation changes with increasing data. Initially, the *sentence-transformers/paraphrase-MiniLM-L3-v2* model generally shows lower Spearman’s and Kendall's Tau values. However, as the number of records increases, the performance of *LaBSE* deteriorates in terms of both Spearman’s and Kendall's Tau, eventually becoming the lowest among all models in the higher record range. This suggests that while *BAAI/bge-base-en-v1.5* struggles most with prediction accuracy, *LaBSE's* ability to preserve the relative ordering of the data appears to degrade with larger datasets within this evaluation, ultimately making it the least effective in this aspect at higher record counts.

Combining the performance results from Figure 4 with the timing data from Table 2 reveals a trade-off between model accuracy and computational efficiency. The *all-MiniLM-L6-v2* model, which exhibits the fastest processing times, demonstrates reasonably competitive performance, though it doesn't excel in any single metric. In contrast, while *sentence-transformers/all-roberta-large-v1* shows strong accuracy, it is considerably slower, as indicated by its higher processing times. The *BAAI/bge-large-en-v1.5* model, which performs well in capturing data relationships, also becomes notably more time-consuming as the dataset size increases.

Comparing the performance of the sentence embedding models across the two datasets (mteb-stsbenchmark-sts in Figure 3 and sts-companion in Figure 4) reveals both consistent patterns and interesting divergences. Notably, *sentence-transformers/all-roberta-large-v1* consistently demonstrates strong performance in terms of prediction accuracy on both datasets, achieving high R^2 values and low error rates. Similarly, *all-mpnet-base-v2* maintains a relatively stable and competitive performance across the board in both evaluations. This suggests that these models possess a degree of robustness and generalizability across different semantic textual similarity tasks. Furthermore, the tendency for performance gains to diminish with increasing data size (beyond a certain point) is observed in the results for both datasets, indicating a potential saturation point for these model architectures on this type of task.

However, the identification of the weakest performing model differs between the two evaluations. In Figure 3, *sentence-transformers/paraphrase-MiniLM-L3-v2* generally shows the poorest performance across most metrics. In contrast, Figure 4 highlights *BAAI/bge-base-en-v1.5* as having the lowest accuracy and reveals a degradation in *LaBSE's* correlation performance with larger data. This discrepancy suggests that the specific characteristics of each dataset can influence the relative strengths and weaknesses of the models. The sts-companion dataset, for instance, might present different semantic nuances or complexities that impact the models in varying ways compared to the *mteb-stsbenchmark-sts* dataset. Therefore, while some models have consistent behavior, the optimal choice depends on the utilized dataset.

The comparative performance and efficiency data detailed in this section can be used as a practical guide for model selection, enabling practitioners to weigh crucial trade-offs between predictive effectiveness, nuanced semantic understanding, and computational cost. This allows for the identification of the most suitable model tailored to the specific task requirements and available computational resources within an English-language context.

CONCLUSION

This study provided a comparative analysis of several sentence embedding models, evaluating their performance on semantic similarity tasks across two datasets and assessing their embedding generation efficiency. The findings highlight significant trade-offs between model accuracy and speed, alongside model-specific strengths and weaknesses in capturing semantic relationships. The results offer insights for selecting the most suitable model based on application-specific priorities and resource constraints.

In terms of performance metrics, *all-roberta-large-v1* consistently demonstrated strong results in prediction accuracy, achieving high R^2 values and low error rates across both datasets. For capturing semantic relationships and preserving the relative ordering of sentences, *BAAI/bge-large-en-v1.5* generally excelled, particularly evident in its high Spearman’s and Kendall's Tau coefficients. *all-mpnet-base-v2* consistently offered a balanced and competitive performance across all metrics.

Conversely, the models yielding the weakest performance varied depending on the metric and the dataset. *BAAI/bge-base-en-v1.5* tended to show the lowest R^2 and highest error rates on the sts-companion dataset. Regarding the preservation of nuanced semantic understanding, *sentence-transformers/paraphrase-MiniLM-L3-v2* generally showed lower correlation values, while *LaBSE's* performance in Spearman’s and Kendall's Tau deteriorated significantly with larger datasets in the "sts-companion" evaluation.

Considering the interplay between performance and computational efficiency, *sentence-transformers/all-MiniLM-L6-v2* model stands out as the most time-efficient, consistently generating embeddings rapidly across varying dataset sizes, albeit with a slight compromise in top-tier semantic representation accuracy. In contrast, models demonstrating high accuracy, such as *all-roberta-large-v1*, often incurred a higher computational cost. Notably, *sentence-transformers/LaBSE* generally exhibited weaker performance metrics coupled with moderate to slower processing speeds, making it a less favorable choice when considering both factors.

Moving forward, we plan to apply the top-performing models from this evaluation, such as *all-mpnet-base-v2*, *sentence-transformers/all-distilroberta-v1*, and *sentence-transformers/all-MiniLM-L6-v2*, to custom datasets. This will allow us to assess their generalization capabilities and fine-tune them for specific applications. Future work may also explore techniques to optimize these models for efficiency and performance on domain-specific data.

AUTHOR CONTRIBUTIONS

Conceptualization, [*M.S.*]; methodology, [*M.S.*]; validation, [*M.S.*]; writing – original draft preparation, [*M.S.*]; writing – review and editing, [*B.P.*, *M.S.*]; supervision, [*B.P.*].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137-1155. https://doi.org/10.1162/153244303322533223

[2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. https://doi.org/10.48550/arXiv.1301.3781

[3] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543). https://doi.org/10.3115/v1/D14-1162

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30. https://doi.org/10.48550/arXiv.1706.03762

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186). https://doi.org/10.18653/v1/N19-1423

[6] Pavlyshenko, B., & Stasiuk, M. (2024). Data augmentation in text classification with multiple categories. Electronics and information technologies, (25). http://dx.doi.org/10.30970/eli.25.6

[7] Pavlyshenko, B., & Stasiuk, M. (2025). Using Large Language Models for Data Augmentation in Text Classification Models. International Journal of Computing, 24(1), 148-154. https://doi.org/10.47839/ijc.24.1.3886

[8] Pavlyshenko, B. (2014). Clustering of authors' texts of english fiction in the vector space of semantic fields. Cybernetics and Information Technologies, 14(3), 25-36. https://doi.org/10.2478/cait-2014-0030

[9] Pavlyshenko, B. (2013). Classification analysis of authorship fiction texts in the space of semantic fields. Journal of Quantitative Linguistics, 20(3), 218–226. https://doi.org/10.1080/09296174.2013.799914

[10] Han, M., Zhang, X., Yuan, X., Jiang, J., Yun, W., & Gao, C. (2021). A survey on the techniques, applications, and performance of short text semantic similarity. Concurrency and Computation: Practice and Experience, 33(5), e5971. https://doi.org/10.1002/cpe.5971

[11] Risch, J., Möller, T., Gutsch, J., & Pietsch, M. (2021). Semantic answer similarity for evaluating question answering models. https://doi.org/10.48550/arXiv.2108.06130

[12] Vrbanec, T., & Meštrović, A. (2017, May). The struggle with academic plagiarism: Approaches based on semantic similarity. In 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 870-875). IEEE. https://doi.org/10.23919/MIPRO.2017.7973544

[13] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. https://doi.org/10.18653/v1/S17-2001

[14] mteb/stsbenchmark-sts. Retrieved from https://huggingface.co/datasets/mteb/stsbenchmark-sts

[15] Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, *33*, 16857-16867. https://doi.org/10.48550/arXiv.2004.09297

[16] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. https://doi.org/10.48550/arXiv.1907.11692

[17] Feng, F., Yang, Y., Cer, D., Arivazhagan, N., & Wang, W. (2020). Language-agnostic BERT sentence embedding. https://doi.org/10.48550/arXiv.2007.01852

АНАЛІЗ СЕМАНТИЧНОЇ СХОЖОСТІ З ВИКОРИСТАННЯМ ВЕКТОРНИХ ПРЕДСТАВЛЕНЬ РЕЧЕНЬ НА ОСНОВІ ТРАНСФОРМЕРІВ

Богдан Павлишенко , ***Микола Стасюк*** *
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 Львів, Україна

АНОТАЦІЯ

Вступ. Моделі трансформерів стали центральним елементом обробки природної мови, демонструючи еталонну продуктивність в оцінці семантичної схожості, що є критично важливим завданням для різноманітних додатків. Ці моделі фіксують деталі взаємозв'язків між текстами, розширюючи можливості оцінки семантичної спорідненості.

Матеріали та методи. Ефективність моделей вбудовування речень, зокрема all-mpnet-base-v2, all-MiniLM-L6-v2, paraphrase-multilingual-mpnet-base-v2, bge-base-en-v1.5, all-roberta-large-v1, all-distilroberta-v1, LaBSE, paraphrase-MiniLM-L3-v2, bge-large-en-v1.5, на двох наборах даних. До даних застосовано такі кроки попередньої обробки: переведення в нижній регістр, видалення стоп-слів, очищення від спеціальних символів і цифр та лематизація. Від'ємні значення косинусного коефіцієнта подібності, що вказують на семантичну несхожість, розглядалися як еквівалент оцінки подібності, що дорівнює 0, а невід'ємні значення косинусного коефіцієнта подібності масштабувалися до діапазону 0-5. Для оцінки використовувалися такі показники, як R^2, MSE, RMSE, MAE, коефіцієнт кореляції Спірмана та коефіцієнт кореляції рангу Кендала.

Результати. Продуктивність моделей загалом покращується зі збільшенням обсягу даних. За результати оцінювання моделей вбудовування речень виявилено різницю в ефективності. all-roberta-large-v1 продемонструвала високу точність з високими значеннями R^2 і низькими помилками. BAAI/bge-large-en-v1.5 чудово вловлює семантичні зв'язки, демонструючи високі значення коефіцієнтів Спірмена та Кендалла. all-MiniLM-L6-v2 демонструє найшвидше генерування вбудувань. BAAI/bge-base-en-v1.5 показує найнижчу точність. Час обробки, як правило, збільшується зі збільшенням розміру даних.

Висновки. Це дослідження висвітлює компроміс між продуктивністю та обчислювальною ефективністю при вбудовуванні речень. Вибір моделі залежить від балансування цих факторів відповідно до конкретних потреб програми. У випадках, коли вимагається висока точність, слід надавати перевагу all-roberta-large-v1, а тоді як пріоритетом є швидкість, краще використовувати all-MiniLM-L6-v2. BAAI/bge-large-en-v1.5 найкраще підходить для завдань, що вимагають розуміння семантичних деталей.

Ключові слова: семантична схожість, вбудування речень, трансформери.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
15 May, 2025	02 June, 2025	09 June, 2025	26 June, 2025