International Journal of Mathematical, Engineering and Management Sciences
Vol. 10, No. 6, 1860-1879, 2025
https://doi.org/10.33889/IJMEMS.2025.10.6.086

Ram Arti Publishers

Multi-Transformer-Based Ensemble Embedding Model for Enhanced Vector Search in NoSQL Database: A Comparative Statistical and Performance Analysis

Narut Butploy
Department of Computer Technology,
Faculty of Industrial Technology, Kamphaeng Phet Rajabhat University, Kamphaeng Phet, Thailand.
Corresponding author: narut@kpru.ac.th

Kanokwan Khiewwan
Department of Computer Technology,
Faculty of Industrial Technology, Kamphaeng Phet Rajabhat University, Kamphaeng Phet, Thailand.
E-mail: kanokwan_kh@kpru.ac.th, kanokwan@live.kpru.ac.th

Jaturong Thongchai
Department of Computer Technology,
Faculty of Industrial Technology, Kamphaeng Phet Rajabhat University, Kamphaeng Phet, Thailand.
E-mail: jaturong_t@kpru.ac.th

Sawet Somnugpong
Department of Computer Technology,
Faculty of Industrial Technology, Kamphaeng Phet Rajabhat University, Kamphaeng Phet, Thailand.
E-mail: sawet_s@kpru.ac.th

Pakin Maneechot
Department of Smart Grid Engineering,
Faculty of Industrial Technology, Kamphaeng Phet Rajabhat University, Kamphaeng Phet, Thailand
E-mail: pakin_m@kpru.ac.th

Phrommate Verapan
Department of Information Technology,
Faculty of Science and Technology, Kamphaeng Phet Rajabhat University, Kamphaeng Phet, Thailand
E-mail: phrommate_v@kpru.ac.th

Khumphicha Tantisantisom
Department of Information Technology,
Faculty of Science and Technology, Kamphaeng Phet Rajabhat University, Kamphaeng Phet, Thailand.
E-mail: khumphicha_t@kpru.ac.th

Karthikeyan Velmurugan
Department of Technology Engineering,
Faculty of Industrial Technology, Kamphaeng Phet Rajabhat University, Kamphaeng Phet, 62000, Thailand.
E-mail: karthikeyan_v@kpru.ac.th

(Received on February 14, 2025; Revised on May 26, 2025; Accepted on June 29, 2025)

Abstract
Transformer-based embedding models are widely used for similarity search as they are reliable and efficient for capturing semantic similarity. This study uses all-MiniLM-L6-v2, paraphrase-MiniLM-L6-v2 and all-distilroberta-v1 transformer-based embedding models to find the similarity search for Wikipedia documents. All three transformer models are ensembled for enhanced semantic search, and Principal Component Analysis (PCA) is applied to ensure smooth assembly of a different dimensionality model. To understand the strength of the proposed transformer models, 2,000 Wikipedia documents were arbitrarily selected and converted into vectors before storing them in MongoDB. The ground truth of the proposed transformer-based models was examined using 996 TREC questions. The all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 consume less memory than all-distilroberta-v1 model. However, the ensemble process abruptly increased the memory usage to 924.79 MB, higher than individual models. Following that, the average execution time for each query increased to 0.1031 seconds. Beneficially, the ensemble+PCA attained higher precision@10 and recall, resulting in a higher F1 score with an average of 0.5094. The error analysis method indicates that the ensemble+PCA approach significantly improved the semantic search with a higher relevant rate to the raised query. Furthermore, ensemble-based PCA methods are recommended for large dataset handling and are suitable for real-time applications.

Keywords- Sentence transformer, Vector search, NoSQL Databases, Ensemble with PCA, Semantic search.

1. Introduction

In the 21st Century's Digital Age, internet usage has become essential for day-to-day life, from education to scientific development, job to business development, entertainment to life-saving events, and other purposes. There are several search engines in practice, but retrieving direct results is not applicable unless required to process a large set of search results to obtain accurate results. Conventional search engines process queries and find information through keyword-based retrieval. This method has to process a large set of datasets and is reliable for general searchings. Named entity recognition with WordNet was combined with the traditional search methods to improve the keyword-based search method. Furthermore, it expanded the search queries using query-oriented spreading activation. Though lexical knowledge improved, structured ontology and meaning expansion failed with scalability, limited generalisation, and slower query execution time (Ngo et al., 2010). Further, Named Entities (NEs), WordNet words (WWs), and latent concept discovery were introduced with the traditional keyword-based searching model to improve the semantic similarity. However, predefined ontologies struggled to scale, and the word sense disambiguation method failed to perform effectively for real-time applications (Ngo & Cao, 2018).

The traditional keyword-based information retrieval was time-consuming, lacking scalability and limited generalisation. When searching for information from a limited and structured dataset like Wikipedia, encyclopedia, and other databases, conventional search engines were ineffective in retrieving the results with higher accuracy (Chen et al., 2011; Pawar et al., 2016). In the last two decades, advancements in information retrieval models have led to the development of several approaches for structured datasets, with vector search-based datasets gaining popularity (Abualigah & Hanandeh, 2015; Gysel et al., 2018). A GPU-accelerated similarity search model is developed to perform with the high dimensional indexes. Integrating product quantisation and the GPU-optimized K selection algorithm improves k-nearest neighbor search operations. The developed model significantly improved the similarity search compared to other GPU models and achieved 8.5 times faster searching, such as processing 95 million images over 35 minutes and one billion vectors within 12 hours. However, it occupies a more significant memory for processing, a reliable model for large-scale dataset processing with a recall@1 of 0.4517 (Johnson et al., 2019). A semantic image retrieval based on the user's interest selection and assigned different weights to the images using interest-weighted summation (IWS) and interest-weighting (IW) methods are developed. The support vector machine sorts the photos by mapping their vector dimensions. Notably, the support vector machine transforms the images into higher dimensionality vectors and aligns linearly. The support vector machine independently lacks accuracy with an average of 0.78, whereas incorporating IWS with the support vector machine significantly increased the accuracy with an average of 0.85. However, IW attained lower accuracy

than IWS and the support vector machine. The IW method assigns lower weights for more minor features due to higher grid segmentation and sensitivity to over-segmentation, resulting in lower accuracy than IWS. The image-related retrieval models were not required to use the SQL platform as it was complicated to handle high-dimensional vector data. Further in their study, NoSQL-based vector search models are reviewed to assess the importance of similarity findings for higher vector dimensionality. NoSQL databases offer greater scalability, flexibility, and optimised indexing compared to SQL databases (Hu et al., 2022).

In recent years, vector search models have gained popularity due to their ability to handle high-dimensional vector data and their effectiveness in real-time applications such as similarity search, AI-driven tasks, and chatbots. The system transformed images, texts, and audio into a numerical structure and stored them in a database for query retrieval. Vector database management systems widely handle high-dimensional vector data for fast retrieval and similarity search. The system stores the transformed numerical data in a NoSQL database using unique indexing methods while arranging the vector data in an accessible sequence. The required information was retrieved using a cosine similarity and Euclidean distance. The faster response to query retrieval was the activation of fast nearest-neighbour searches. Additionally, vector databases are more effective for transforming, storing, and retrieving information from unstructured data (Taipalus, 2024).

NoSQL platform for independent query searching, transforming natural language into structured queries. They also designed intelligent search engines to process queries and segment them into seven major categories. The query autocompletion model uses term frequency-inverse document frequency (TF-IDF) suggestions to provide query autosuggestions. Then, text classification models were used to identify and classify the queries. Named entity recognition and entity mapping models were used to assemble the queries before the query generation model. During this process, the system transforms natural language queries into structured database queries and optimises them for auto-completion based on previous user queries. If the present query model fails to retrieve the information, the system automatically recommends an alternative model, resulting in the intelligent search engine achieving a higher accuracy of 93.6%, which surpasses other models (Kaur et al., 2024).

The system transforms a natural language query (NLQ) into a NoSQL query and employs Bidirectional Encoder Representations from Transformers (BERT) to interpret the natural language. It processes the NLQ, converts it into a structured NoSQL query, and utilises BERT to enhance natural language understanding. Deep learning techniques classify the databases, and the Levenshtein Distance Algorithm ensures the readiness of the queries. The NLP, BERT and query optimisation tools significantly improved the model's performance. The queries are preprocessed and attempted to find the difference between the NLQ and structured NoSQL queries. The training and proposed model accuracies were nearly similar, with an error rate of 11.24% (Hossen et al., 2023). The text to the ESQ transformation model is used to find vaccine adverse events from NoSQL databases. This study proposes a two-stage controllable model. The first module executes question-to-question transformation using BART, preprocessing the question into a reliable and standardised pattern to extract accurate information from NoSQL databases. Secondly, the Elasticsearch Query Condition Extraction (ECE), a combination of DistilBERT and BiLSTM models, was used to extract the key information value from the queries. It marks the findings into the ECE module, resulting in higher accuracy with a difference of 28.9 % as compared to the baseline model of Seq2Seq (Zhang et al., 2023).

The above literature and **Table 1** show that several techniques are in practice for finding semantic similarity using SQL and NoSQL. Most studies have failed to compare transformer-based embeddings in retrieving information and lack an approach to ensembling the models. Vector search methods have gained popularity

due to their contextual similarity match, which is lacking in conventional methods and mainly relies on keyword matching. However, the higher dimensionality of the vector data increases the query execution time and creates complexity in finding a similarity with large databases. On the other hand, limited explorations are observed on MiniLM-based models for text-to-vector transformation and query retrieval functions using MongoDB. Reportedly, computational efficiency and information retrieval from different queries are not examined statistically. Considering these research gaps, in this study, we have performed information retrieval from the large datasets of 2000 Wikipedia JSON files for 996 TREC questions. Each JSON file is converted into vector data using three large language models (LLM): all-MiniLM-L6-v2, paraphrase-MiniLM-L6-v2, and all-distilroberta-v1. The all-MiniLM-L6-v2 model is lightweight, has higher performance over a competitive operation, has fine-tuning capability, and has cosine similarity that favours improving the system's efficiency. Though the paraphrase-MiniLM-L6-v2 model performs on the MiniLM architecture, the paraphrasing capability is well suited for sentence matching and improved semantic similarity search for higher accuracy and faster operation. The all-distilroberta-v1 model performs under the RoBERTa Architecture, which differs from the MiniLM architecture. It performs effectively in language understanding and has larger embedding power than other models. Furthermore, to improve the accuracy of the retrieval process, all three models are ensembled, and PCA is applied to reduce the higher vector dimensions before storing them in MongoDB. The converted vector dimensions are presented in the histogram view to understand the vector dimensions distribution, central tendency, skewness, and data spread. A scatter plot analyses a high F1 score-TREC question from each LLM and ensemble model for an insightful view of the vector dimensions spread. Further, the performance metrics analysis, which includes Precision@10, Recall, MRR, F1 score, AET, and MU, is performed for 996 TREC questions to find the efficiency of the proposed LLM and ensemble models.

Table 1. Recent studies of text and image transformation to vector and comparative discussion with the proposed LLM.

Transformation	Database	Model / algorithm	Vector dimension/ feature size	Limitations compared to the proposed LLM	References
Image + text	Image dataset (SIMAT)	CLIP, FastText, LASER, and LaBSE embeddings with delta-vector transformation	Image: 512-D	Poor delta-vector transformation, highly sensitive for tuning, ineffective for cross-modal transfer, and limited feasibility for word-level examination. Less optimised for dense textual information retrieval and failed to reduce the dimensionality.	Couairon et al. (2022)
Text	TREC dataset	Neural Vector Space Model (NVSM), Latent Semantic Indexing (LSI), Deep learning-based IR	Document vectors: 64, 128, and 256-D, and word vectors: 300-D	It requires high storage capacity, is unsuitable for retrieval operations, is inefficient for weighing the performance metrics score, and has poor scalability. Unsupervised retrieval is ineffective for complex queries and is less robust due to the lack of an ensemble approach.	Gysel et al. (2018)
Text	Traditional text-based weighing	Probabilistic Latent Semantic Indexing (PLSI), Neural IR models	TF-IDF weights	Requires higher tuning, not effective for short queries and inaccuracy in information retrieval. Ineffective for semantic understanding and complex queries.	Anh & Moffat (2002)
Text	TREC/AP88-99 dataset	Word2Vec, GloVe, SBERT, Factor analysis, PCA	Word2Vec and GloVe: 300-D Sentence-BERT: 768-D	An effective semantic search requires higher training. Lack of correlation, purely based on variance and complexity in operation. Static embeddings are less effective than contextual models.	Brundha & Meera (2022)

Table 1 continued…

Text + image	Multimodal datasets	ComposeAE (Autoencoder-based), Deep metric learning, ResNet-17 for images, BERT for text	Image: 512-D, Text: 768-D	High tuning is required to perform longer queries. Complex in operation, not suitable for a real-time application. Higher dimensionality and focused on image retrieval. Text retrieval is ineffective.	Anwaar et al. (2021)
Text + knowledge graph	Graph databases	BERT, GPT-based rewriting, Knowledge graph embeddings	BERT: 768-D	Less accuracy, suitable for task-oriented operation, cost ineffective, unsuitable for retrieving the information, particularly for question-answer operation, and incapable of in-depth semantic search. The knowledge graph structure limits general document retrieval capabilities.	Wu et al. (2023)
Text	FAISS, approximate nearest neighbour search	Bi-encoder (DSSM, BERT, RoBERTa, ERNIE), T5-based encoder-decoder, Poly-encoder, ColBERT	BERT: 768-D	High computational time consumes higher memory usage, and fine-tuning is required for moderate semantic search capability. Higher dimensionality and complexity for real-time applications.	Zhao et al. (2024)
Text	Traditional document retrieval	TTRM (Target-oriented transformation networks), BiLSTM, CNN, Word2Vec, Context-Conserving Transformation (CCT)	Word2Vec: 300-D	Complex layer stacking is highly sensitive, and source data highly correlates with performance. Slower transformation process and no dimensionality reduction.	Wang et al. (2020)
Text	-	Vector Space Model (VSM) and Latent Semantic Indexing (LSI)	300-D	Lack of automation, moderate effectiveness for transformation-based query processing and regular supervision are required. Older vector-based methods are ineffective for semantic understanding	Dietrich et al. (2013)
Text	Dense vector retrieval	GPT-based embeddings	-	Complex architecture requires more resources for information retrieval, is weak in context and has low precision because of noise. Lack of multiple embedding models and scalability tests for large datasets.	Yang et al. (2024)
Text	Embedding vulnerabilities and retrieval security	GTR-base and openAI text-embeddings-ada-002	768-D and 256-D	Security threats are high and hackable, complex in reconstruction, and safe quantization of vector information was poorly handled. Ineffective optimising retrieval performance.	Zhuang et al. (2024)
Text	Vector database	LLM and event extraction	Sentence embedding models	It requires several steps of LLM preprocessing, high computational time, and merging of information retrieval, which is ineffective and lacks coherent text-blocking features. Lack of multiple retrieval models and reliance on single-sentence embedding models.	Tan et al. (2024)
Text	Wikipedia-based retrieval and vector database	Wikipedia-based retrieval augmentation, likely integrating a BERT-based transformer.	-	It follows several procedures, including external retrieval and reformulation, with a higher possibility of overlapping the information, keyword-based operation, and unpredictable performance due to irrelevant context handling. High storage requirements reduce efficiency.	Abdullahi et al. (2024)
Text	SQL storage	Word2Vec, GloVe, and pseudo-query vector estimation	Word2Vec and GloVe: 50, 100, 200 and 300-D	Ineffective for context-based information retrieval, requires strong queries to attain higher precision, queries dependent, highly sensitive to initiate the process and not robust in operation. Lack of real-world retrieval evaluation.	Zamani & Croft (2016)

Table 1 continued…

Image	Precomputed image vectors	Graph-based ANN indexes	96-D, 128-D, 200-D, 256-D, 960-D	Suitable for small and medium data processing, as it consumes higher memory usage. Complex to scale, ineffective recall and high cost for operation. Lack of semantic interpretation.	Azizi et al. (2025)
Text + image	LAION-5B	Inverted file with flat vectors	778-D	Slower access requires several filtering and preprocessing methods: high memory usage and cost-effectiveness. Lack of fast information retrieval.	Emanuilov & Dimov (2024)
Text + image	Metadata, SQL and tabular data	Chroma vector index and GTR embedding model	768-D	High dimensional and lacking in schema matching. A larger architect requires several resources and it is expensive. The complex is in operation due to the addition of the pipeline. Limitations of language and fine-tuning.	Ghali et al. (2025)
Text	Vector database	OpenAI, LLM and GPT4ALL	1536-D	Prompt tuning requires each query, which is inefficient for information retrieval on static data, and the operation and maintenance costs are comparatively higher. Due to real-time implementation issues, multimodal support has failed.	Pokhrel et al. (2025)
Text	IMAC-Mind: 4 large datasets368-Dan	SBERT and ADA algorithms	384-D and 1536-D	Requires tuning cosine similarity thresholds. Human interaction-based semantic search complications.	Gottfried et al. (2025)
Text	TREC-996 Question and NoSQL	all-MiniLM-L6-v2, paraphrase-MiniLM-L6-v2, all-distilroberta-v1 and ensemble model + PCA model	384, 384, 768 and 384-D	all-MiniLM-L6-v2: Efficient, fast, and effective for short text retrieval. paraphrase-MiniLM-L6-v2: Reduced query execution time with less storage requirement. Fine-tuning paraphrasing captures the nuance of semantic similarity. all-distilroberta-v1: A similarity screening process through a vast and complex architecture. Ensemble model + PCA: A unified model combines features from all transformer-based embedding models. The PCA approach reduces dimensions and enhances higher semantic similarity detection.	Present study

2. Transformer-Based Embedding Generation

Transformer-based embedding generation models were widely used to handle large and diverse datasets; through the sentence transformer library, they embed high-quality sentences and efficient models for retrieving information and checking semantic similarity.

2.1 All-MiniLM-L6-v2

The all-MiniLM-L6-v2 was a high-quality sentence embedding model mostly used for text-to-vector transformation. Initially, input texts were tokenised and processed for embedding to obtain efficient semantic similarity. The embedding process is performed through fewer transformation layers and matches the contextual differences between the tokens within the input text. Fewer transformer layers and fixed smaller embedding dimensions enhance computational efficiency, making the model suitable for real-time applications.

2.2 Paraphrase-MiniLM-L6-v2

The all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 models perform similarly since both belong to the transformer-based embedding platform and MiniLM architecture, specifically designed with lightweight transformer structures. However, the paraphrase-MiniLM-L6-v2 model contains paraphrase detection,

which favours eliminating the duplicate results and minor matches with the questions, whereas the all-MiniLM-L6-v2 model lacks fine-tuned paraphrase mining and finds the similarity through extensive document processing.

2.3 All-distilroberta-v1

The all-distrilroberta-v1 model differs in architecture from all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 but performs similar operations as they are all sentence transformers. Its deeper architectural structure enables similarity searches through a vast and complex contextual screening method. A complex contextual screening process increases the similarity searching times and requires high storage compared to the other two models. On the other hand, the larger architectural structure of the all-distrilroberta-v1 model requires higher computational resources to embed the data, which makes this model unsuitable for faster applications.

2.4 Ensemble + PCA Approach

To enhance embedding quality and address the limitations of the all-MiniLM-L6-v2, paraphrase-MiniLM-L6-v2 and all-distrilroberta-v1 models, this study implements an ensemble approach that combines the strengths of multiple sentence transformer models. The ensemble methods improve performance in various domains, including text-based tasks, by integrating diverse model outputs to enhance feature representation. However, all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 models' vector dimensions differ from all-distrilroberta-v1 models. Further, to improve the model diversity, PCA is applied to reduce the vector dimensions of the all-distilroberta-v1 model that contains 768-D, whereas the other two models have 384-D. To improve the performance of the vector search model and Ensemble of the three models, the vector dimensions of the all-distilroberta-v1 model are reduced by 50% and preserved about 90% of the variance. Though the principal components of all-distilroberta-v1 were reduced to 384, ensembling weight is effectively balanced with all the other models and aligned with a higher information retrieval rate. It provides a complementary strength by achieving efficient processing, in-depth paraphrasing locator, and precise understanding of the queries. The error corrections through ensembling increase document match with TREC queries and reduce over-fittings. The combination of features from all models significantly reduces dimensionality through PCA and enhances efficiency with faster computation.

2.5 Performance of Transformer-Based Embedding Model

This study uses vector search embedding models to analyse 996 TREC questions (provided in the supplementary section) and search for the similarity index in 2000 Wikipedia open sources. At first, a random 2000 Wikipedia documents are extracted in the JSON format and transformed into vector data using four different embedding models, namely, all-MiniLM-L6-v2, paraphrase-MiniLM-L6-v2, all-distilroberta-v1 and Ensemble Model with PCA. The reason for selecting random Wikipedia documents and TREC questions is to understand the robustness and stability of the proposed models. Under certain conditions, domain-oriented and custom selection could lead to higher F1 scores due to the preselection process. Random and diverse datasets are used to maintain the stability of the semantic search process. Considering the optimised cost operation, a lightweight LLM has been chosen to efficiently retrieve the response for raised queries with minimal hardware requirements, execution time, and computational load. Secondly, PCA minimises the vector dimensions, embedding sizes, and storage requirements. Data curation, tunning and customised preprocessing are minimised, making the proposed methods cost-effective and suitable for real-time applications. **Figure 1** shows a schematic view of the developed embedding model's similarity search in the vector search process. A custom Python script reads and parses the JSON files while embedding models convert the Wikipedia text (JSON files) into vector data. Further, the vector data are stored in MongoDB. Stored vector data contains the label of the JSON file for efficient similarity search, such as "title, id, URL, text". This structured format facilitates efficient data storage,

retrieval, and subsequent vectorisation, creating a robust foundation for advanced semantic search and analysis. Each TREC question was further expanded into five sub-questions, as shown in **Figure 2**, to find the similarity search in MongoDB. It favours evaluating the ground truth and performance of the four mentioned embedding models and the semantic search framework. These sub-questions are carefully linked to their corresponding main questions, enabling the development of a comprehensive ground truth evaluation. During this process, text-to-vector dimensions of 996 TREC questions for all models were analysed using histogram view as it can visualise the frequency of vector dimensions distribution across the TREC questions. Then, the system selects high F1-score TREC questions from each model and compares them with other models to analyse the relationship between dimension index and embedding values. Following that, performance analysis metrics, including Precision@10, Recall, F1-score, and Mean Reciprocal Rank (MRR) are used to evaluate the effectiveness of the transformer-based embedding model.

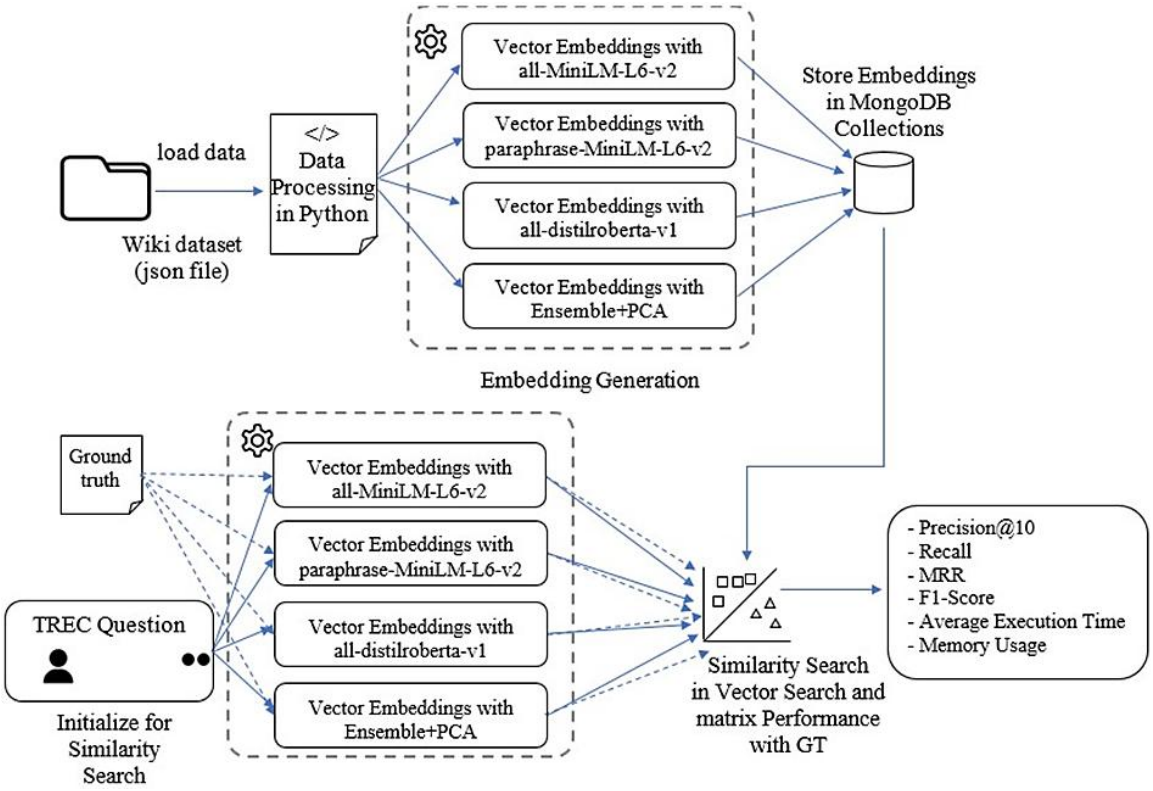

Figure 1. A schematic view of embedding models survivability architecture.

```
"What is considered the costliest disaster the insurance industry has ever faced ?": [
    "2011 Icelandic financial crisis",
    "1998 Atlantic hurricane season",
    "1992 Atlantic hurricane season",
    "1954 Atlantic hurricane season",
    "1955 Atlantic hurricane season"
],
```

Figure 2. An expansion of a TREC question into five sub-questions.

2.5.1 Precision@10

Precision@10 is an evaluation metric for the raised TREC query, which is the ratio of the documents that have the top ten high similarities to the total number of retrieved documents as expressed in Equation (1). The precision@10 evaluation metric is used based on the real-time search engine models such as the top ten relevant information for the query search. Similarly, in this study, precision@10 was adopted to measure the weight of the query's accuracy.

$$Precision@10 = \frac{Number\ of\ Relevant\ Retrieved\ Documents}{Total\ Number\ of\ Retrieved\ Documents} \quad (1)$$

2.5.2 Recall

Recall evaluates completeness and balances precision in retrieval systems, ensuring efficient information retrieval without eliminating critical data. It is the ratio of relevant documents found in the retrieved documents to the number of relevant documents available in the dataset, as expressed in Equation (2).

$$Recall = \frac{Number\ of\ Relevant\ Retrieved\ Documents}{Total\ Number\ of\ Relevant\ Documents\ in\ Ground\ Truth} \quad (2)$$

2.5.3 F1 Score

The F1 score is the primary parameter in measuring the performance metrics that balance precision and recall. A single metrics result concludes the efficiency of the retrieval information with a combined process of precision and recall as expressed in Equation (3). The F1 score depicts the balanced precision and recall, imbalanced data effectively handled, and the information is retrieved effectively.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

2.5.4 Mean Reciprocal Rank

The mean reciprocal rank (MRR) measures the effectiveness of the ranking system as expressed in Equation (4). The MRR catches the first relevant results retrieved and ranks the process, and it applies to real-time query information retrieval.

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{Rank_i} \quad (4)$$

3. Results and Discussion

3.1 Text to Vector Transformation

The system uses the transformer-based embedding model to convert 2000 JSON files into vector data. The total count of the vector dimensions was directly proportional to the embedding dimensions of the transformer-based embedding model. For example, the all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 models contain 382464 counts, as their embedding dimensions are 384, as shown in **Figures 3(a)** and **3(b)**. However, the all-distilroberta-v1 model attained 764928 counts, two times higher than the other two transformer-based embedding models, as shown in **Figure 3(c)**. The mean values of the embedding vectors for all the transformer-based embedding models are around zero, which indicates that the obtained text-to-vector transformation was efficient. However, the Ensemble + PCA model shows a slight variation due to dimensionality reduction. Combining the three transformer-based embedding models increased the embedding weights, resulting in positive vector dimensions and higher variance in amplifying the dimensions, as shown in **Figure 3(d)**. Secondly, the highest and lowest data spreads were observed for paraphrase-MiniLM-L6-v2 and all-distilroberta-v1 models with a deviation of 0.357 and 0.036, respectively, as listed in **Table 2**. The fine-tuning and capturing of a semantic similarity over a more extensive document/sentence causes an increase in the variance over vector dimensions for the paraphrase-MiniLM-L6-v2 model. However, the all-distilroberta-v1 model embeds the cluster around the mean and attains less variance, with a minimum and maximum vector dimension of -0.18 and 0.20, respectively. The

all-MiniLM-L6-v2 model skewness was -0.0083, which represents a nearly symmetric model. The values are moderately above the mean value, resulting in a slightly left skew. The paraphrase-MiniLM-L6-v2 model also attained a slightly left skew with a range of -0.0117. The all-distilroberta-v1 model attained a slightly right skew, which differs from the other two models but is nearly symmetrical. When combining all three transformer-based embedding models, skewness becomes asymmetric due to low kurtosis. During the dimensionality reduction, the stretched values are positive. Notably, they had nearly normal kurtosis and recorded the highest kurtosis for paraphrase-MiniLM-L6-v2 and all-distilroberta-v1 models. The lowest was for the Ensemble + PCA due to flatter distribution/limited outliers. The Ensemble model with PCA combines various tasks to refine the error and noise during the transformation process.

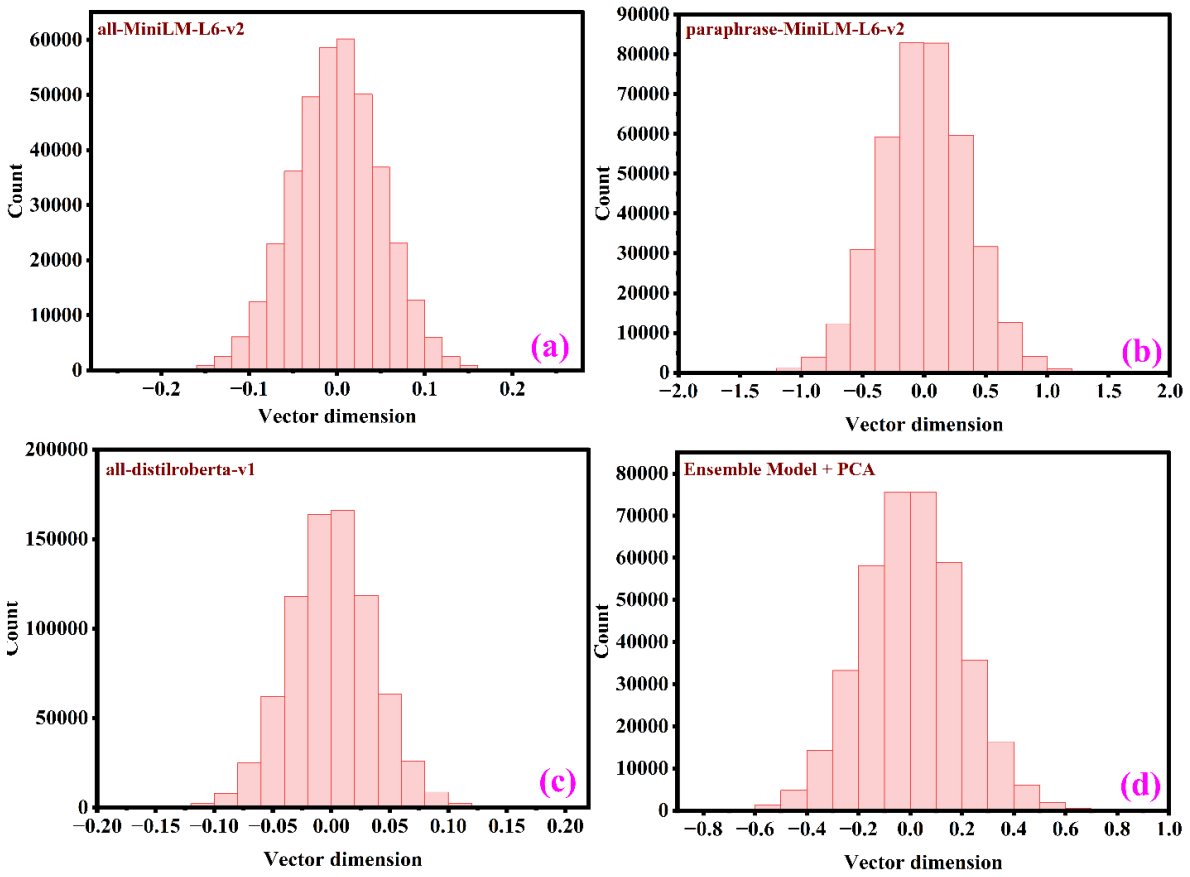

Figure 3. Histogram view of vector dimension for (a) all-MiniLM-L6-v2 model, (b) paraphrase-MiniLM-L6-v2, (c) all-distilroberta-v1 and (d) ensemble model with PCA.

Table 2. Transformer-based embedding models statistical analysis for 2000 Wikipedia documents vector dimension.

Model	Count	Mean	SD	Min	Max	25%	50%	75%	Skewness	Kurtosis
all-MiniLM	382464	0.00022	0.051	-0.277	0.263	-0.034	0.0025	0.034	-0.008	0.074
Paraphrase-MiniLM	382464	0.00184	0.357	-1.811	1.891	-0.234	0.0017	0.238	-0.012	0.202
All-distilroberta	764928	0.00034	0.036	-0.181	0.203	-0.023	0.0002	0.024	0.0146	0.238
Ensemble model with PCA	382464	0.00676	0.192	-0.852	0.942	-0.124	0.0048	0.136	0.057	0.031

3.2 Vector Dimension Distribution

High F1 score TREC questions are selected from each model and compared to others to understand the vector dimension distributions, as listed in **Table 3**. As mentioned, each transformer-based embedding model performs a unique semantic similarity search. For example, the TREC question "What is the wingspan of a condor?" scored a higher F1 score of 0.7 for an all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 model. Notably, the all-distilroberta-v1 model scored 0.3, which is lower than the other two models. However, combining all three models and applying PCA favours achieving a higher F1 score of 1, as listed in **Table 3**. Due to broader document screening, all-distilroberta-v1 model vector dimension ranges are scattered wider, as shown in **Figure 4(a)**. Higher variance in vector dimensions can include an irrelevant task while searching for semantic similarity. The vector dimensions are less aligned with the raised query and attained a lower F1 score. Using PCA, the Ensemble model removes the noise by combining moderate and highly scattered vector dimensions. When a query is raised, this emphasizes essential tasks, aligning vector dimensions into a narrow, more compressed platform. Furthermore, the system efficiently matches the raised queries to achieve a high F1 score.

Though the all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 models perform similarly, the question "What are the different approaches of systems analysis?" indicates that embedding the values from text-vector and vector-text transformations are truly independent for both models. An all-MiniLM-L6-v2 model attained a lower F1 score of 0.2, which shows clustered vector dimension embeddings around the zero. Due to this narrow and compact embedding formation, the all-MiniLM-L6-v2 model failed to capture the similarity. A paraphrase-MiniLM-L6-v2 model exhibits a wider vector dimension, and the embedding values are scattered around 0.5, as shown in **Figure 4(b).** They can capture higher nuanced semantic features than the all-MiniLM-L6-v2 model. When embedding values spread widely around the central mean, the system accelerates semantic features to enhance similarity matching. As mentioned earlier, all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 model operates in a similar transformation platform, but the fine-tuning ability in the paraphrase-MiniLM-L6-v2 model favours identifying the similarities over an extensive document processing as compared to the all-MiniLM-L6-v2 model. PCA-based ensemble model enlarges the vector dimensions from all-MiniLM-L6-v2 and all-distilroberta-v1 models into a wide scatter. However, combining a high F1 score of the paraphrase-MiniLM-L6-v2 model makes the process more manageable than the previous TREC questions because the F1 score is already 1 for a paraphrase-MiniLM-L6-v2 model. Overall, the paraphrase-MiniLM-L6-v2 model gained a higher F1 score than the others. Comparatively, all-distilroberta-v1 models gained lower F1 scores.

Table 3. High F1 score TREC questions for all transformer-based embedding models with ensemble model.

TREC question	F1 score			
	all-MiniLM-L6-v2	paraphrase-MiniLM-L6-v2	all-distilroberta-v1	Ensemble model with PCA
What is the wingspan of a condor?	0.7	0.7	0.3	1.0
What are the different approaches of systems analysis?	0.2	1	0.2	1.0
What's the sacred river of India?	0.23	0.7	0.47	0.82
How many community chest cards are there in Monopoly?	0.3	0.7	0.2	1.0

The all-distilroberta-v1 model attained the highest F1 score of 0.47 for the question, "What's the sacred river of India?". Notably, the all-MiniLM-L6-v2 model achieved a 0.23 F1 score, lower than the other two models and comparatively 0.24 and 0.47 lower than all-distilroberta-v1 and paraphrase-MiniLM-L6-v2 model. In this case, the paraphrase-MiniLM-L6-v2 model also gained a higher F1 score. The wide variation in the F1 score for all-distilroberta-v1 and all-MiniLM-L6-v2 models makes the ensemble model achieve a lower F1 score of 0.82, lower than the above-mentioned two TREC questions, as shown in **Figure 4(c)**.

The paraphrase-MiniLM-L6-v2 model gained a wider vector dimension, while the other two models attained narrow dimensions for the question, "How many Community Chest cards are there in Monopoly?" which exhibits a similar pattern as shown in **Figure 4(d)**. However, ensembling all three models favours attaining a high F1 score of 1. Overall, 9.43% of queries gained an F1 score of 1 out of 996 TREC questions using an ensemble model, whereas zero queries achieved an F1 score of 1 for all-distilroberta-v1 and all-MiniLM-L6-v2 models, and two queries achieved F1 score of 1 for the paraphrase-MiniLM-L6-v2 model.

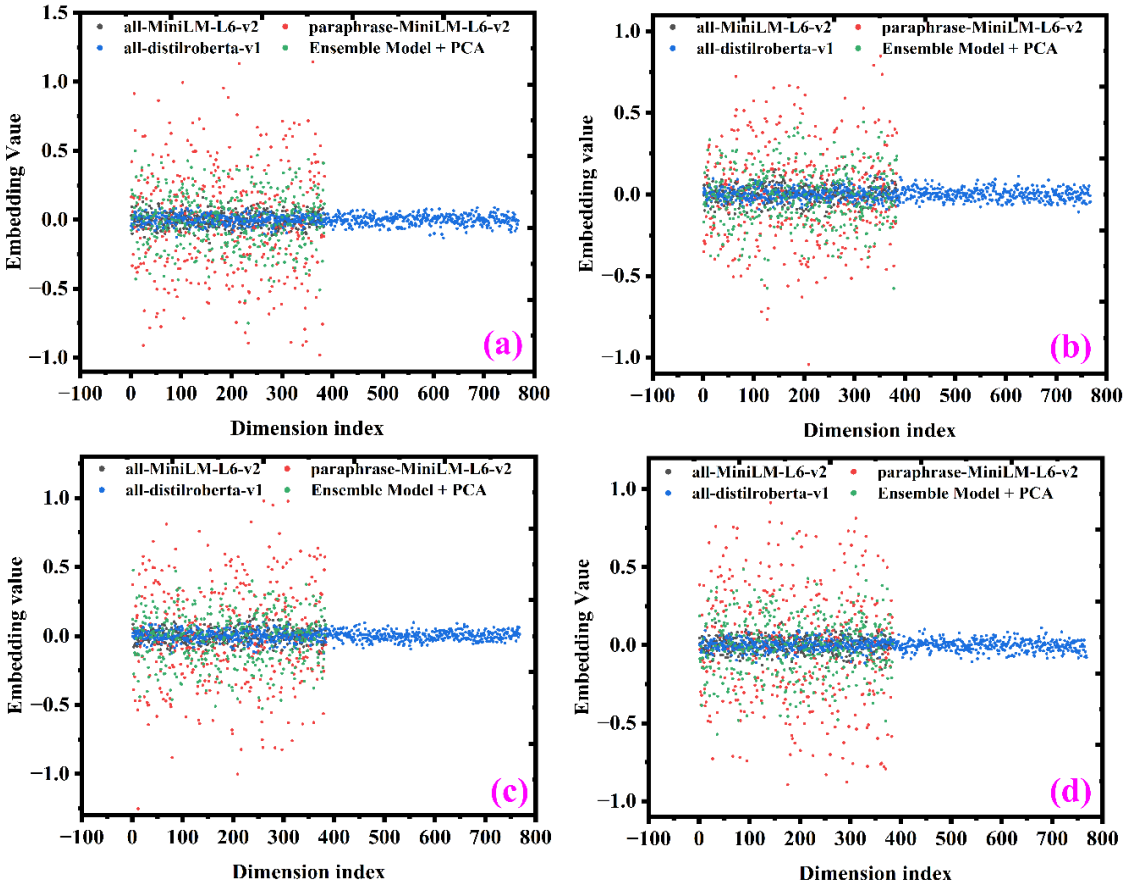

Figure 4. A scatter plot for high F1 score TREC question (a) all-MiniLM-L6-v2 model, (b) paraphrase-MiniLM-L6-v2, (c) all-distilroberta-v1 and (d) Ensemble Model + PCA.

3.3 Text-to-Vector Embedding Generation Process

The system stores the transformed text as vector indexes in a NoSQL database such as MongoDB. This approach enhances flexibility in storing vector data and document titles, URLs, and other descriptions necessary for semantic similarity search. MongoDB has a built-in vector search capability, including cosine similarity search with the stored vector data. Secondly, it is an efficient technique for sorting and filtering the metadata by searching the nearest neighbour search scheme. This study transforms 2000 JSON files into vector data and stores them in MongoDB. The all-distilroberta-v1 model gained a higher storage capacity of 38.5 MB due to increased embedding dimensions 768, as listed in **Table 4**. The paraphrase-MiniLM-L6-v2 model used the lowest storage capacity of 36.47 MB as this model contains high-magnitude components that favour serialising the vector data in MongoDB. The primary process of fine-tuning tasks

minimises the vector data repetitions and increases semantic density. When all three transformer-based embedding models are combined, storage capacity remains closer. Though all-distilroberta-v1 model vector dimensions are 768, ensembling the models with PCA compressed the dimensions into 384. It favours maintaining the storage capacity closer to all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 models. This reduced storage capacity of an ensemble model proves that the noises are effectively eliminated and compact in searching for semantic similarity.

Table 4. Text-to-vector transformation for transformer-based embedding models embedding dimensions and MongoDB storage capacity.

Model	Total documents processed	Embedding dimension	MongoDB storage capacity (MB)
all-MiniLM-L6-v2	2000	384	38.06
paraphrase-MiniLM-L6-v2	2000	384	36.47
all-distilroberta-v1	2000	768	38.5
Ensemble Model + PCA	2000	384	40.68

3.4 Semantic Similarity Analysis

The all-MiniLM-L6-v2 model gained a semantic similarity score between 0.56 and 0.78, and the TREC question frequency is plotted against the similarity score, as shown in **Figure 5(a)**. The peak similarity scores lie within the range of 0.62 and 0.68. The analysis shows that over 250 TREC questions attained a peak semantic similarity score, and over 600 questions achieved a similarity score higher than 0.60. It demonstrates that the all-MiniLM-L6-v2 model avoids overconfidence in assigning similarity matches. The mean and median values are 0.6363 and 0.6356, which reflects that the model is symmetrical. The histogram view reflects that the semantic similarity scores are slightly rightward-tailed with no strong skewness. Beneficially, a 384 vector dimension does not affect the searching fluctuation and is less sensitive to minor word changes. On the other hand, the cosine similarity function makes the semantic similarity into a dense structure. However, only three TREC questions have gained scores above 0.75, which concludes that the all-MiniLM-L6-v2 model gained second rank in the three transformer-based embedding models. Similarly, the mean and median values of the paraphrase-MiniLM-L6-v2 model scored 0.6734 and 0.6746, respectively. For the 996 TREC questions, semantic similarity scores attained a 0.0332 standard deviation. Comparatively, similarity scores are highly dense, such as below 0.60 for only 13 TREC questions, whereas the widespread all-MiniLM-L6-v2 model score was below 0.6 with 113 TREC questions, as shown in **Figure 5(b)**. Secondly, 11 questions attained a higher score of 0.75, which exceeded previous scores. As mentioned, a fine-tuned paraphrase detection catches the similarity and minimised document mismatching. The average similarity score is higher than the all-MiniLM-L6-v2 model. Smaller skewness and kurtosis were close to the normal distribution, meaning the semantic similarity scores' outliers were incredibly low. Although the all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 models operate under a similar transformer architecture, the paraphrase-MiniLM-L6-v2 model specialises in detecting paraphrases. When queries match, the model assigns higher similarity scores as the sentence embeddings are precisely optimised. Overall, the paraphrase-MiniLM-L6-v2 model attained a peak semantic similarity score of 0.65–0.72 for 695 TREC questions, whereas the all-MiniLM-L6-v2 model attained only 287 TREC questions. The all-distilroberta-v1 model has a median value slightly lower than the mean. However, it attains a semantic similarity score below 0.60 for 310 TREC questions, higher than the other two transformer-based embedding models. Similarly, 212 TREC questions attained a peak score of 0.65 – 0.72, which is also lower than that of other models, as shown in **Figure 5(c)**. Due to the higher vector dimensionality of this model, dramatic changes in the score occurred. Generally, the higher vector dimensionality captures finer semantic similarity differences. Because they are more nuanced, allowing the model to distinguish subtle variations in meaning and improve retrieval accuracy in complex queries. An increase in embedding spaces separates irrelevant vectors, resulting in lower similarity found over the extensive database. However, this model is

a more balanced approach to finding similarities than the all-MiniLM-L6-v2 model, although it does not attain a higher score. When three transformer-based embedding models are ensembled, mean and median values are nearly similar, such as 0.6684 and 0.6697, as shown in **Figure 5(d)**. A TREC question of 113, 13, and 310 from all three transformer-based embedding models attained a similarity score below 0.6. Ensembling all the models with PCA reduced to 12 questions, making the ensemble model more effective. The Ensemble + PCA model obtained a semantic similarity score between 0.65 and 0.72 for 675 TREC questions. However, this score is lower than the paraphrase-MiniLM-L6-v2 model, although the ensemble model achieves a higher similarity score. Because combining all three transformer-based embedding models operates under a single model behaviour. They capture more semantic features, and above all of these, PCA regulates the score by eliminating noises, redundancy, and other forms of error. Preserving the variance and ensuring the dimensionality reduction accelerates higher match for the queries and effectively operates under balanced mode. This ensemble model with PCA is a robust and highly efficient method for finding semantic similarity in vector search models.

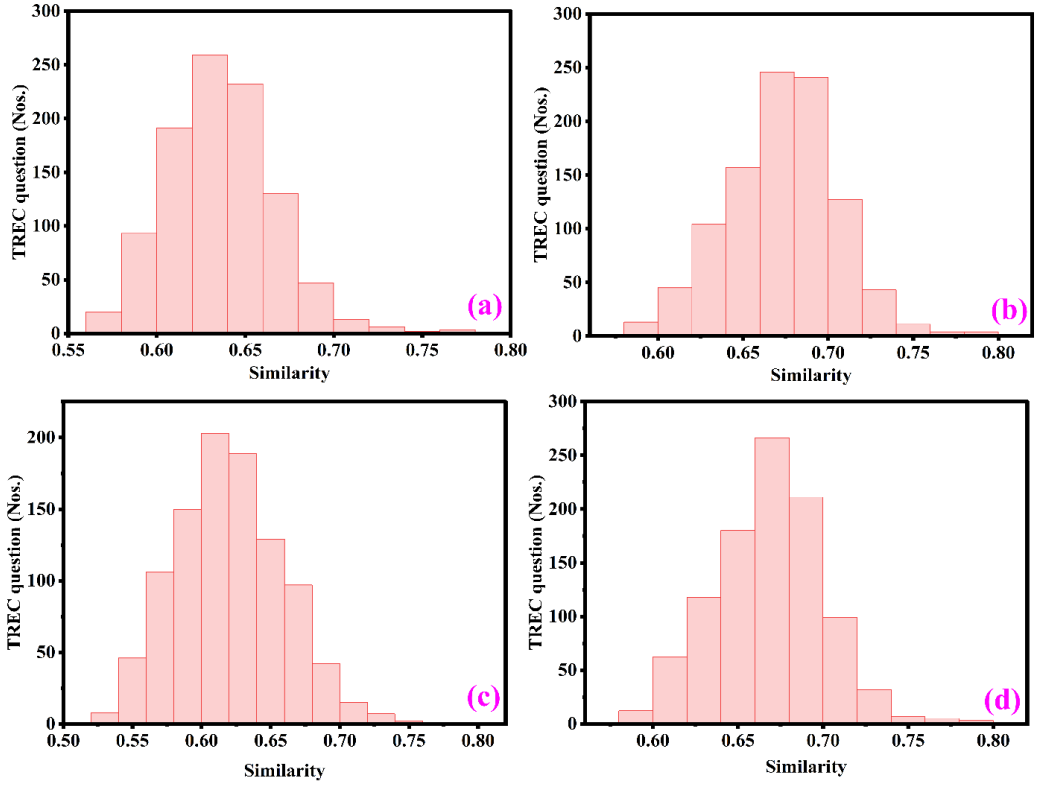

Figure 5. Semantic similarity frequency for 996 TREC questions using (a) all-MiniLM-L6-v2 model, (b) paraphrase-MiniLM-L6-v2, (c) all-distilroberta-v1, and (d) Ensemble Model + PCA.

3.5 Transformer-Based Embedding Models Metrics Analysis

Table 5 depicts the overall performance metrics of all the transformer-based embedding models with ensemble-based PCA models. The primary metrics of precision@10 analysis projected over the 10 retrieved results for queries raised. The all-MiniLM-L6-v2 model retrieves approximately 1.08 relevant results over the top 10 findings. This 0.1085 precision@10 is moderately relevant to the queries for analysing the ground truth. However, these moderate relevance score-based models are widely used for general-purpose tasks. Comparatively, the paraphrase-MiniLM-L6-v2 model scored an average of 3.92 relevant results for the

TREC queries. Fine-tuning and paraphrase detection match the queries with a relevant document over a wide spread of vector dimensions and catch highly similar results compared to the all-MiniLM-L6-v2 model. This results in paraphrase-MiniLM-L6-v2 models being recommended for paraphrase detection. In all the other two transformer-based embedding models, the all-distilroberta-v1 model gained the lowest precision@10 of 0.0729, approximately 0.73 relevant results over the top ten retrieved results. Combining all the transformer-based embedding models gained a precision@10 of 0.4320, showing their strength and best performance for the TREC queries. An increase in relevant results for Precision@10 directly affects recall, as higher precision tends to improve the matching results for TREC queries. This means the system successfully retrieves items, and their proportion is measured in terms of recall. Following precision@10, the ensemble model gained a higher recall of 0.6722, which depicts both ranking relevant items at the top and retrieving a large proportion of relevant items. Similarly, MRR is correlated with precision@10, which measures how early the relevant answers appear for the raised TREC queries and ranks the quality of the search. If the precision@10 is high, MRR follows a similar pattern because both are screening the perfect match for the questions and providing a rank and score for the results. The all-MiniLM-L6-v2 and all-distilroberta-v1 models emphasise lower MRR scores due to lower precision, making these models not recommended for individual semantic similarity searches. However, ensembling all the transformer-based embedding models significantly improved the MRR score to 0.5626.

The recall and MRR have a strong positive correlation with the precision@10. The F1 score is the primary performance analysis metric for vector search algorithms. The F1 score concludes the performance of the models under single-oriented and well-balanced relevance findings and effecting similarity. Due to its lack of balance, the all-distilroberta-v1 model achieved the lowest F1 score of 0.0877. Following that, the all-MiniLM-L6-v2 model recorded the second-lowest F1 score of 0.1274. For individual semantic similarity searches, these models are not recommended.

Comparatively, the paraphrase-MiniLM-L6-v2 model scored a high F1 due to high precision@10 and recall. This model is well-balanced, has a higher ranking, and is efficient for paraphrasing-based operations. Finally, combining all transformer-based embedding models significantly enhanced the overall F1 score, achieving an average of 0.5094. The high relevance results in over precision@10 and broader retrieval for search queries, which improved the F1 score. Precisely concluded that the ensemble model with a combination of PCA processes is well-balanced and efficient for paraphrasing and similar general-purpose tasks. A higher F1 score compiles with a higher average execution time and memory taken for processing the queries and finding the similarity search. In this case, the ensemble model consumes more time to execute and find a similarity for raised TREC queries with an average of 0.1031 seconds. Due to a complex combination of three transformer-based embedding models execution, the memory usage drastically increased to 924.79 MB, 17.53 times higher than a paraphrase-MiniLM-L6-v2 model. Comparatively, the t-SNE is widely used for high-dimensional data visualisation rather than indexing, failing to maintain the global distance scaling features (Kim et al., 2021). Due to the stochastic behaviour of UMAP, embedding reproducibility is limited, making it ineffective for indexing and the semantic similarity process (Pamuji et al., 2024). PCA handles millions of vectors using low memory resources and faster query response, is linear in operation, has fast indexing, and vector shapes are precisely maintained and reliable indexing compared to the t-SNE or UMAP. This method provides a richer, more contextually relevant representation of textual data, enabling complex query handling and nuanced semantic relationships. Due to linear transformation, PCA may have limitations in semantic embeddings when the variances are not in higher order, which could lead to non-Gaussian distributions. However, in this study, PCA retains 90% of the variance and performs stable operations over a low-dimensional vector to retrieve information from a large dataset and reduce computational resources. Overall, the ensemble model with PCA is recommended for a vector search algorithm using MongoDB.

Table 5. Performance metric analysis for transformer-based embedding models.

Model	Precision@10	Recall	MRR	F1 Score	AET (Sec.)	MU (MBRA)
all-MiniLM-L**6**-v**2**	0.1085	0.165	0.0633	0.1274	0.0341	59.0078
paraphrase-MiniLM-L**6**-v**2**	0.3919	0.6143	0.0984	0.4631	0.0180	52.7383
all-distilroberta-v**1**	0.0729	0.1190	0.0525	0.0877	0.0309	184.9258
Ensemble-model	0.4320	0.6722	0.5626	0.5094	0.1031	924.79

3.6 Comparative Error Analysis

Figure 6 shows an error analysis of three transformer models with an ensemble-based PCA approach to enhance the precision rate for the raised query "What is the wingspan of a condor?". As mentioned earlier, each transformer model has its unique architecture in semantic similarity check. The all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 model gained seven relevant matches to the raised queries; however, their relevant matches differ. Their fine-tuning capabilities and paraphrase-based search enable them to track higher relevant searches over the ten results. Comparatively, the all-distilroberta-v1 model struggled to match with a higher relevant rate due to increased vector dimensions. In this study, the F1 score is allocated based on the best matches of the Precision@10, resulting in the lowest F1 score. Though all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 models gained higher Precision@10, they failed to attain the unity in the F1 score. Ensembling the three transformer models with the PCA approach favours attaining higher Precision@10, and all the ten search estimation results are relevant to the queries raised, as shown in **Figure 6**. Further, it is recommended that the ensembling of the transformer-based semantic search models with PCA is necessary to attain a higher F1 score.

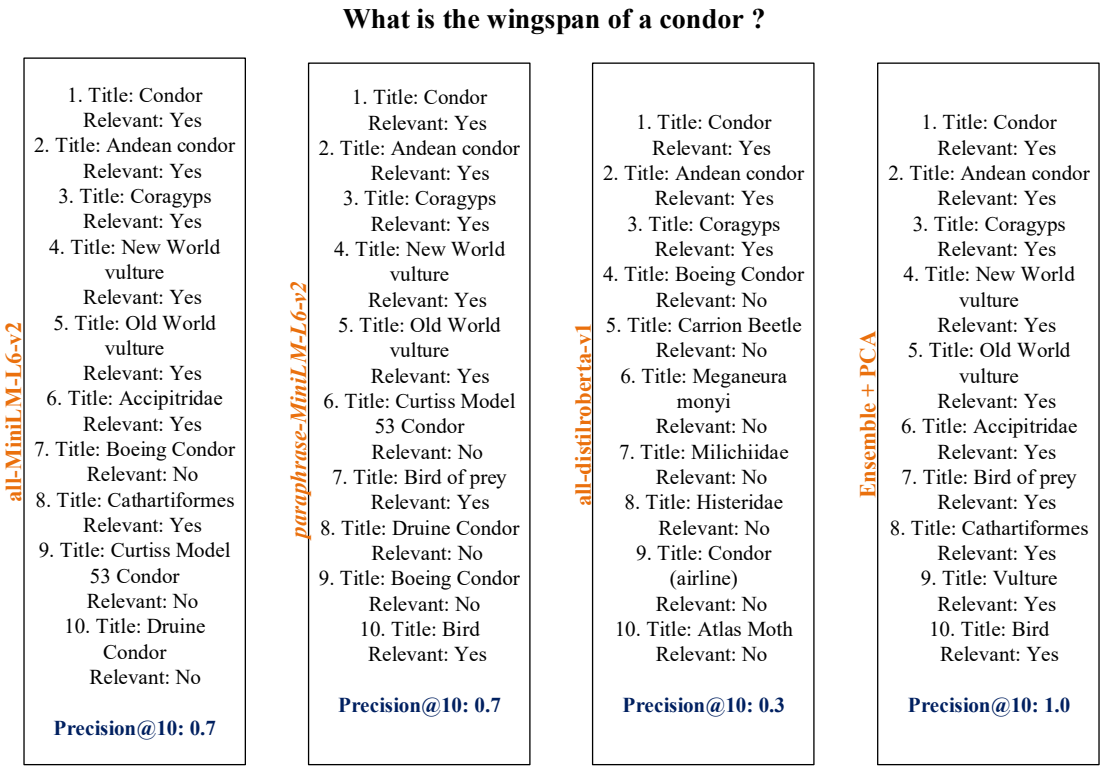

Figure 6. Error analysis for transformer models with ensemble Model + PCA approach.

3.7 Comparison of Proposed LLM with Other Models

The Proposed all-MiniLM-L6-v2, paraphrase-MiniLM-L6-v2 and all-distilroberta-v1 models have gained lower F1 scores due to the diverse nature of operations. Ensembling the three models with PCA significantly improves the F1 score with a range of 0.5094. The comparative study with traditional methods like Term Frequency-Inverse Document Frequency (TF-IDF) and Best Matching 25 (BM25) model exhibit lower F1 scores than the proposed three LLMs, as listed in **Table 6**. The average execution time for queries is lower due to ineffective context-understanding behaviour. Due to large sparse matrices, they consume high computational resources, resulting in higher memory usage than the other three LLM. Further, it is found that TF-IDF and BM25 are comparatively not recommended for a semantic similarity search.

Table 6. Comparative analysis of traditional models with proposed LLM.

Model	Precision@10	Recall	MRR	F1 Score	AET (Sec.)	MU (MBRA)
all-MiniLM-L**6**-v**2**	0.1085	0.165	0.0633	0.1274	0.0341	59.0078
paraphrase-MiniLM-L**6**-v**2**	0.3919	0.6143	0.0984	0.4631	0.0180	52.7383
all-distilroberta-v**1**	0.0729	0.1190	0.0525	0.0877	0.0309	184.9258
Ensemble + PCA Model	0.4320	0.6722	0.5626	0.5094	0.1031	924.79
TF-IDF	0.0190	0.0319	0.0145	0.0229	0.0403	3838.2227
BM25	0.0355	0.0585	0.1105	0.0427	0.0082	1464.18

4. Conclusion

This study performs a text-to-vector transformation for 2000 Wikipedia JSON files and stores them in MongoDB. Three transformer-based embedding models were used for vector transformation due to their unique features: fewer transformer layers, a fine-tuned paraphrase detection process, and complex contextual screening methods. An all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 model has a smaller dimension of 384, and all-distilroberta-v1 has a higher dimension of 768. Though all-MiniLM-L6-v2 has a lower dimensionality, vector embedding values are narrow to zero, failing to retrieve the information effectively. A fine-tuning paraphrase detection in the paraphrase-MiniLM-L6-v2 model screens the broader documentation search with higher nuanced semantic features. Resulting in a wider embedding values dispersion. Notably, all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2 model has a smaller dimensionality, but the paraphrase-MiniLM-L6-v2 model is equipped with a higher order of semantic similarity features detection. On the other hand, the all-distilroberta-v1 model with a complex architectural structure failed to retrieve the information effectively compared to the other models. Though the query execution time for the all-distilroberta-v1 model is lower than the all-MiniLM-L6-v2 model, it failed to achieve a high F1 score. Combining all three models with the PCA significantly improved the information retrieval, resulting in an average F1 score of 0.5094. The findings concluded that an ensemble model with PCA was suitable for large-scale dataset processing and proved efficient for real-time applications. The proposed method can be effectively utilised in various real-time applications such as academic research article searching platforms and digital libraries. Under the extensive mode of operation, it can be used to retrieve patient information from the hospital's registry, perform product searches across the e-commerce domain, and use smart devices to understand queries effectively. Further, custom rule-based advanced filtering methods are recommended to reduce complexity in MiniLM models. Transforming the long texts into clauses can significantly improve the similarity search for the paraphrase-MiniLM-L6-v2 model. The all-distilroberta-v1 model should performed for complex queries rather than general queries examination.

Conflict of Interest
The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments
The authors thank the Department of Computer Technology, Faculty of Industrial Technology, Kampaeng Phet Rajabhat University, Kampaeng Phet Thailand, for providing the technical and lab facilities to complete this research. This research received no external funding.

AI Disclosure
The author(s) declare that no assistance is taken from generative AI to write this article.

References

Abdullahi, T., Singh, R., & Eickhoff, C. (2024). Retrieval augmented zero-shot text classification. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval* (pp. 195-203). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/3664190.3672514.

Abualigah, L.M.Q., & Hanandeh, E.S. (2015). Applying genetic algorithms to information retrieval using vector space model. *International Journal of Computer Science, Engineering and Applications, 5*(1), 19-28. https://ssrn.com/abstract=3872452.

Anh, V.N., & Moffat, A. (2002). Impact transformation: effective and efficient web retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 3-10). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/564376.564380.

Anwaar, M.U., Labintcev, E., & Kleinsteuber, M. (2021). Compositional learning of image-text query for image retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1140-1149). IEEE. Waikoloa, US.

Azizi, I., Echihabi, K., & Palpanas, T. (2025). Graph-based vector search: an experimental evaluation of the state-of-the-art. *Proceedings of the ACM on Management of Data*, *3*(1), 1-31. https://doi.org/10.1145/3709693.

Brundha, J., & Meera, K.N. (2022). Vector model based information retrieval system with word embedding transformation. In *2022 10th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing* (pp. 01-04). IEEE. Nagpur, India. https://doi.org/10.1109/icetet-sip-2254415.2022.9791503.

Chen, Y., Wang, W., & Liu, Z. (2011). Keyword-based search and exploration on databases. In *2011 IEEE 27th International Conference on Data Engineering* (pp. 1380-1383). IEEE. Hannover, Germany. https://doi.org/10.1109/icde.2011.5767958.

Couairon, G., Douze, M., Cord, M., & Schwenk, H. (2022). Embedding arithmetic of multimodal queries for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4950-4958). IEEE Computer Society. New Orleans, US.

Dietrich, T., Cleland-Huang, J., & Shin, Y. (2013). Learning effective query transformations for enhanced requirements trace retrieval. In *2013 28th IEEE/ACM International Conference on Automated Software Engineering* (pp. 586-591). IEEE. Silicon Valley, CA, USA. https://doi.org/10.1109/ase.2013.6693117.

Emanuilov, S., & Dimov, A. (2024). Billion-scale similarity search using a hybrid indexing approach with advanced filtering. *Cybernetics and Information Technologies, 24*(4), 45-58. https://doi.org/10.2478/cait-2024-0035.

Ghali, M.K., Farrag, A., Won, D., & Jin, Y. (2025). Enhancing knowledge retrieval with in-context learning and semantic search through generative AI. *Knowledge-Based Systems*, *311*, 113047. https://doi.org/10.1016/j.knosys.2025.113047.

Gottfried, K., Janson, K., Holz, N.E., Reis, O., Kornhuber, J., Eichler, A., Banaschewski, T., Nees, F., & IMAC-Mind Consortium. (2025). Semantic search helper: a tool based on the use of embeddings in multi-item questionnaires as a harmonization opportunity for merging large datasets-a feasibility study. *European Psychiatry*, *68*(1), e8. https://doi.org/10.1192/j.eurpsy.2024.1808.

Gysel, C.V., De Rijke, M., & Kanoulas, E. (2018). Neural vector spaces for unsupervised information retrieval. *ACM Transactions on Information Systems*, *36*(4), 1-25. https://doi.org/10.1145/3196826.

Hossen, K.M., Uddin, M.N., Arefin, M., & Uddin, M.A. (2023). Bert model-based natural language to nosql query conversion using deep learning approach. *International Journal of Advanced Computer Science and Applications, 14*(2), 810-821. https://doi.org/10779/dro/du:26095237.v2.

Hu, W., Sheng, Y., & Zhu, X. (2022). A semantic image retrieval method based on interest selection. *Wireless Communications and Mobile Computing*, *2022*(1), 1-6. https://doi.org/10.1155/2022/3029866.

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, *7*(3), 535-547. https://doi.org/10.1109/tbdata.2019.2921572.

Kaur, G., Agrawal, P., Shelar, H., & Abraha, G.T. (2024). Intelligent search engine tool for querying database systems. *International Journal of Mathematical, Engineering & Management Sciences*, *9*(4), 914-930. https://doi.org/10.33889/ijmems.2024.9.4.048.

Kim, J.H., Jung, S.H., & Hwang, U.J. (2021). The research trends and keywords modeling of shoulder rehabilitation using the text-mining technique. *Journal of The Korean Society of Physical Medicine*, *16*(2), 91-100. https://doi.org/10.13066/kspm.2021.16.2.91.

Ngo, V.M., & Cao, T.H. (2018). Discovering latent concepts and exploiting ontological features for semantic text search. *Information Retrieval*. arXiv preprint arXiv:1807.05578.

Ngo, V.M., Cao, T.H., & Le, T.M.V. (2010). Combining named entities with wordnet and using query-oriented spreading activation for semantic text search. In *2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future* (pp. 1-6). IEEE. Hanoi, Vietnam. https://doi.org/10.1109/rivf.2010.5633401.

Pamuji, A.Z., Fatichah, C., & Navastara, D.A. (2024). Deep learning-based topic modeling for apex legends user reviews. In *2024 Ninth International Conference on Informatics and Computing* (pp. 1-6). IEEE. Medan, Indonesia. https://doi.org/10.1109/icic64337.2024.10957448.

Pokhrel, S., KC, B., & Shah, P.B. (2025). A practical application of retrieval-augmented generation for website-based chatbots: combining web scraping, vectorization, and semantic search. *Journal of Trends in Computer Science and Smart Technology*, *6*(4), 424-442. https://doi.org/10.36548/jtcsst.2024.4.007.

Pawar, S.S., Manepatil, A., Kadam, A., & Jagtap, P. (2016). Keyword search in information retrieval and relational database system: two class view. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques* (pp. 4534-4540). IEEE. Chennai, India. https://doi.org/10.1109/iceeot.2016.7755576.

Taipalus, T. (2024). Vector database management systems: fundamental concepts, use-cases, and current challenges. *Cognitive Systems Research*, *85*, 101216. https://doi.org/10.1016/j.cogsys.2024.101216.

Tan, H., Zhan, S., Lin, H., Zheng, H.T., & Chan, W.K. (2025). QAEA-DR: a unified text augmentation framework for dense retrieval. *IEEE Transactions on Knowledge and Data Engineering*, *37*(6), 3669-3683. https://doi.org/10.1109/tkde.2025.3543203.

Wang, L., Luo, Z., Li, J., & Chen, C. (2020). Target-oriented transformation networks for document retrieval. In *Proceedings of the 2020 9th International Conference on Computing and Pattern Recognition* (pp. 447-454). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/3436369.3437413.

Wu, Y., Hu, N., Bi, S., Qi, G., Ren, J., Xie, A., & Song, W. (2023). Retrieve-rewrite-answer: a kg-to-text enhanced llms framework for knowledge graph question answering. *Computation and Language*. arXiv preprint arXiv:2309.11206.

Yang, L., Xu, M., & Ke, W. (2024). Enhancing question answering precision with optimized vector retrieval and instructions. *Information Retrieval*. https://doi.org/10.48550/arXiv.2411.01039.

Zamani, H., & Croft, W.B. (2016). Estimating embedding vectors for queries. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval* (pp. 123-132). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/2970398.2970403.

Zhang, W., Zeng, K., Yang, X., Shi, T., & Wang, P. (2023). Text-to-esq: a two-stage controllable approach for efficient retrieval of vaccine adverse events from nosql database. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* (pp. 1-10). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/3584371.3613008.

Zhao, W.X., Liu, J., Ren, R., & Wen, J.R. (2024). Dense text retrieval based on pretrained language models: a survey. *ACM Transactions on Information Systems*, *42*(4), 1-60. https://doi.org/10.1145/3637870.

Zhuang, S., Koopman, B., Chu, X., & Zuccon, G. (2024). Understanding and mitigating the threat of vec2text to dense retrieval systems. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region* (pp. 259-268). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/3673791.3698414.