KEC Conference 2025

ISSN: 3021-940X (Print) DOI: https://doi.org/10.3126/injet.v2i2.78597

Semantic Similarity Analysis for Exam Questions Using Sentence Transformer Model

Nischal Shakya [1, *], Milan Shrestha [2], Roshan Subedi[3], Nitesh Swarnakar[4], Upendra Prasad Neupane[5, *], Sharad Kumar Ghimire[6]

[1]Pulchowk Engineering Campus, Tribhuvan University, Pulchowk, Nepal, 075bct055.nischal@pcampus.edu.np
[2]Pulchowk Engineering Campus, Tribhuvan University, Pulchowk, Nepal, 075bct050.milan@pcampus.edu.np
[3]Pulchowk Engineering Campus, Tribhuvan University, Pulchowk, Nepal, 075bct068.roshan@pcampus.edu.np
[4]Pulchowk Engineering Campus Tribhuvan University, Pulchowk, Nepal, 075bct058.nitesh@pcampus.edu.np
[5]Sagarmatha Engineering College, Tribhuvan University, Sanepa, Lalitpur, Nepal, info.upendrapn@gmail.com
[6]Pulchowk Engineering Campus, Tribhuvan University, Pulchowk, Lalitpur, Nepal, skghimire@ioe.edu.np

Abstract

The aim is to explore the effectiveness of using the SBERT model and vector database for performing question similarity analysis. This involves building a vector database by training a sentence transformer model on a large corpus of text data. The vector dataset is then used to analyse question similarity by retrieving similar questions and similarity scores to a given search query. The model is trained on a large corpus of ALLNLI datasets, other paraphrase datasets such as MRPC and PAWS, and the semantic similarity of datasets such as STS, and finally adapted on 9,282 custom-prepared engineering datasets. The sentence transformer model is trained using the datasets mentioned above with MNR Loss as the loss function. The effectiveness of the model is evaluated by using the STS test dataset and test set of the MRPC. The results demonstrate that using a sentence transformer model and vector database for question similarity analysis outperforms the baseline method of keyword matching. The approach achieved a Spearman correlation value of 0.863 on the STS benchmark and an accuracy of 88.7% on the MRPC test. The Spearman correlation value in the SBERT paper for the NLI-large dataset was below 0.80. These values show that continuous training of the model on other datasets besides NLI helps to increase the performance and performs better for downstream tasks. This suggests that the use of the sentence transformer model and vector database is a promising approach for performing question similarity analysis, which could have significant implications for information retrieval systems.

Keywords: Indexing, Information retrieval, Sentence transformer, Vector database

1. Introduction

Natural Language Processing (NLP) has become crucial in processing and analyzing large amounts of textual data, enabling machines to understand human language. With the emergence of advanced models like GPT-3 and BERT, NLP shows significant promise across various domains. One key application is similarity detection, which identifies semantic relationships between texts. In education, detecting similarities in exam questions is essential for maintaining the integrity and quality of assessments.

Natural language processing, specifically the Sentence-BERT (SBERT) model, plays a crucial role in detecting semantic similarities in engineering exam questions. Repetitive or paraphrased questions often compromise exam quality, while traditional methods struggle to identify such variations. An advanced model trained on a comprehensive academic dataset improves the detection of similar questions, reduces redundancy, and enhances exam diversity. The approach offers significant benefits in academic settings by streamlining the exam creation process, minimizing human errors, and ensuring fairness in assessments. The following discussion explores the model's development and evaluation, emphasizing its impact on question paper design and academic integrity.

1.1 Problem Definition

In academic institutions, particularly in engineering education, the repetition of exam questions across different years can lead to compromised exam quality and fairness. This issue arises due to the manual and often error-prone process of creating question papers, where similar or paraphrased questions may inadvertently be

*Corresponding Author

included. Existing tools for plagiarism detection are inadequate in identifying subtle similarities in academic questions, especially when specialized vocabulary or technical terms are used.

The challenge lies in developing an efficient, automated system capable of detecting semantic and lexical similarities between exam questions. Traditional methods, such as keyword matching, fail to address the complexities of paraphrasing and rewriting in academic contexts. Therefore, there is a need for a robust NLP-based solution that can identify question repetition, assist educators in designing diverse question papers, and uphold academic integrity. This paper aims to solve this problem by applying the sentence transformer model (SBERT) to detect semantic similarities in engineering exam questions, providing an advanced tool for educators to ensure the quality and diversity of exam content.

2. Literature Review

In Mathematical Structures of Language (1951/1968), a mathematical framework was introduced to analyze the distribution of words in a text, which paved the way for computational methods in language processing. This was further developed in A Neural Probabilistic Language Model (2003), which introduced a neural network-based model capable of capturing long-term dependencies in sentences, outperforming traditional n-gram models. Building on this, Natural Language Processing (Almost) from Scratch (2011) proposed a neural network-based approach that learned word representations directly from raw text, without the need for extensive feature engineering, marking a significant step in the rise of deep learning in NLP.

The Word2Vec model (2013) advanced the field by introducing efficient word vector representations through the Continuous Bag-of-Words (CBOW) and Skip-Gram models, capturing semantic relationships between words and improving NLP task performance. This was followed by Understanding LSTM Networks (2015), which explored Long Short-Term Memory (LSTM) networks, known for their ability to capture long-term dependencies in sequential data, essential for tasks in NLP. In Attention is All You Need (2017), the Transformer architecture was introduced, a novel model based entirely on attention mechanisms, which improved performance in handling long-range dependencies, allowed parallelization, and reduced training times compared to previous models.

The introduction of ELMo (2018), a deeply contextualized word representation, allowed for context-dependent embeddings that significantly improved performance across various NLP tasks by better capturing complex linguistic phenomena like polysemy and syntax. That same year, BERT (2018) revolutionized NLP by introducing a transformer-based model that utilized bidirectional pre-training on massive corpora of unlabeled text, fine-tuned for various downstream tasks, and achieved state-of-the-art results across multiple NLP benchmarks.

Further advancements in sentence-level embeddings were made with Sentence-BERT (2019), which used a Siamese network to generate efficient sentence embeddings, significantly enhancing tasks like semantic similarity, clustering, and information retrieval. In COBERT (2021), a fine-tuned version of DistilBERT was developed for COVID-19-related question answering, outperforming previous models in this domain with high Exact Match (EM) and F1 scores. Another significant contribution to information retrieval was made with FAQ Retrieval using BERT-based query-answer Relevance, which improved the accuracy of FAQ retrieval using BERT-based query-answer relevance scoring.

3. Methodology

3.1. Data Preparation

Data was sourced from a variety of publicly available datasets hosted on Hugging Face, including AllNLI (All Natural Language Inference), a collection encompassing MNLI (Multi-Genre Natural Language Inference), SNLI (Stanford Natural Language Inference), and CoLA (Corpus of Linguistic Acceptability). These datasets provide diverse resources for training and evaluating Natural Language Inference (NLI) models. Additionally, the MRPC (Microsoft Research Paraphrase Corpus) dataset supported training and evaluation for paraphrase identification, enabling the detection of sentences with equivalent meanings. The PAWS (Paraphrase Adversaries from Word Scrambling) dataset contributed further by focusing on paraphrases generated through word scrambling. To enhance the model’s ability to assess textual similarity, the STS (Semantic Textual Similarity) dataset was incorporated into the training and evaluation processes.

The primary dataset was collected from Pokhara University's past question papers. Using Google Docs and Google Drive API, exam questions were extracted from their digital format for preprocessing and model training.

3.2. Data Cleaning and Model Training

The data retrieved using the Google Docs and Google Drive API was raw and contained irrelevant information such as marks, headings, and noisy data due to improper JSON parsing. To clean the dataset, regular expressions (regex) were applied, followed by manual corrections to remove mathematical questions, fix instances where a single question was split across multiple rows, and add subject and year details to enhance contextual information. These steps ensured a properly structured dataset suitable for further processing. The primary dataset was not directly suitable for training a sentence transformer, as it required supervised conditions. To address this, the text-davinci-003 language model from OpenAI was used to generate paraphrased versions of each question. This approach doubled the dataset size to over 9000 rows, making it more robust for training paraphrase identification and semantic similarity models.

The preprocessing phase involved tokenization using the tokenizer of a selected distilbert-base-uncased, followed by padding and truncation to ensure uniform input size. A transformer-based architecture, such as BERT (Bidirectional Encoder Representations from Transformers) or RoBERTa (Robustly Optimized BERT Pretraining Approach), was fine-tuned separately on each dataset. The model’s performance was evaluated using key metrics, including accuracy, F1-score, precision, and recall.

3.2.1. Multiple Negative Ranking (MNR) Loss

To optimize semantic similarity learning, we integrate Multiple Negative Ranking (MNR) Loss, which refines SBERT embeddings for retrieval tasks. For each input batch of sentence pairs (a_i, p_i), where a_i is the anchor and p_i is the positive (similar) sample, the loss function treats all others p_i $(j \neq i)$ as negative samples. The objective function is formulated as:

$$L(x, y, \theta) \; = \; -\frac{1}{K}\sum_{i=1}^{K}[S(x_i, y_i) - log\sum_{j=1}^{K} e^{S(x_i, y_i)}] \qquad \text{Equation (1)}$$

where:

- K is the batch size,
- $S(x, y)$ represents the similarity score,
- x_i, y_i are positive pairs, while $x_i,\ y_i\ (i\ \neq j)$ are negative pairs,
- θ represents the model’s parameters.

This loss function ensures the model effectively differentiates between similar and dissimilar pairs, leading to more accurate embedding representations for semantic search applications.

3.2.2. Spearman Correlation

To evaluate the semantic similarity between questions, we used the Spearman correlation coefficient. The Spearman correlation coefficient is a non-parametric measure of rank correlation, assessing how well the relationship between two variables can be described using a monotonic function. The formula for the Spearman correlation coefficient ρ is given by:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} {d_i}^2}{n\,(n^2-1)} \qquad \text{Equation (2)}$$

3.2.3. Pearson Correlation

The Pearson correlation coefficient measures the linear relationship between two variables. It is defined as the covariance of the two variables divided by the product of their standard deviations. The formula for the Pearson correlation coefficient r between two variables X and Y is:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}} \qquad \text{Equation (3)}$$

3.2.4. Model Training Details

The model was trained using the following parameters:

- Optimizer: Adam optimizer was used with default parameters
- Batch Size: The batch size was set to 256
- Max Sequence Length: The max sequence length was set to 512
- Epochs: The model was trained for 1 epoch.
- Loss Function: MNR Loss was used to optimize the model for semantic similarity tasks.

3.3. System Block Diagram

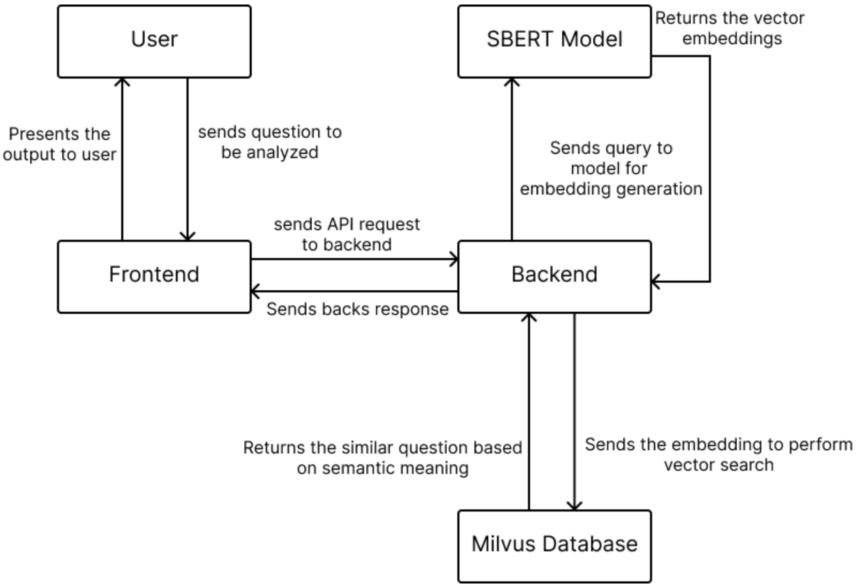
User
SBERT Model
Returns the vector embeddings
Presents the output to user
sends question to be analyzed
Sends query to model for embedding generation
sends API request to backend
Frontend
Backend
Sends backs response
Returns the similar question based on semantic meaning
Sends the embedding to perform vector search
Milvus Database

Figure 1. System Block Diagram

Figure 1 illustrates the workflow of a system leveraging SBERT (Sentence-BERT) for semantic similarity detection. The process begins when a user submits a query through the Frontend, which transmits the request to the Backend via an API. The Backend utilizes SBERT to generate vector embeddings of the input query, effectively capturing its semantic meaning. These embeddings are then sent to Milvus, a high-performance vector database, which performs a similarity search by comparing the query's embedding with stored embeddings. Milvus retrieves the most semantically relevant question and returns the result to the user, ensuring accurate and contextually relevant responses.

3.3.1. Inverted File Index (IVF)

We utilize the Inverted File Index (IVF) to enhance document retrieval efficiency. This indexing method structures data as a dictionary mapping unique terms to document identifiers, allowing for fast and scalable searches in large datasets. Given a query term t, the system retrieves the corresponding document list D_t from the index, enabling efficient lookup:

$$D_t = \{d_1, d_2, \ldots, d_n\} \text{ } where \text{ } t \epsilon d_i \qquad \text{Equation (4)}$$

where D_t represents the set of documents containing the term t_i, and d_i denotes individual documents. This approach significantly improves retrieval speed and accuracy, making it ideal for large-scale search applications.

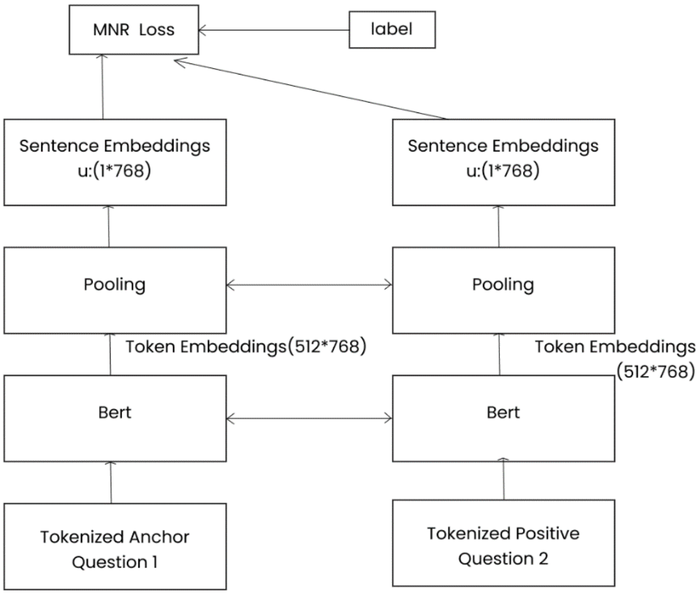
MNR Loss
label
Sentence Embeddings u:(1*768)
Sentence Embeddings u:(1*768)
Pooling
Pooling
Token Embeddings(512*768)
Token Embeddings (512*768)
Bert
Bert
Tokenized Anchor Question 1
Tokenized Positive Question 2

Figure 2. System Architecture Diagram (Modeling)

Figure 2 illustrates the training process of SBERT using Multiple Negative Ranking (MNR) Loss. During training, two BERT models process an anchor question along with a corresponding positive (semantically similar) question. Each model generates token-level embeddings with dimensions of 512 tokens × 768. These token embeddings then undergo a pooling operation, such as mean pooling, to produce sentence embeddings of size 1 × 768. The MNR Loss function compares the anchor and positive question embeddings while simultaneously incorporating negative (dissimilar) samples. This approach enhances the model's ability to distinguish between semantically similar and dissimilar pairs, thereby improving its effectiveness in capturing semantic similarity.

4. Result and Analysis

4.1. Results

The model was initially fine-tuned on the AllNLI dataset for one epoch, transforming DistilBERT-base-uncased into an SBERT model.

Table 1. Summary of Model Performance Metrics Across Datasets and Fine-Tuning Stages

Dataset	Cosine	Euclidean	Manhattan	Dot	Accuracy (MRPC)
AllNLI (1 epoch)	—	0.8373	0.8373	0.7952	—
MRPC (1 epoch)	0.8328	—	0.8325	0.7925	—
PAWS	0.8696	0.8566	—	0.8563	—
STS-B	0.8691	0.8685	0.8678	—	—
Custom (Eng. Exam questions)	—	0..8702	0.8695	—	88.7%

4.2. Analysis

The results demonstrate that transforming BERT into SBERT and systematically fine-tuning it on multiple datasets, including AllNLI, MRPC, PAWS, and STS-B, led to consistent improvements in performance. Each stage of fine-tuning allowed the model to develop a deeper understanding of sentence-level semantics by learning from diverse linguistic patterns and relationships. The inclusion of datasets covering tasks such as sentence-pair classification, paraphrase detection, and semantic similarity enhanced the model's generalization ability. The final stage of fine-tuning on a domain-specific dataset of engineering exam questions further refined the model, enabling it to capture nuanced similarities in question phrasing. The steady increase in Spearman correlation scores throughout the training process confirms that domain adaptation significantly improves performance, ensuring more accurate detection of semantically similar questions.

5. Conclusion

The Question Similarity Detection and Analysis model, developed using the SBERT framework, provides an effective solution for identifying duplicate questions in university assessments. By leveraging SBERT's advanced sentence embeddings, the model achieves superior semantic understanding compared to traditional methods. The fine-tuning process ensures adaptability to engineering domain-specific questions, enhancing the detection of redundant and recurring content. The improved semantic representation of exam questions contributes to maintaining assessment quality by reducing duplication and ensuring diverse and comprehensive evaluations. This approach holds significant potential for automating question validation processes in academic settings, ultimately preserving the integrity and uniqueness of university examinations.

References

Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C., 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3, pp.1137-1155. Available at: https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf [Accessed 2 April 2025].

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P., 2011. Natural language processing (almost) from scratch. Available at: https://arxiv.org/abs/1103.0398 [Accessed 2 April 2025].

Harris, Z.S., 1968. *Mathematical structures of languages*. New York: Wiley.

Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. Available at: https://arxiv.org/abs/1301.3781 [Accessed 2 April 2025].

Olah, C., 2015. Understanding LSTM networks. Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ [Accessed 2 April 2025].

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L., 2018. Deep contextualized word representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, *Volume 1 (Long Papers)*, pp.2227-2237. New Orleans, Louisiana: Association for Computational Linguistics. Available at: https://aclanthology.org/N18-1202/ [Accessed 2 April 2025].

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K., 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. Available at: https://arxiv.org/abs/1810.04805 [Accessed 2 April 2025].

Reimers, N. and Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Available at: https://arxiv.org/abs/1908.10084 [Accessed 2 April 2025].

Alzubi, J.A., Jain, R., Singh, A., Parwekar, P. and Gupta, M., 2021. COBERT: COVID-19 question answering system using BERT. *Arabian Journal for Science and Engineering*, 46(12), pp.12183-12193. Available at: https://link.springer.com/article/10.1007/s13369-021-05810-5 [Accessed 2 April 2025].

Sakata, W., Shibata, T., Tanaka, R. and Kurohashi, S., 2019. FAQ retrieval using query-question similarity and BERT-based query-answer relevance. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.1113-1116. New York: Association for Computing Machinery. Available at: https://dl.acm.org/doi/10.1145/3331184.3331304 [Accessed 2 April 2025].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention is all you need. Available at: https://arxiv.org/abs/1706.03762 [Accessed 2 April 2025].