Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 24, No. 3, July 2025, pp. 451∼464
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 10/C/C3/DT.05.00/2025
DOI: 10.30812/matrik.v24i3.4727

Enhancing Semantic Similarity in Concept Maps Using Large Language Models

Muhammad Zaki Wiryawan[1], Didik Dwi Prasetya[1], Anik Nur Handayani[1], Tsukasa Hirashima[2], Wahyu Styo Pratama[1], Lalu Ganda Rady Putra[3]
[1]Universitas Negeri Malang, Malang, Indonesia
[2]Hiroshima University, Hiroshima, Jepang
[3]Universitas Bumigora, Mataram, Indonesia

Article Info

Article history:

Received December 18, 2024
Revised December 27, 2024
Accepted June 19, 2025

Keywords:

Concept Map;
Large Language Model;
Semantic Similarity;
Transformer.

ABSTRACT

This research uses advanced models, Generative Pre-trained Transformer-4 and Bidirectional Encoder Representations from Transformers, to generate embeddings that analyze semantic relationships in open-ended concept maps. The problem addressed is the challenge of accurately capturing complex relationships between concepts in concept maps, commonly used in educational settings, especially in relational database learning. These maps, created by students, involve numerous interconnected concepts, making them difficult for traditional models to analyze effectively. In this study, we compare two variants of the Artificial Intelligence model to evaluate their ability to generate semantic embeddings for a dataset consisting of 1,206 student-generated concepts and 616 link nodes (Mean Concept = 4, Standard Deviation = 4.73). These student-generated maps are compared with a reference map created by a teacher containing 50 concepts and 25 link nodes. **The goal** is to assess the models' performance in capturing the relationships between concepts in an open-ended learning environment. **The results show that demonstrate** that Generative Pretrained Transformers outperform other models in generating more accurate semantic embeddings. Specifically, Generative Pre-trained Transformer achieves 92% accuracy, 96% precision, 96% recall, and 96% F1-score. This highlights the Generative Pretrained Transformer's ability to handle the complexity of large, student-generated concept maps while avoiding overfitting, an issue observed with the Bidirectional Encoder Representations from Transformer models. **The key contribution** of this research is the ability of two complex models and multi-faceted relationships among concepts with high precision. This makes it particularly valuable in educational environments, where precise semantic analysis of open-ended data is crucial, offering potential for enhancing concept map-based learning with scalable and accurate solutions.

Corresponding Author:

Muhammad Zaki Wiryawan,
Department of Electrical Engineering and Informatics, Faculty of Engineering,
State University of Malang, Malang, Indonesia,
Email: didikdwi@um.ac.id

How to Cite:
M. Z. Wiryawan, D. D. Prasetya, A. N. Handayani, T. Hirashima, W. S. Pratama, and L. G. R. Putra, "Enhancing Semantic Similarity in Concept Maps Using Large Language Models", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, vol.

24, no. 3, pp. 452–462, Jun. 2025, doi: 10.30812/matrik.v24i3.4727.

1. INTRODUCTION

Concept maps are an effective visual tool for representing an individual's knowledge based on their cognitive understanding. Concept maps illustrate the meaningful relationships between concepts by linking various ideas or concepts through nodes and connecting lines [1, 2]. These maps are particularly valuable in educational settings, where they are used to foster critical thinking, an essential skill for learners in the modern world [3, 4]. Students can better organize their thoughts, systematically arrange information, and better understand the material using concept maps. Additionally, concept maps connect prior knowledge with new information, making them a powerful tool for enhancing learning. With the advent of computer-based technologies, there has been a significant shift toward more interactive and dynamic concept maps, offering an improved learning experience [5]. These advancements help make the learning process more engaging and effective. However, challenges remain in ensuring that concept maps created by teachers and students are aligned, given the variations in understanding and learning styles. As a result, a more objective approach is needed to assess the semantic similarity between concept maps, particularly those that are open-ended [6]. Open-ended concept maps allow individuals to freely generate and connect ideas without predefined constraints, resulting in unique, personalized structures that reflect each person's understanding of a topic [7]. While this approach encourages diverse perspectives, it also complicates consistent assessment. Measuring semantic similarity in this context is essential to identifying the alignment or divergence between different maps, providing a more objective and comprehensive way to evaluate how knowledge is represented.

Semantic similarity is a metric used to assess how closely two concepts or entities are related in terms of meaning. Concept maps evaluate the degree of relationship or proximity between two or more concepts based on the meanings they represent. For example, the concepts of "query formulation" and "query evaluation" exhibit a high degree of semantic similarity, as both are crucial in database query processing and optimization. Measuring the semantic similarity between concept maps can assess how accurately a concept map represents the knowledge the creator intends to communicate. However, this task is challenging due to the variations in terminologies and conceptual structures used by different individuals. As a result, an accurate tool is needed to capture semantic meaning consistently. In concept maps, traditional methods often depend on manual effort and individual interpretation to identify relationships between concepts. This makes the process time-consuming and prone to errors, particularly when dealing with large datasets [1, 8]. Technology-driven approaches, such as Word2Vec and GloVe [9, 10], have been used to assess semantic similarity; however, these models have limitations as they focus primarily on individual words without considering the broader context. Additionally, token-based approaches like TF-IDF measure similarity by calculating the angle between word vectors, but these methods remain lexical [11]. They rely on word vectors without capturing the text's full meaning, often overlooking deeper semantic connections.

Another approach, FastText, improves upon Word2Vec and GloVe by incorporating sub-word information, which allows for better handling of out-of-vocabulary words and capturing more nuanced semantic relations [12, 13]. However, like the methods above, FastText still does not fully capture contextual meaning at a sentence or document level. This is where transformer-based models, such as GPT (Generative Pre-Trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers), present a more advanced solution for semantic similarity. Unlike earlier models, GPT and BERT can understand and generate language by considering the broader context of words within sentences or documents [14]. This ability to capture contextual meaning enables these models to assess semantic similarity much deeper, moving beyond mere lexical matching. By leveraging their powerful architecture, GPT's generative capabilities, and BERT's bidirectional attention mechanism, these models can capture intricate relationships between concepts, making them well-suited for more complex tasks like concept map analysis. These models provide better semantic understanding by considering the entire context in which words or concepts appear and capture relationships between concepts.

This research is important because it addresses a critical **gap** in the accurate and efficient semantic similarity analysis in open-ended concept maps, a widely used tool in educational settings. Concept maps help students organize their thoughts, connect prior knowledge with new information, and foster critical thinking. However, evaluating open-ended concept maps remains challenging due to variations in students' understanding, terminology, and conceptual structures. Previous research has not resolved some gaps, namely the limitations of traditional semantic similarity methods—such as Word2Vec, GloVe, and FastText—which rely on lexical-level representations and fail to capture the broader contextual meaning of concepts. When applied to diverse, open-ended educational content, these models often struggle with semantic ambiguity, lack of contextual awareness, and low scalability. Moreover, prior studies have not sufficiently explored the use of transformer-based models like GPT and BERT in concept map analysis. **The difference** between this research and the previous one is the integration of advanced transformer-based language models (GPT and BERT) to enhance semantic similarity analysis in concept maps. Unlike previous methods, GPT and BERT can process local and global context, enabling a more dynamic and accurate understanding of the relationship between concepts. This research applies these models specifically to concept map evaluation. This area has not been extensively studied with transformer architectures, thus offering a novel approach to semantic analysis in educational tools. **This research aims** to improve the accuracy and efficiency of semantic similarity measurement in open-ended concept maps by applying transformer-based models. **The contributions** of this

research to the development of science are, introducing the use of GPT and BERT for analyzing semantic similarity in concept maps, providing a more context-aware and scalable method compared to traditional models; secondly, demonstrating the effectiveness of transformer models in educational data analysis, particularly in automatic assessment, knowledge discovery, and content organization; and finally, advancing the field of educational technology by enabling more objective and precise evaluation tools, which can support personalized learning and curriculum development. This integration marks a significant advancement in concept map analysis by offering a more contextualized and precise approach, ultimately contributing to developing more effective educational assessment tools.

2. RESEARCH METHOD

This research adopts an experimental methodology to investigate how the GPT and BERT models can enhance semantic similarity analysis within concept mapping. The research follows a systematic series of interconnected stages, including data collection, preprocessing, and model training using GPT and BERT. The key steps involve generating contextual embeddings, calculating semantic similarity between concepts, and evaluating the performance of both models in terms of accuracy and efficiency. Each phase is designed to ensure that the models can effectively capture the semantic relationships between concepts in the concept map, thereby improving the overall representation and understanding of knowledge. A detailed overview of each stage is provided in Figure 1.

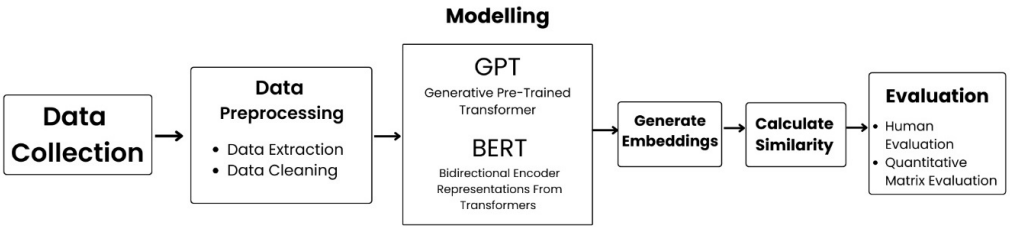


Figure 1. Research flow diagram

2.1. Data Collection

The data utilized in this study consists of concept maps derived from learning materials on relational databases. The dataset was produced by 27 students who participated in prior research [8, 15]. comprising 1,206 concepts and 616 link nodes. Each student's concept map is compared against a reference map containing 50 concepts and 25 link nodes. The dataset is intentionally diverse, capturing various concepts and relationships across different topics and cognitive levels. Given the open-ended nature of the concept maps, students were allowed to generate and connect ideas without predefined constraints. This open-ended approach resulted in varied concept map structures that reflect individual students' understanding and cognitive frameworks. To ensure sufficient variation and representation, the dataset includes maps from multiple students, each with their learning style and understanding of the subject matter. These concepts cover a range of educational backgrounds and varying levels of complexity in the relationships between concepts. For example, some concepts show straightforward, linear relationships between concepts, while others involve more complex, hierarchical, or non-linear structures. Figure 2 illustrates how nodes (concepts) and link nodes collectively form propositions in a concept map. A regular expression approach in Python is employed to identify the number of concepts and link nodes within the dataset, enabling the detection of patterns in the relationships between concepts [16, 17].

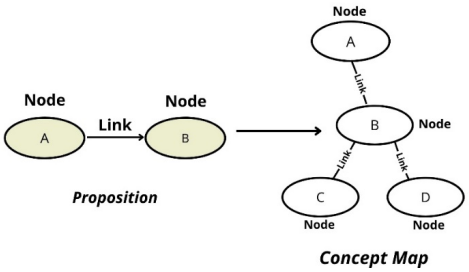


Figure 2. Collect concept data

2.2. Data Preprocessing

The data preprocessing stage in this research involves a series of simple but crucial procedures that aim to prepare the dataset to be used optimally in the modeling process [18, 19]. This stage begins with Data extraction from a SQL-based database obtained from the concept map build kit, where the concept map-related data and concept descriptions are extracted into raw text format (.txt) for easy manipulation and further analysis. Figure 3 explains the extraction process from SQL on the concept map build kit to text. After the extraction is successful, the data in the text format is converted into CSV format. The CSV format allows each row of data to represent a single entity or concept, and each column describes the characteristics or description of the concept, making it easier to learn the model. The data cleaning process involves removing unnecessary punctuation, such as commas, periods, and symbols, to prevent interference with the semantic understanding process [20]. This step ensures the data focuses on meaningful content by eliminating irrelevant elements [19, 21]. Once cleaned, the structured and consistent CSV data is prepared for the semantic similarity modeling stage, where GPT and BERT models are trained to better capture inter-concept relationships. This preprocessing enhances the dataset's quality, optimizing the models' performance by mapping semantic similarities within educational concepts.

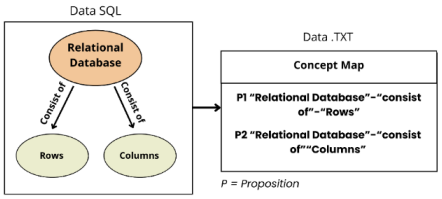


Figure 3. Illustration of extracting a concept map

2.3. Modelling for Semantic Similarity

The modeling phase of this research centers on assessing the semantic similarity between two concept maps—those created by teachers and those created by students using embeddings generated by two different models, GPT and BERT. Each model generates embeddings from the concepts within the maps. An embedding is a high-dimensional vector that converts concepts, words, or data into numerical representations [22, 23]. These vectors capture various semantic relationships among the items, allowing the model to process and comprehend the data. Each dimension in an embedding vector corresponds to a distinct feature or attribute of the input data, enabling the model to perform mathematical operations, such as calculating dot products or measuring distances between vectors [24, 25]. The embedding vectors are then used to quantify the degree of similarity between the two concept maps being compared. Figure 4 below explains how the GPT model embeds an open-ended concept map.

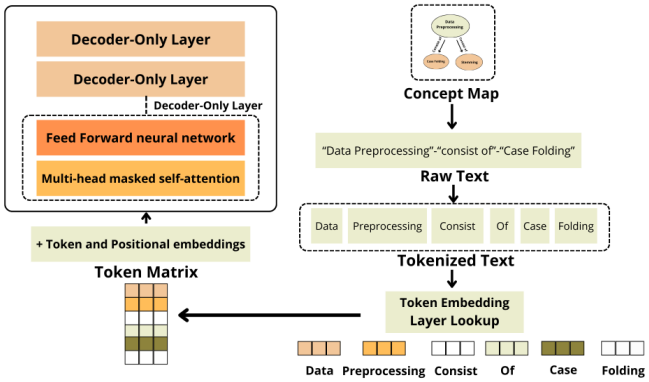


Figure 4. GPT embedding process

The embedding process in the GPT model starts with data preprocessing, where the raw text is broken into tokens through tokenization, such as separating the words "Data" and "Preprocessing", and case folding is performed to convert all letters into lowercase. After tokenization, the processed tokens are converted into numerical representations through the Token Embedding Layer, resulting in a matrix that includes information about the position and meaning of each token. Next, the model architecture consists of decoder-only layers, which include several layers with two main components: multi-head masked self-attention, which allows

the model to focus on the context of relevant tokens, and a feed-forward neural network, which processes information to generate more complex representations. Finally, the model generates text output based on the processed representations, thus enabling natural language understanding and generation. This embedding process is critical as it transforms the raw text into a format the model can understand, supporting its ability to understand and generate language. Generative Pre-trained Transformer (GPT) models, including GPT-2, GPT-3.5, and GPT-4, use the Transformer architecture, which involves a self-attention mechanism to capture relationships between tokens in a text sequence [26–28]. The mathematical formulation of the self-attention mechanism applied at each layer of a Transformer can be expressed in Equation 1.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{dk}}\right)V \tag{1}$$

In this formula, Q, K, and V are Query, Key, and Value, respectively, which are vector representations of the inputs generated through the linear transformation of the embedding [29, 30]. The dot product is used to measure the level of relevance or similarity between each input element. Then, the result is divided by the key dimension to maintain the stability of the calculation. Softmax is used to convert this relevance score into a probability, which shows how much attention is paid to each element. The final result is obtained by multiplying this probability by the Value. This results in a new vector representation, combining relevant information from all input elements according to the calculated attention scale. $QK^T\sqrt{dk(V)}$. At each GPT layer, the input's embedding undergoes multiple transformations using self-attention and feed-forward neural networks, resulting in more complex and in-depth semantic representations [31]. At the end of the process, the output is a semantically rich embedding vector representing the context the model understands.

Unlike GPT, BERT is based on the Transformer encoder architecture, processing text bidirectionally, meaning that the embedding of each token takes into account the context of both the preceding and following words, allowing the model to understand the full context of the sentence. To capture richer and deeper information, BERT is trained using Masked Language Modeling (MLM) and next-sentence prediction (NSP). The embedding process in the BERT model starts with data preprocessing, where raw text is broken down into tokens through tokenization, resulting in tokens such as "[CLS]", "man", "is", "riding", "horse", and "[SEP]". These tokens are then converted into numerical representations through the Token Embedding Layer, which generates a vector for each token. Mathematically, the token representation for the token t_i in the input, it can be written as $E(t_i)\epsilon Rd$ where $E(t_i)$ is an embedding vector for the token t_i, and d is the dimension of the embedding vector space. Since Transformer does not have sequence information in its architecture, Positional Encoding provides information about each token's position in the sentence sequence. The positional encoding is computed using the sinusoidal formula, as described in Equation 2.

$$\begin{gathered} PE(p, 2i) = sin\left(\frac{p}{10000^{2i/d}}\right) \\ and \\ PE(p, 2i+1) = cos\left(\frac{p}{10000^{2i/d}}\right) \end{gathered} \tag{2}$$

In this formula, p refers to the token's position in the sequence, i is the dimension index, and d is the total number of dimensions in the embedding. Sine and cosine functions are used alternately across even and odd dimensions to effectively produce unique patterns that encode token positions. In addition, BERT also adds Segment Embedding to distinguish between two text segments, which is often used in tasks such as question answering. For example, for the question and context in a question-answering task, we want to give different labels to tokens coming from the first and second sentences. This segment embedding can be expressed as $S(t_i)\epsilon Rd$, where $S(t_i)$ is the embedding segment for the t_i. Thus, the outputs of the three embeddings—token, positional, and segment—are combined to produce the final representation of each token in the input, as formulated in Equation 3.

$$X_i = E(t_i) + PE(p_i) + S(t_i) \tag{3}$$

As shown in the equation above, X_i refers to the final embedding of the i-th token, which is obtained by summing the token embedding $E(t_i)$, the positional encoding $PE_{(pi)}$, and the segment embedding $S(t_i)$. The token embedding captures the semantic meaning of the token, the positional encoding adds information about the token's position in the sequence, and the segment embedding helps distinguish tokens belonging to different segments, such as question and context. This combined embedding reflects both the identity and contextual role of the token. The resulting vector X_i is then passed through BERT's Transformer layers, where attention mechanisms further refine it based on surrounding tokens, enabling deep contextual representation.

The BERT model then processes this representation through several layers of transformers, which include multi-head self-attention and feed-forward neural networks, to produce an output that understands the context and relationships between tokens.

Finally, the model produces a representation that can be used for various natural language processing tasks, such as text classification or information extraction. The entire embedding stage is described in Figure 5.

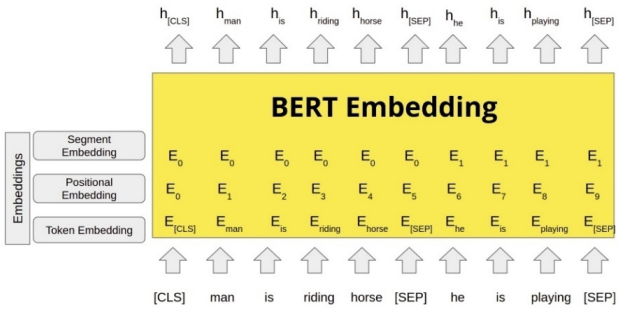


Figure 5. BERT embedding process

After the embedding for the concepts of the two maps is generated, a cosine similarity calculation is carried out for each embedding pair between the concepts of the teacher's map and the student's map. Cosine similarity is a metric used to measure the degree of similarity between two embedding vectors [32]. Cosine similarity is robust to variations in term frequency and vector length, which is an advantage when working with concept maps where the distribution of concepts and their relationships may vary significantly. Although cosine similarity does not fully account for all aspects of semantic complexity (e.g., hierarchical relationships), it offers a simple yet powerful measure for evaluating general semantic similarity in concept map analysis. Unlike other methods, such as Euclidean or Manhattan distance, cosine similarity focuses on the direction of the vectors rather than their magnitude, making it particularly suitable for tasks involving high-dimensional embeddings like those generated by transformer models. Cosine similarity is mathematically expressed using the following formula, as shown in Equation 4.

$$\textit{Cosine Similarity}\ (A, B) = \frac{A.B}{||A|| \times ||B||} \tag{4}$$

Where is the embedding vector of the two concepts to be compared, is the dot product of the two vectors, i.e., the result of the multiplication of the corresponding elements of the two vectors, A .B and $||A||$ and $||B||$ are the norms (magnitude) of the vector, which is calculated by taking the square root of the sum of the squares of each element in the vector. Using this formula, each pair of concepts from the teacher and student maps is compared to calculate cosine similarity, and the results are arranged in the form of a matrix that depicts the degree of similarity between concepts on both maps. Further analysis was conducted to determine whether the cosine similarity value was high enough for the two concepts to be compatible, using a specific threshold such as 0.8 [33]. This threshold was chosen as it is suitable for assessing high similarity in cases with similar contexts. In addition, the cosine similarity calculation results for embedding the three models were compared to evaluate their respective performance in capturing semantic relationships between concepts. Evaluation using precision, recall, and F1 score metrics helps assess the models' ability to identify semantic similarity accurately. The comparison between BERT and GPT-4 using cosine similarity provides insight into how the models can detect context patterns in concept maps.

2.4. Evaluation

This evaluation process involves expert human judgment to assess the semantic similarities between concepts generated by the model in the concept map. The experts manually evaluate whether two concepts are semantically similar, and their assessments are then compared with the model's predictions. This comparison creates a confusion matrix, from which evaluation metrics such as accuracy, precision, recall, and F1 score are derived. Three experts participate in the manual assessment to determine if two concepts are semantically similar. The model's predictions are then analyzed to gauge how well it identifies these semantic similarities. The confusion matrix reveals the model's performance in correctly classifying similar and dissimilar concepts. Accuracy is calculated to show how often the model makes correct predictions overall. Precision indicates how accurately the model predicts similarity when it claims two similar concepts. At the same time, recall measures how much of the true semantic similarity the model can detect. The F1 score combines precision and recall, providing a balanced evaluation of the model's performance. This methodology allows for a thorough evaluation of the model's effectiveness in identifying semantic similarity in concept mapping, ensures the predictions are empirically validated by experts, and offers insights into the efficacy of GPT and BERT models in concept mapping tasks.

3. RESULT AND ANALYSIS

This section presents the results of each modeling stage in this research, starting with data embedding. After the data is preprocessed, the concept map entries are transformed into embeddings, which are high-dimensional vector representations that encapsulate the semantic meaning of the concepts. This transformation enables GPT models to perform mathematical comparisons of semantic similarities. Table 1 illustrates how each concept in the teacher's and student's maps is translated into vectors, which are crucial for calculating semantic similarity.

Table 1. Embedding results of the GPT model

Preprocessed Text	Embedding
Teacher two dimensional tables consist of columns or attributes two dimensional tables are also referred to as relations two dimensional tables have only one primary key	[-0.01038203202188015, -0.01166575588285923, -0.028480438515543938, 0.016527071595191956, -3.197801561327651e-05, -0. 0009487633942626417, 0.029518641531467438, 0.0030847962480038404, -0.03779620677232742, -0.008579205721616745, -0.043857067823410034, 0.02825596183538437, -, (array) . . .]
Student 1 relational database definition two dimensional table two dimensional table consists of rows two dimensional table consists of columns	[-0.013112479820847511,0.01584303006529808,-0.024239106103777885, 0.02574309892952442,0.00361578818410635,0.01940588653087616, 0.024297513067722,0.015857631340622902,-0.035657767206430435, -0.005508555565029383,-0.0355993621051311, (array) . . .]

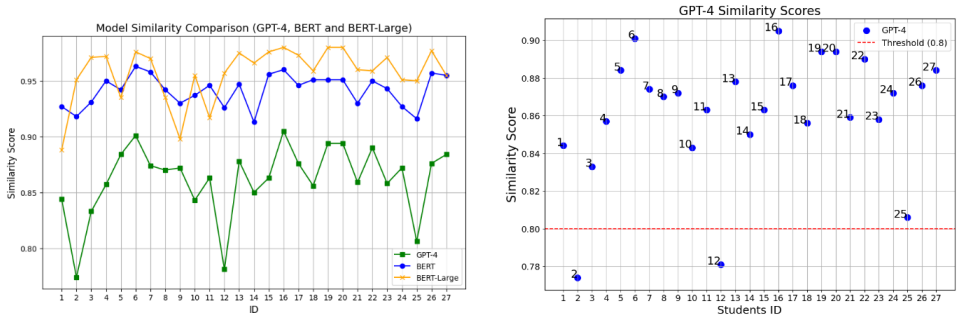
For example, in the teacher's concept map, the sentence "Relational databases use a 'two-dimensional table' structure" is converted into a numerical vector consisting of several values, such as [-0.01038, -0.01166, -0.02848, . . .]. These values represent the semantic relationships of the concept, where each number indicates a position in the vector space generated by the GPT and BERT models. Likewise, the concepts on the student concept map, such as "relational database definition two-dimensional," are converted into vectors such as [-0.13112, 0.01584, -0.02423, . . .]. This difference in values in embedding shows how the model understands these concepts based on its semantic context. With this embedding, the model can then calculate semantic similarity between teacher and student concepts using metrics such as cosine similarity. The more similar the two concepts are, the closer their embedding values are in the vector space, showing the similarity of students' and teachers' understanding. This embedding process is important to transform the concept from a text format into a mathematical representation that the model can analyze to determine the degree of similarity between concepts.

Table 2. Pembagian data untuk Training dan Testing

ID Student	GPT 4	BERT Based	BERT Large	Ground Truth
1	0.844	0.927	0.888	1
2	0.774	0.918	0.951	0
3	0.833	0.931	0.971	0
4	0.857	0.95	0.972	1
5	0.884	0.942	0.935	1
6	0.901	0.963	0.976	1
7	0.874	0.958	0.970	1
8	0.870	0.942	0.935	1
9	0.872	0.93	0.898	1
10	0.843	0.937	0.955	1
11	0.863	0.946	0.917	1
12	0.781	0.926	0.957	1
13	0.878	0.947	0.975	1
14	0.850	0.913	0.966	1
15	0.863	0.956	0.976	1
16	0.905	0.96	0.980	1
17	0.876	0.946	0.973	1
18	0.856	0.951	0.959	1
19	0.894	0.951	0.980	1
20	0.894	0.951	0.980	1
21	0.859	0.93	0.960	1
22	0.890	0.95	0.959	1
23	0.858	0.943	0.971	1
24	0.872	0.927	0.951	1
25	0.806	0.916	0.950	1
26	0.876	0.957	0.977	1
27	0.884	0.955	0.955	1

On the other hand, BERT and BERT Large tend to give higher values, with many values above 0.9. For example, in student 6 (0.963 for BERT and 0.976 for BERT Large) and student 5 (0.942 for BERT and 0.935 for BERT Large), these two models provide very high similarity predictions in most cases. This leads to the tendency that the BERT model may over-predict similarity more often, which could indicate overfitting, especially since there is a tendency to give scores above the 0.8 threshold even though, in some cases, the actual similarity could be lower. Meanwhile, the similarity scores on BERT Large were often slightly higher than the standard BERT. Still, the variation between students was not much different, which suggests that BERT Large tends to be more consistent in assessing similarity, with a tendency to give slightly higher scores overall.

Based on the semantic similarity prediction results in Table 2, some interesting patterns exist for analysis. In general, higher semantic similarity indicates that the models tend to perceive the concepts as more similar, but very high values across students may indicate overfitting. The GPT-4 model showed more variable results than BERT and BERT Large, with many values below 0.9, such as for student 2 (0.774), student 12 (0.781), and student 25 (0.806). These values suggest that GPT-4 tends to be more cautious in assessing semantic similarity, better able to capture more subtle differences between student and teacher concept maps, and less likely to give high scores that may not be relevant. When values are lower than 0.8, as seen in student 2 and student 12, the model is more cautious in identifying similarities that may not be significant enough. This suggests that GPT-4 does not suffer from overfitting and can better handle more complex semantic differences. The entire Similarity results is described in Figure 6.



(a) (b)

Figure 6. Similarity results (a) comparison of the similarity results of the GPT and BERT models (b) GPT-4 similarity results with threshold

The overall visualization suggests that while BERT and BERT-Large appear to excel in semantic similarity scores, they both exhibit a significant drawback characterized by overfitting. GPT-4 demonstrates a commendable capacity to navigate complex conceptual variations, offering a more dependable option for such tasks. However, it tends to exhibit slightly diminished performance on simpler assignments. This underscores the significance of choosing models that align with the task's complexity and the variability present in the data, indicating that simpler models like BERT might yield superior outcomes in certain scenarios. However, more intricate models like GPT-4 demonstrate superior capabilities for tasks necessitating a profound grasp of semantics.

Table 3. Evaluation results of the BERT and GPT models

Model	**ACCURACY**	**PRECISION**	**RECALL**	**F1-SCORE**
GPT-4	**0.92**	**0.96**	**0.96**	**0.96**
BERT	0.926	0.926	1	0.962
BERT Large	0.926	0.926	1	0.962

Based on the model performance results shown in Table 3, BERT and BERT-Large have the same accuracy and precision values of 0.926, as well as a perfect recall of 1.0, which indicates the tendency of both models to predict almost all concept pairs as "similar," indicating overfitting. While this high recall value means that the two models hardly miss out on completely similar concepts, it also indicates that the model cannot distinguish between truly similar concepts and those that are not. On the other hand, GPT-4 has a slightly lower accuracy value (0.92) but shows a higher precision (0.96) and a recall of 0.96, which results in an F1-score of 0.96. This shows that GPT-4 is more balanced and selective in predicting semantic similarities, where the model can distinguish concepts better and avoid overfitting experienced by BERT and BERT-Large. Thus, despite its slightly lower accuracy value, GPT-4 is more accurate and careful in determining whether two concepts are truly similar, making it more reliable for tasks that require a deeper and more accurate understanding of semantics.

Table 4. Time and memory usage of BERT and GPT model

Model	Inference Time (s)	All Memory Usag (MB)	Embeddings Memory Usage (MB)
BERT	18	70.195	105.469
BERT-Large	66	86.367	79.102
GPT-4	18	220.246	217.093

The performance analysis of the three models, BERT, BERT Large, and GPT-4, presented in Table 4, highlights significant differences in inference time and memory usage. BERT shows a relatively fast inference time (18 seconds) with embedding memory usage of 105.469 MB and total memory usage of 70.195 MB. Despite having a faster inference time than BERT-Large, BERT tends to have a higher embedding memory usage. BERT-Large, on the other hand, takes longer inference time (66 seconds) but has lower embedding memory usage (79.102 MB) despite its higher total memory usage (86.367 MB). This shows that BERT-Large is more efficient in embedding memory usage despite its longer inference time. On the other hand, GPT-4 has an inference time equivalent to BERT (18 seconds) but with a much higher embedding memory usage of 217,093 MB and a total memory usage of 220,246 MB. Although GPT-4 has a similar inference time to BERT, its overall memory usage is much larger, reflecting the higher complexity and depth of the model.

The findings of this research are that transformer-based models, specifically GPT and BERT, significantly outperform traditional word embedding methods such as Word2Vec, GloVe, and FastText in measuring semantic similarity within open-ended concept maps. The results show that GPT and BERT can capture local and global context, leading to more accurate identification of concept relationships, even when students use varied terminology or non-standard structures. This improvement is particularly evident in complex, unstructured student-generated content where traditional models fail to maintain semantic coherence. Compared to previous research, which primarily relied on lexical-based similarity using static embeddings, the transformer-based approach in this study offers dynamic contextual understanding. For instance, prior works using Word2Vec and FastText demonstrated reasonable performance in structured and small-scale datasets, but struggled with scalability and contextual depth. In contrast, this research demonstrates that GPT and BERT scale more effectively across larger datasets and provide more nuanced semantic judgments that align more closely with human evaluation. Overall, the results show that GPT-4 produces more accurate semantic similarity predictions than BERT and BERT-Large in the context of concept maps. **This is in line** with several other studies that highlight the importance of AI-based tools to make it easier to measure semantic relationships in education [34, 35]. The results also found that the BERT and BERT-Large models tend to overfit, giving a relatively high similarity score [36, 37]. To overcome overfitting in BERT and BERT-Large models, several techniques that can be applied later are fine-tuning with model size restrictions and model hyperparameter training. This technique can help improve the model's generalization ability and reduce the possibility of the model overfitting to the training data, thereby improving the accuracy in predicting semantic similarity.

Additional studies of GPT-4's performance in scenarios involving deep semantic understanding confirm that newer models have more sophisticated mechanisms to avoid over-predicting [38]. The study noted that improved parameters and attention mechanisms in GPT-4 allow for more flexible handling of smaller semantic differences, making it a more accurate tool. The contributions of this research to the development of science are as follows. Firstly, this research introduces the application of GPT and BERT in analyzing semantic similarity within concept maps, offering a more context-aware and scalable alternative to previous methods. Secondly, it demonstrates the practical effectiveness of transformer models in educational data analysis, especially for automatic evaluation, knowledge extraction, and content organization. Finally, it advances the field of educational technology by enabling the development of more objective, accurate, and personalized assessment tools. This integration marks a significant advancement in concept map analysis by offering a more contextualized and precise approach, ultimately contributing to enhancing educational evaluation and learning analytics.

4. CONCLUSION

This study explores the application of GPT-4 and BERT models in semantic similarity analysis of concept maps, specifically to compare the semantic understanding between concepts created by students and teachers in relational database materials. Concept map data is processed through pre-processing stages, such as format conversion to a more structured format and text cleaning. Next, the data were converted into high-dimensional vector representations using GPT-4 and BERT, and semantic similarity was measured by the cosine similarity method. Results show significant differences in the assessment of semantic relationships: GPT-4 produced a wider variety of values with accuracy, precision, recall, and F1-score of 0.92, 0.96, 0.96, and 0.96, respectively, indicating its ability to assess relevant similarities selectively. On the other hand, BERT and BERT-Large provided more consistent results but tended to suffer from overfitting, with precision and accuracy values of 0.926, recall of 1.0, and F1-score of 0.962. Analysis of inference time and memory usage shows BERT is more efficient with 18 seconds and 70.195 MB memory usage compared to GPT-4 -4 which requires 220.246 MB memory.

The results of this study indicate that model selection depends on the needs of the application. GPT-4 excels in deep semantic understanding with higher accuracy, suitable for tasks that require detailed judgment. In contrast, BERT is more time and memory-efficient but prone to overfitting. This study demonstrates the potential application of the transformer model in education, especially for developing concept map auto-evaluation tools that help educators objectively and efficiently assess student understanding. By using GPT-4, semantic similarity evaluation becomes more accurate and opens up opportunities for the development of interactive adaptive learning systems to improve student feedback and understanding more effectively

5. ACKNOWLEDGEMENTS

The Acknowledgments section is optional. Research sources can be included in this section.

6. DECLARATIONS

AUTHOR CONTIBUTION

FUNDING STATEMENT

COMPETING INTEREST

REFERENCES

[1] D. D. Prasetya, T. Widiyaningtyas, and T. Hirashima, "Interrelatedness patterns of knowledge representation in extension concept mapping," *Research and Practice in Technology Enhanced Learning*, vol. 20, no. 09, pp. 2–18, may 2024, https://doi.org/10.58459/rptel.2025.20009.

[2] A. Pinandito, C. P. Wulandari, D. D. Prasetya, Y. Hayashi, and T. Hirashima, "Knowledge Reconstruction with Kit-Build Concept Map: A Review from Student Experience," in *7th International Conference on Sustainable Information Engineering and Technology 2022*. New York, NY, USA: ACM, nov 2022, pp. 263–270, https://doi.org/10.1145/3568231.3568274. [Online]. Available: https://dl.acm.org/doi/10.1145/3568231.3568274

[3] X. Wang, C. F. Lee, Y. Li, and X. Zhu, "Digital Transformation of Education: Design of a "Project-Based Teaching" Service Platform to Promote the Integration of Production and Education," *Sustainability (Switzerland)*, vol. 15, no. 16, pp. 02–21, aug 2023, https://doi.org/10.3390/su151612658.

[4] S. Papadakis, "Tools for evaluating educational apps for young children: a systematic review of the literature," *Interactive Technology and Smart Education*, vol. 18, no. 1, pp. 18–49, may 2021, https://doi.org/10.1108/ITSE-08-2020-0127.

[5] S. Schneider, F. Krieglstein, M. Beege, and G. D. Rey, "How organization highlighting through signaling, spatial contiguity and segmenting can influence learning with concept maps," *Computers and Education Open*, vol. 2, p. 100040, dec 2021, https://doi.org/10.1016/j.caeo.2021.100040.

[6] F. Sciarrone and M. Temperini, "A Sentence-Embedding-Based Dashboard to Support Teacher Analysis of Learner Concept Maps," *Electronics*, vol. 13, no. 9, p. 1756, may 2024, https://doi.org/10.3390/electronics13091756.

[7] R. Mandasari and S. Winduwati, "Upaya Public Relations Pusbisindo dalam Mengampanyekan Penggunaan Bahasa Isyarat Indonesia di Kalangan Masyarakat," *Prologia*, vol. 6, no. 2, pp. 355–361, nov 2022, https://doi.org/10.24912/pr.v6i2.15572.

[8] D. D. Prasetya and T. Hirashima, "Associated Patterns in Open-Ended Concept Maps within E-Learning," *Knowledge Engineering and Data Science*, vol. 5, no. 2, p. 179, dec 2022, https://doi.org/10.17977/um018v5i22022p179-187. [Online]. Available: http://journal2.um.ac.id/index.php/keds/article/view/38346

[9] C.-H. Chuan, K. Agres, and D. Herremans, "From context to concept: exploring semantic relationships in music with word2vec," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1023–1036, feb 2020, https://doi.org/10.1007/s00521-018-3923-1.

[10] F. Sakketou and N. Ampazis, "A constrained optimization algorithm for learning GloVe embeddings with semantic lexicons," *Knowledge-Based Systems*, vol. 195, no. 14, pp. 02–10, may 2020, https://doi.org/10.1016/j.knosys.2020.105628.

[11] F. Lan, "Research on Text Similarity Measurement Hybrid Algorithm with Term Semantic Information and TF-IDF Method," *Advances in Multimedia*, vol. 2022, no. 7, pp. 1–11, apr 2022, https://doi.org/10.1155/2022/7923262.

[12] C. Tulu, "Experimental Comparison of Pre-Trained Word Embedding Vectors of Word2Vec, Glove, FastText for Word Level Semantic Text Similarity Measurement in Turkish," *Advances in Science and Technology Research Journal*, vol. 16, no. 4, pp. 147–156, oct 2022, https://doi.org/10.12913/22998624/152453.

[13] M. Umer, Z. Imtiaz, M. Ahmad, M. Nappi, C. Medaglia, G. S. Choi, and A. Mehmood, "Impact of convolutional neural network and FastText embedding on text classification," *Multimedia Tools and Applications*, vol. 82, no. 4, pp. 5569–5585, feb 2023, https://doi.org/10.1007/s11042-022-13459-x.

[14] A. P. Bhopale and A. Tiwari, "Transformer based contextual text representation framework for intelligent information retrieval," *Expert Systems with Applications*, vol. 238, no. 3, p. 121629, mar 2024, https://doi.org/10.1016/j.eswa.2023.121629.

[15] D. D. Prasetya, A. Pinandito, Y. Hayashi, and T. Hirashima, "Analysis of quality of knowledge structure and students' perceptions in extension concept mapping," *Research and Practice in Technology Enhanced Learning*, vol. 17, no. 1, p. 14, dec 2022, https://doi.org/10.1186/s41039-022-00189-9.

[16] M. Shin, M. Yoo, S. Kang, S. Choi, and S. Kim, "Proposal of smart contract collection and detection automation framework based on regular expression pattern matching," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 28, no. 4, pp. 454–466, apr 2024, https://doi.org/10.6109/jkiice.2024.28.4.454.

[17] M. Sun, G. Xie, F. Zhang, W. Guo, X. Fan, T. Li, L. Chen, and J. Du, "PTME: A Regular Expression Matching Engine Based on Speculation and Enumerative Computation on FPGA," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 18, no. 1, pp. 1–28, mar 2025, https://doi.org/10.1145/3655626.

[18] K. M. S. Prasad, "Text mining: identification of similarity of text documents using hybrid similarity model," *Iran Journal of Computer Science*, vol. 6, no. 2, pp. 123–135, jun 2023, https://doi.org/10.1007/s42044-022-00127-4.

[19] C. P. Chai, "Comparison of text preprocessing methods," *Natural Language Engineering*, vol. 29, no. 3, pp. 509–553, may 2023, https://doi.org/10.1017/S1351324922000213.

[20] Z. Jin, "Principle, Methodology and Application for Data Cleaning techniques," *BCP Business & Management*, vol. 26, pp. 724–732, sep 2022, https://doi.org/10.54691/bcpbm.v26i.2032.

[21] A. Petukhova and N. Fachada, "TextCL: A Python package for NLP preprocessing tasks," *SoftwareX*, vol. 19, no. 10, p. 101122, jul 2022, https://doi.org/10.1016/j.softx.2022.101122.

[22] V. Mehta, S. Bawa, and J. Singh, "WEClustering: word embeddings based text clustering technique for large datasets," *Complex & Intelligent Systems*, vol. 7, no. 6, pp. 3211–3224, dec 2021, https://doi.org/10.1007/s40747-021-00512-9.

[23] H. Yang, S. Wei, and Y. Wang, "STFEformer: Spatial–Temporal Fusion Embedding Transformer for Traffic Flow Prediction," *Applied Sciences*, vol. 14, no. 10, p. 4325, may 2024, https://doi.org/10.3390/app14104325.

[24] M. Chiny, M. Chihab, A. A. Lahcen, O. Bencharef, and Y. Chihab, "Effect of word embedding vector dimensionality on sentiment analysis through short and long texts," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 2, p. 823, jun 2023, https://doi.org/10.11591/ijai.v12.i2.pp823-830.

[25] P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe, "A Statistical Interpretation of Spectral Embedding: The Generalised Random Dot Product Graph," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 84, no. 4, pp. 1446–1473, sep 2022, https://doi.org/10.1111/rssb.12509.

[26] Y. Shin, J. Choi, H. Wi, and N. Park, "An Attentive Inductive Bias for Sequential Recommendation beyond the Self-Attention," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8, pp. 8984–8992, mar 2024, https://doi.org/10.1609/aaai.v38i8.28747.

[27] R. K. Singh, "Advancements in Natural language Processing: An In-depth Review of Language Transformer Models," *International Journal for Research in Applied Science and Engineering Technology*, vol. 12, no. 6, pp. 1719–1732, jun 2024, https://doi.org/10.22214/ijraset.2024.63408.

[28] R. Jia, Z. Zhang, Y. Jia, M. Papadopoulou, and C. Roche, "Improved GPT2 Event Extraction Method Based on Mixed Attention Collaborative Layer Vector," *IEEE Access*, vol. 12, no. 12, pp. 160 074–160 082, 2024, https://doi.org/10.1109/ACCESS.2024.3487836.

[29] A. de Santana Correia and E. L. Colombini, "Attention, please! A survey of neural attention models in deep learning," *Artificial Intelligence Review*, vol. 55, no. 8, pp. 6037–6124, dec 2022, https://doi.org/10.1007/s10462-022-10148-x.

[30] Y. Tian, F. Han, M. Zhu, X. Xu, and Y. Li, "Research on sign language gesture division and gesture extraction in complex background," in *International Conference on Computer Vision, Application, and Algorithm (CVAA 2022)*, H. Imane, Ed. SPIE, apr 2023, p. 21, https://doi.org/10.1117/12.2673290.

[31] S. Lyu, X. Zhou, X. Wu, Q. Chen, and H. Chen, "Self-Attention Over Tree for Relation Extraction With Data-Efficiency and Computational Efficiency," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 2, pp. 1253–1263, apr 2024, https://doi.org/10.1109/TETCI.2023.3286268.

[32] K. Singh, M. Mishra, and E. S. Singh, "Content-based Recommender System Using Cosine Similarity," *International Journal for Research in Applied Science and Engineering Technology*, vol. 12, no. 5, pp. 2541–2548, may 2024, https://doi.org/10.22214/ijraset.2024.61835.

[33] Y. Li, J. Wang, B. Pullman, N. Bandeira, and Y. Papakonstantinou, "Index-based, High-dimensional, Cosine Threshold Querying with Optimality Guarantees," *Theory of Computing Systems*, vol. 65, no. 1, pp. 42–83, jan 2021, https://doi.org/10.1007/s00224-020-10009-6.

[34] T. Alqahtani, H. A. Badreldin, M. Alrashed, A. I. Alshaya, S. S. Alghamdi, K. bin Saleh, S. A. Alowais, O. A. Alshaya, I. Rahman, M. S. Al Yami, and A. M. Albekairy, "The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research," *Research in Social and Administrative Pharmacy*, vol. 19, no. 8, pp. 1236–1242, aug 2023, https://doi.org/10.1016/j.sapharm.2023.05.016.

[35] V. J. Owan, K. B. Abang, D. O. Idika, E. O. Etta, and B. A. Bassey, "Exploring the potential of artificial intelligence tools in educational measurement and assessment," *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 19, no. 8, p. em2307, aug 2023, https://doi.org/10.29333/ejmste/13428.

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

[37] Y.-G. Xu, X.-P. Qiu, L.-G. Zhou, and X.-J. Huang, "Improving BERT Fine-Tuning via Self-Ensemble and Self-Distillation," *Journal of Computer Science and Technology*, vol. 38, no. 4, pp. 853–866, jul 2023, https://doi.org/10.1007/s11390-021-1119-0.

[38] G. Le Mens, B. Kovács, M. T. Hannan, and G. Pros, "Uncovering the semantics of concepts using GPT-4," *Proceedings of the National Academy of Sciences*, vol. 120, no. 49, pp. 1–7, dec 2023, https://doi.org/10.1073/pnas.2309350120.

[This page intentionally left blank.]