International Transactions on Electrical Engineering and Computer Science
Priyadarshini et.al., Vol: 4, No: 4, pp: 194-212, December 2025

Automatic Query Expansion based Document Retrieval System using Hyper Tuned Graph Enhanced Bi-directional Encoder Representation from Transformers with Manifold Ranking Algorithm

D. Y. B. Priyadarshini[1], Aquter Babu[1]

Abstract: Document retrieval system automatic query expansion (AQE) adds relevant phrases to user searches to improve search results. AQE enhances recollection but introduces irrelevant phrases, increases computing complexity, and slows retrieval. Machine learning and deep learning models may be computationally expensive and sluggish in AQE-based document retrieval. These models also need large datasets and careful tweaking, which may be resource-intensive. The suggested AQE model uses HT-GEBERT Enhanced to generate contextual query expansions and Manifold Ranking to order words by relevance. This combination improves document retrieval accuracy and reduces query drift. Start by augmenting the corpus user query using Pseudo Adversarial Embedding (PAE) to increase data variety for robust model training. To maintain model analysis consistency, the augmented text and response are pre-processed using tokenization, lemmatization, acronym expansion, stop word removal, hyperlink removal, and spell correction. Next, Eccentricity-Based Keyword Extraction (EKE) extracts key phrases. After keyword extraction, the Hyper-Tuned Graph Enhanced Bidirectional Encoder Representation from Transformers (HT-GEBERT) model vectorises words and optimizes its hyper parameters using the Artificial Gorilla Troops Optimization Algorithm. Finally, a ranking-based query expansion approach re-ranks the phrase using a manifold ranking algorithm and splits the text into pieces for cosine similarity relevance assessment to obtain the document. The suggested method achieves 94% accuracy, 94.5% PPV, and 5.5% FDR in dataset 1. The suggested method uses query expansion, embedding, and optimization to retrieve documents.

Keywords: Eccentricity Based Keyword Extraction, Artificial Gorilla Troops Optimization Algorithm, HT-GEBERT, Automatic query expansion, manifold ranking, Pseudo Adversarial Embedding.

History
Received: 15-07-2025; Revised: 05-09-2025;
Accepted: 30-09-2025

✉ D. Y. B. Priyadarshini
beulahmca@gmail.com

[1]Department of Computer Science and Technology, Dravidian University, Kuppam – 517425, India

1. Introduction

Query expansion is widely used in various applications, including question-answering systems, multimedia information retrieval, and information filtering, and is applied across multiple domains. It is an efficient technique to improve the performance and reliability of document retrieval systems [1]. There are three types of query expansion: manual, interactive, and automatic. In manual query expansion, the user adds terms to the query, which yields good results but is time-consuming. In interactive query expansion, the

https://doi.org/10.62760/iteecs.4.4.2025.164

system suggests terms, and the user selects the most relevant ones, though this approach is also inefficient [2]. To solve this, Automatic Query Expansion (AQE) was developed to automatically identify the most suitable terms [3]. An AQE based document retrieval system enhances search results by expanding user queries with related terms. It identifies synonyms, related concepts, or alternative phrases to improve precision. AQE leverages techniques like natural language processing (NLP), statistical models to suggest expansions. This approach reduces the impact of ambiguous or incomplete queries, ensuring more relevant documents are retrieved [4]. By refining search terms automatically, AQE improves search engine accuracy and user satisfaction. However, AQE has the risk of introducing irrelevant terms, which can reduce precision and retrieve less relevant documents. Automatic query expansion can introduce irrelevant terms, leading to noise and decreased precision. It may over-rely on synonyms, missing context-specific meanings and reducing retrieval effectiveness [5]. Additionally, it requires substantial computational resources, potentially slowing down the retrieval process. Consequently, researchers have introduced machine learning algorithms in recent years to enhance query expansion in document retrieval systems. Machine learning (ML), is a subfield of artificial intelligence (AI), involves creating computer algorithms that automatically improve their performance through experience and data utilization. Various algorithms, including K-Nearest Neighbors (KNN), Random Forest (RF), Decision Trees (DT), Support Vector Machines (SVM), and Linear Regression (LR), have been employed for automatic query expansion in document retrieval systems. Machine learning techniques for automatic query expansion can suffer from overfitting, leading to poor generalization on unseen data [6]. They may also require large amounts of labeled training data, which can be challenging to obtain. Additionally, the complexity of these models can result in longer processing times, impacting system efficiency [7]. Due to various drawbacks in machine learning techniques for automatic query expansion in document retrieval systems, researchers are increasingly turning to deep learning approaches.

Deep learning techniques like Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), and Deep Neural Networks (DNN) transmit several drawbacks for automatic query expansion in document retrieval systems. These models generally necessitate substantial amounts of labeled training data, which can be challenging and time-consuming to gather [8]. Additionally, they are computationally intensive, demanding significant resources for training and inference, which can lead to slower response times [9]. Overfitting is another concern, where the model learns noise in the training data rather than generalizable patterns, reducing its effectiveness on unseen queries. Furthermore, the complexity of these architectures can make them challenging to interpret, hindering the understanding of how specific features influence query expansion. Lastly, they may struggle with the diversity and ambiguity of natural language, potentially leading to less effective query enhancements [10]. To address these challenges, a deep learning approach is introduced for automatic query expansion in document retrieval systems, employing a hyper-tuned Graph Enhanced BERT model combined with a manifold ranking algorithm. A major contribution of the work is given below in document retrieval systems, automatic query expansion leverages BERT in combination with ranking algorithms.

- User query from the corpus document is augmented using the Pseudo Adversarial Embedding (PAE) approach, which modifies the query to enhance the model's robustness.
- The augmented Query text and answer are pre-processed to enhance the data quality for improving model performance.
- Eccentricity-based Keyword Extraction (EKE) is utilized to extract keywords from the pre-processed text to identify important keywords in the document.
- After extracting keywords, the HT-GEBERT model converts words into vectors, with hyper parameters optimized using the Artificial Gorilla Troops Optimization Algorithm.
- After transforming the words into vector representations, a ranking-based query expansion method is implemented to retrieve the document effectively.

The next sections of this research work are organized as follows: Section 2 describes several papers related to query expansion in document retrieval systems using various machine learning and deep learning algorithms. Section 3 presents the proposed methodology for a query expansion-based document retrieval system employing a manifold ranking algorithm. The results obtained from the suggested method are discussed in Section 4, while the conclusion of research work is outlined in Section 5.

2. Literature Survey

There are numerous methods available for automatic query expansion in document retrieval systems. Below, examine and review some of these query expansion approaches.

Lorenzo Massai [11] has developed a Convolutional Neural Network for Query Expansion. Query Expansion involves using neural networks to improve search results by reformulating user queries based on context, intent, and related terms. The deep learning model for query expansion achieved an accuracy of 85%, with precision at 80%, recall at 75%, and an F1 score of 0.77. The error rate was recorded at 0.15, indicating a robust performance overall.

Keet Sugathadasa [12] has introduced a Mapper Neural Network to manipulate Legal Document Retrieval using Document Vector Embeddings. This approach leverages deep learning to understand the context and meaning within legal texts, improving the precision and relevance of search results. The result shows the achieved accuracy of 90.34%, and Precision at 90%. The model's F1 score is 88.32, indicating a balanced performance in retrieving relevant documents.

Xiaowang et al [13] have suggested a Recurrent Neural Network to operate an Automatic Query Expansion. Automatic query expansion enhances search results by dynamically generating relevant terms based on user queries. Using RNNs for automatic query expansion yielded an accuracy of 82%, and a precision at 78%. The model demonstrated its effectiveness in improving query results.

Wei Ou and Van-Nam Huynh [14] have designed Deep Neural Network for Effective Query Expansion. The effective query expansion leverage neural networks to enhance query relevance by understanding semantic relationships between terms. The effective query expansion achieved an accuracy of 81%, with precision at 75% and F1_Score at 73% for reflecting its balanced performance in improving query relevance.

Luis M. de Campos et al [15] have developed a Long Short Term Memory to utilize a robust query expansion. It involves generating adversarial to improve the model's resilience against noisy. In this approach, the model achieved an accuracy of 87%. With a precision of 82% and recall of 77%.The error rate was 0.11, and the F1 score of 0.79 shows the model's capacity for maintaining a good trade-off between precision and recall.

Harun Bolat and Baha Şen [16] have introduced a Variational Auto encoder to operate Multi-Task Learning for Query Expansion. Query expansion trains a model on multiple related tasks simultaneously, allowing it to leverage shared information across tasks to enhance query relevance. In this approach, the model achieved an accuracy of 86%, with a precision of 81% and a recall of 78%, the error rate of 0.13, F1 score of 0.80 highlights its balanced performance.

Amol P. Bhopale and Ashish Tiwari [17] have designed a Bi Long Short Term Memory to operate a Query Expansion in Retrieval Systems. The retrieval systems were a technique used to improve search results by automatically adding related terms or phrases to a user's original query. In this context, the model achieved an accuracy of 85%, the precision was recorded at 80%, with a recall of 75%, the error rate of 0.15, while the F1 score of 0.77 showcases the model's ability to enhance search effectiveness.

2.1 Research Gap

Based on the reviews mentioned above, some existing automatic query expansion-based document retrieval systems address several drawbacks, such as Deep learning for query expansion may suffer from high computational costs, limited interpretability of results, and the risk of specific datasets [11]. Enhanced document retrieval using deep learning techniques may suffer from overfitting on training data, leading to reduced performance on unseen documents [12].

But, RNNs can be computationally intensive and may require extensive data to achieve optimal performance, potentially limiting their scalability [13]. However, these models can specific query patterns, reducing their generalization to diverse search contexts [14]. Still, it can sometimes lead to overfitting to specific adversarial patterns, limiting the model's adaptability to unseen queries [15]. Yet, balancing multiple objectives increases model complexity, making it harder to tune and requiring more data for effective training [16]. However, the effectiveness of query expansion heavily depends on the quality of the expansion terms, which can vary significantly based on the underlying algorithm and dataset used [17]. So the Hyper Tuned Graph Enhanced Bi-directional Encoder Representation from Transformers (HGE-BERT) reduces overfitting through hyper parameter tuning and graph-based relationships. It also minimizes computational complexity via efficient attention mechanisms and bidirectional encoding for enhancing document retrieval.

3. Proposed methodology

Searching for relevant material that satisfies the information needs of a user, within a large document collection, is a critical activity for web search engines. Query Expansion techniques are widely used by search engines for the disambiguation of user's information need and for improving the information retrieval (IR) performance. The retrieved list of documents is sometimes large and contains many irrelevant documents, especially when searching the Web. The main issue encountered in the retrieval of documents that are not related to user needs is the vocabulary mismatch problem. To automatically obtain the information, a variety of techniques are used, including word weighting and reweighting, term selection, pseudo-relevant feedback, and others. Nevertheless, while expanding the query expansion techniques may introduce irrelevant information. Therefore, the current study focuses on developing efficient, automated methods for query expansion to extract relevant information. The architecture of the proposed automatic query expansion system is given in Fig. 1.

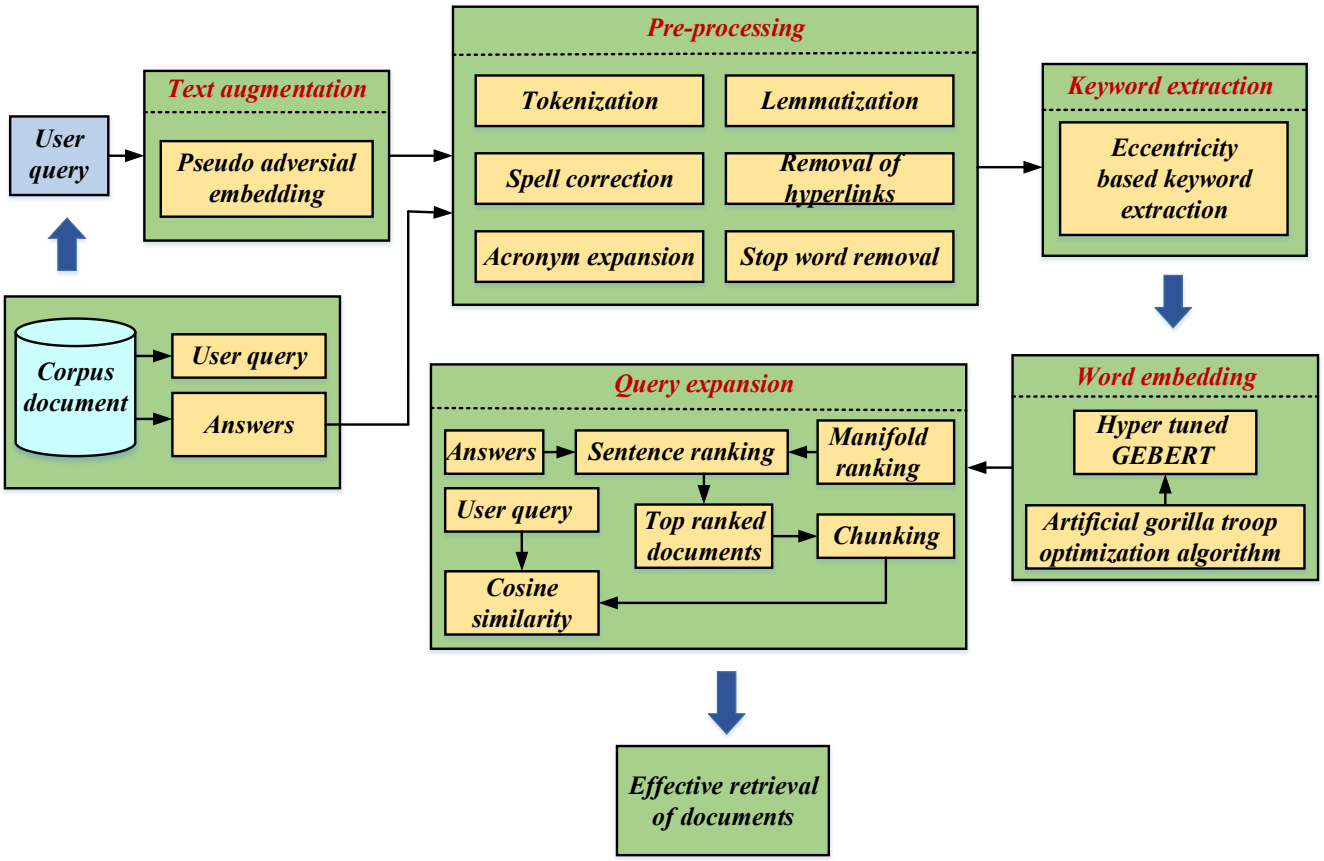

Fig. 1: Flow diagram for the proposed methodology

At First, Pseudo Adversarial Embedding (PAE) is used to improve the user query from the corpus document. Then the Pre-processing methods including tokenization, lemmatization, acronym expansion, stop word removal, hyperlink removal, and spell correction are applied to the augmented text and answer to ensure a consistent format for efficient model analysis. After preprocessing, Eccentricity-Based Keyword Extraction (EKE) is utilized to identify the key terms in the document. Once the keywords are extracted, the Hyper-Tuned Graph Enhanced Bi-directional Encoder Representation from Transformers (HT-GEBERT) model converts the words into vector format, and the Artificial Gorilla Troops Optimization Algorithm (AGTOA) is employed to tune hyper parameters such as learning rate and weight decay. The sentence is then reranked using a manifold ranking algorithm on the basis of higher rank using a ranking-based query expansion approach. The text is then divided into chunks for the purpose of determining relevance using cosine similarity in order to obtain the document.

3.1 Text augmentation

Text augmentation is a technique used to artificially increase the size of a text dataset by applying various transformations, such as synonym replacement, random insertion, deletion, and swapping of words. These techniques help enhance the diversity of the data, improving the performance and generalization of machine learning models. It's particularly useful in situations with limited labeled data. At first, the user query extracted from the corpus document was enhanced through the PAE approach.

3.1.1 Pseudo Adversial Embedding

Pseudo-adversarial embedding is a technique in machine learning where a model is trained to generate slightly perturbed data embeddings that mimic adversarial, specifically organizing aggressive attacks [18]. This helps the model become more robust to small changes or noise in the data, improving generalization and resilience against potential adversarial inputs. It strikes a balance between regular training and full adversarial training. Embedding generation is the process of converting data (like text, images, or signals) into fixed-size numerical vectors that capture relevant features or patterns. These vectors are used in machine learning models to represent the data in a lower-dimensional, structured form as shown below in equation (1)

$$E_x = f_0(x) \tag{1}$$

Where, E_x is the embedding of input, x denotes the generated model, f_0 represents parameterized by 0.

Adversarial embedding refers to a technique where small, often imperceptible perturbations are added to input data (like images or text) to mislead machine learning models into making incorrect predictions. These embeddings are crafted specifically to exploit model weaknesses, highlighting vulnerabilities in neural networks as shown below in equation (2)

$$E_x^{adv} = E_x + \epsilon.\nabla_{E_x} L(f_0(x), y) \tag{2}$$

Where, E_x^{adv} denotes the embedding of the adversarial, E_x is the embedding of input, x denotes the generated model, f_0 represents the parameterized by 0, ϵ is a small perturbation, $\nabla_{E_x} L$ denotes the gradient of the loss function. A perturbed loss function introduces random noise or perturbations to the standard loss function during training, which helps the model escape local minima and improves generalization as shown below in equation (3)

$$L_{adv} = L(f_0(x + \delta), y) \tag{3}$$

Where, L_{adv} denotes the perturbed loss function, L, y is the original loss function, $f_0(x + \delta)$ represents the model's prediction. Exploration in embedding space refers to the process of probing different regions of the learned representation space to discover diverse patterns or features. It helps models generalize by ensuring that embedding vectors capture a wide range of semantic relationships in equation (4)

$$E_{explore} = E_x + \alpha.r \tag{4}$$

Where $E_{explore}$ denotes the sum of the base energy, E_x represents the original energy, α is the scaling factor, r denotes the random vector.

Augmented adversarial training loss combines the standard loss with adversarial examples generated through augmentation techniques, enhancing the model's robustness against adversarial attacks as shown below in equation (5)

$$L_{avg} = L(f_0(x + \delta_{avg}), y) \quad (5)$$

Where, L_{avg} denotes the augmented adversarial training loss, L represents the loss function, f_0 is the model being evaluated, x denotes the original input data, δ_{avg} represents the average perturbation, y is the true label.

The user query from the corpus document was augmented using the Pseudo Adversarial Embedding (PAE) approach, which generated enhanced representations of the query by introducing controlled perturbations. This augmentation improved the model's robustness in understanding diverse query formulations and led to more accurate and relevant responses during retrieval and classification tasks.

3.2 Pre Processing

Preprocessing involves transforming raw data into a format suitable for analysis or modeling by cleaning, normalizing, or encoding it. This step enhances data quality and consistency, facilitating more effective learning and improving model performance. Then, the augmented text is pre-processed using various text preprocessing techniques. It is essential because the text must be translated into a comprehensible and consistent format before it can be fed into a model for further analysis and learning.

3.2.1 Tokenization

Tokenization is the process of dividing text into smaller units called tokens, which can be words, phrases, or symbols. This step is crucial in natural language processing (NLP) as it transforms unstructured text into a structured format that algorithms can process [19]. By breaking down sentences into tokens, models can better understand and analyze the relationships between words. Tokenization facilitates various NLP tasks, such as text classification, sentiment analysis, and machine translation. Additionally, it helps in handling variations in language, such as different word forms or punctuation, making it essential for effective language understanding.

3.2.2 Lemmatization

Lemmatization is the process of reducing a word to its base or dictionary form, known as its lemma, by removing inflections and variations [20]. Unlike stemming, which often produces crude and non-words, lemmatization considers the context and meaning of the word, ensuring that the output is a valid word. For example, "running" and "ran" are both reduced to the lemma "run." Lemmatization is widely used in natural language processing (NLP) tasks such as text classification, information retrieval, and sentiment analysis, as it helps normalize words and reduce dimensionality. By focusing on base forms, lemmatization enhances the quality of text data, making it easier for models to understand and process language effectively.

3.2.3 Expanding Acronyms

Expanding acronyms involves providing the full form or meaning of an abbreviation or acronym [21]. This process helps clarify the terms for those who may not be familiar with them. For instance, "NASA" expands to "National Aeronautics and Space Administration." Similarly, "AI" stands for "Artificial Intelligence," and "HTML" means "Hypertext Markup Language." Expanding acronyms is essential in technical writing, education, and communication to enhance understanding. It ensures that all readers, regardless of their background knowledge, can grasp the content effectively.

3.2.4 Removal of Hyperlinks

Removal of hyperlinks involves eliminating embedded links from text documents or web content to enhance readability and clarity [22]. This process helps reduce distractions for readers, allowing them to focus on the main content without being sidetracked by external references. In data preprocessing, removing hyperlinks can also prevent noise in natural language processing tasks, ensuring that algorithms are trained on relevant textual data. Furthermore, it aids in protecting sensitive information that may be contained within URLs. Overall, hyperlink removal contributes to a cleaner presentation of information and improved user experience.

3.2.5 Spell Correction

Spell correction is the process of identifying and correcting misspellings in text, often using algorithms that compare the input words against a dictionary or a corpus of correctly spelled words [23]. It typically involves techniques such as Levenshtein distance, phonetic matching, and machine learning models to predict the most likely correction based on context. Spell correction is widely used in text processing applications, such as word processors, search engines, and messaging platforms, to enhance user experience by reducing errors. It also plays a crucial role in natural language processing tasks, improving the quality of text input for sentiment analysis, translation, and information retrieval. Ultimately, effective spell correction helps maintain clarity and accuracy in communication.

3.2.6 Stop Word Removal

Stop word removal is the process of eliminating common words from text that carry little semantic value, such as "and," "the," "is," and "in." This technique is often employed in natural language processing (NLP) and information retrieval to reduce noise in the data and focus on the meaningful words that contribute to the context [24]. By removing stop words, the computational efficiency of text analysis can be improved, as fewer tokens need to be processed. It also enhances the accuracy of models by reducing dimensionality and allowing them to better capture the underlying patterns in the data. Overall, stop word removal is a crucial preprocessing step in tasks such as text classification, sentiment analysis, and search engine optimization. Following these preprocessing steps, the cleaned and structured text was then transformed into numerical representations through techniques like vectorization or embedding. This transformation allowed the model to effectively interpret and analyze the textual data by capturing the semantic relationships and contextual meanings within the text. Ultimately, these efforts aimed to enhance the model's performance by providing it with high-quality input, facilitating more accurate predictions and insights during subsequent analysis and learning phases.

3.3 Keyword Extraction

Keyword extraction is the process of identifying and extracting significant words or phrases from a text that represent its main topics or themes. This technique is widely used in natural language processing (NLP) to improve information retrieval, content summarization, and search engine optimization. Highlighting the most relevant keywords helps in organizing and understanding large volumes of unstructured text data efficiently. Then, extracting keywords from the pre-processed text using Eccentricity-based keyword extraction (EKE). This method focuses on identifying key keywords from the document by utilizing an eccentricity-based centrality measure for extraction.

3.3.1 Eccentricity based Keyword Extraction

Eccentricity-based keyword extraction is a method that identifies important keywords by analyzing the structural properties of a text's underlying graph representation, where words are nodes connected by edges based on their co-occurrences [25]. This approach measures the eccentricity of each word, which reflects its distance from other words in the graph, allowing for the selection of keywords that are central or significant within the context. By focusing on words with high eccentricity scores, the effectively highlights terms that contribute meaningfully to the overall content and context of the text. Distance calculation in the context of graph theory refers to determining the shortest path length between two nodes in a graph, representing the minimal number of edges that must be traversed to connect them. This metric helps quantify the relationship between nodes, providing insights into their proximity or relevance in the underlying structure of the data as shown below in equation (6)

$$d(w_i, w_j) = shortest\_path(w_i, w_j) \quad (6)$$

Where, $d(w_i, w_j)$ is the difference between two nodes, w_i, w_j represents the shortest path in the graph.

Eccentricity in graph theory is a measure of the maximum distance from a given node to any other node in the graph, indicating how far node is from its most distant counterpart. It helps identify the "centrality" or importance of a node within the

network, with lower eccentricity values suggesting greater centrality as shown below in equation (7)

$$e(w_i) - \max d(w_i, w_j) \quad (7)$$

Where, $e(w_i)$ is the eccentricity of the word, w_i, w_j represents the maximum distance in the graph.

Centrality measurement quantifies the importance or influence of a node within a graph or network by assessing its position relative to other nodes. Various centrality metrics, such as degree, closeness, between ness, and eigenvector centrality, provide different perspectives on a node's connectivity and role in the overall structure as shown below in equation (8)

$$C(w_i) = \frac{1}{e(w_i)+1} \quad (8)$$

Where, $C(w_i)$ denotes the centrality measure, $e(w_i)$ is the eccentricity of the word.

Keyword scoring is the process of assigning numerical values to words or phrases based on their relevance and importance within a specific context or document. This scoring helps prioritize keywords for extraction or analysis, enabling models to focus on the most significant terms that capture the main themes or topics as shown below in equation (9)

$$S(w_i) = weight(w_i) \times C(w_i) \quad (9)$$

Where, $S(w_i)$ represents the each word score, $C(w_i)$ denotes the centrality measure, w_i is the weighted centrality.

The final keyword list is a ranked collection of the most significant terms extracted from a text, typically based on their relevance or importance scores. These keywords represent the core concepts or topics of the document, aiding in summarization and information retrieval as shown below in equation (10)

$$K_{final} = sort(K, S(w_i)) \quad (10)$$

Where, K_{final} denotes the final keyword, K represents the candidate keywords extracted from the text, $S(w_i)$ is the score of each keyword.

In conclusion, the Eccentricity-based Keyword Extraction (EKE) method provides an efficient approach to identifying significant keywords from pre-processed text. By leveraging the eccentricity-based centrality measure, this technique highlights the most relevant terms, contributing to better summarization and insight extraction from the document. This approach not only enhances keyword identification but also ensures that key information is effectively represented, making it a valuable tool for various text analysis tasks.

3.4 Word Embedding

After extracting the keywords, a Hyper Tuned Graph Enhanced Bi-directional Encoder Representation from transformers (HT-GEBERT) model is utilized to convert the word into vector form. The hyperparameters present in the GEBERT, such as learning rate and weight decay, were optimally selected using Artificial Gorilla Troops Optimization Algorithm (AGTOA).

3.4.1 Hyper-tuned Graph Enhanced Bidirectional Encoder Representation from Transformers

Hyper-tuned Graph Enhanced Bidirectional Encoder Representation from Transformers (Hyper-tuned Graph-BERT) is an advanced natural language processing model that integrates graph-based structures with BERT architecture [26]. It enhances BERT by incorporating graph representations, allowing it to capture complex relationships between words and phrases. The model is fine-tuned (hyper-tuned) to optimize performance on specific tasks. This combination enables superior performance in tasks requiring contextual understanding and structural dependencies, such as text classification and entity recognition. BERT's input representation consists of three types of embeddings: token embeddings for each word or sub-word unit, segment embeddings to differentiate between sentence pairs (e.g., in a question-answering task), and positional embeddings to capture word order. These embeddings are summed and fed into BERT's transformer model. This allows BERT to handle tasks like classification, translation, and more as shown below in equation (11)

$$E_{input} = E_{token} + E_{segment} + E_{position} \quad (11)$$

Where, E_{input} is the input representation, E_{token} denotes the word embeddings, $E_{segment}$ represents the

sentence level embeddings, $E_{position}$ is the positional information to retain word order.

Each transformer layer operates by applying a multi-head self-attention mechanism, followed by a feed-forward network. The output of each layer is passed through normalization and residual connections, ensuring stable learning. This process helps the model capture complex dependencies in the data. The output of each layer as shown below in equation (12)

$$H_{l+1} = FFN\left(MultiHeadAttention(H_l)\right) + H_l \quad (12)$$

Where, H_l denotes the hidden state of the previous layer, H_{l+1} represents the transformer layer, FFN is the feed forward network.

Multi-head self-attention computes attention weights by comparing the query with corresponding keys to determine the relevance of each token. The mechanism then generates weighted combinations of values based on these attention scores. This allows the model to focus on different parts of the input sequence simultaneously as shown below in equation (13)

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (13)$$

Where, Q denotes the query, K is the key, V is the value, $\sqrt{d_k}$ denotes the dimension of the key matrices, QK^T is the attention weight mechanism.

Graph-enhanced embeddings incorporate graph structures by utilizing an adjacency matrix A, which captures the relationships between entities. This matrix is used to adjust the embeddings based on the graph's connectivity. As a result, the model represents structured data and entity interactions as shown below in equation (14)

$$E_{graph} = A.E_{input} \quad (14)$$

Where, E_{input} is the input representation, E_{graph} represents the graph embedding.

The combined embedding representation merges graph-enhanced embeddings with standard BERT embeddings. This integration captures both graph-based relationships and contextual word information. It allows the model to leverage both structural and semantic features for improved performance on various tasks as shown below in equation (15)

$$E_{combined} = \alpha E_{BERT} + \beta E_{graph} \quad (15)$$

Where, $E_{combined}$ represents the combined embedding, αE_{BERT} and βE_{graph} are the weights that balance the contribution of BERT and graph based embedding.

The fine-tuning loss function is task-specific, such as cross-entropy loss for classification tasks. During fine-tuning, the model adjusts its weights based on this loss to optimize performance for the target task. This process allows the pre-trained model to adapt to new tasks effectively as shown below in equation (16)

$$L = -\sum_{i=1}^{N} y_i \log(\tilde{y}i) \quad (16)$$

Where, L denotes the loss function, y_i represents the true label, $\tilde{y}i$ is the predicted probability, N denotes the number of cross entropy. This hybrid model has proven to be effective in understanding complex linguistic structures, making it highly suitable for tasks like question answering, relation extraction, and sentiment analysis.

3.4.2 Artificial Gorilla Troops Optimization Algorithm

The Artificial Gorilla Troops Optimization Algorithm (AGTOA) is a nature-inspired optimization technique that mimics the foraging behavior of gorilla troops in search of food. It employs a hierarchical structure to model the social dynamics and collaborative strategies of gorillas, enhancing the exploration and exploitation of the search space [27]. This algorithm is particularly effective for solving complex optimization problems across various domains. In the exploration phase, one of the primary strategies employed is the use of randomized search patterns. This involves gorilla troops unpredictably moving through the solution space, enabling them to cover a wide area and discover diverse potential solutions. In the exploitation phase of the Artificial Gorilla Troops Optimization Algorithm (AGTOA), the primary strategy is to perform a refined local search around the most promising solutions identified during the exploration phase.

Step 1: Initialization

The learning rate and weight decay are initialized based on the population of the Artificial Gorilla Troops and the mathematical formulation for this initialization function is mentioned using equations (17) and (18).

$$U = U_1, U_2, \ldots\ldots\ldots U_n \quad (17)$$

$$O = O_1, O_2, \ldots\ldots O_n \quad (18)$$

The learning rate is represented by the symbol U, with an upper limit of 1 and a lower limit of 0. The weight decay is represented by the symbol O, with an upper limit of 0.1 and a lower limit of 0.01.

Step 2: Fitness function

The fitness function for this optimization strategy is based on the minimization of Word Error Rate (WER) mentioned using equations (19) and (20). WER is calculated by comparing the output of a model (e.g., transcribed text) to a reference text (the correct text). It measures the number of words predicted wrongly, taking into account insertions, deletions, and substitutions.

$$Fitness\ function\ =\ minimize\ (WER) \quad (19)$$

$$WER = \frac{S+I+D}{N} \quad (20)$$

Where, S denotes the substitutions, I is the Insertions, D represents the Deletions and N denotes the number of words in the reference.

Step 3: Updation

The fitness value is updated based on the behaviour of Artificial Gorilla Troops, which includes both exploration and exploitation phases. In the Exploration phase, if rand ≥ 0.5, the mechanism for moving towards other gorillas is selected; however, if rand < 0.5, the mechanism for migrating to a known location is chosen. Subsequently, in the Exploitation phase, if C ≥ W, the local search mechanism is selected.

- ***Exploration Phase (****Randomized Search Pattern)*

The Exploration Phase in the Artificial Gorilla Troops Optimization (AGTO) algorithm involves agents (gorillas) randomly searching the solution space. Gorillas move in a semi-random pattern to explore new areas and avoid local optima. This phase emphasizes global search for potential solutions, mimicking the unpredictable movement of real gorillas. It ensures diversity in the search space, increasing the likelihood of finding better solutions. This phase balances exploration with later exploitation phases as shown below in equation (21)

$$X_i^{(t+1)} = X_i^{(t)} + \alpha.rand.(X_{best}^{(t)} - X_i^{(t)}) \quad (21)$$

Where, $X_i^{(t+1)}$ denotes the randomized search pattern, $X_i^{(t)}$ is the position of the iteration, $X_{best}^{(t)}$ represents the best iteration, α denotes the control parameter, $r\ and$ represents the random value between 0 to 1.

- ***Exploitation Phase (****Refined Local Search)*

The Exploitation Phase of the Artificial Gorilla Troops Optimization Algorithm focuses on refining elite solutions by conducting a local search around them. It identifies the best-performing gorillas and generates candidate solutions through small perturbations. Each candidate is evaluated for fitness, replacing less fit solutions as needed. This iterative process enhances solution quality and helps avoid local optima, balancing exploration and exploitation as shown below in equation (22).

$$X_i' = X_i + \alpha.(X_j - X_i) + \beta.(X_k - X_i) \quad (22)$$

Where, X_i' denotes the new solution generation from the current solution, X_i represents the current best solution, $\alpha\ and\ \beta$ is the control parameters, $X_j\ and\ X_k$ denotes the solutions from the neighborhood.

Step 4: Termination

Once the optimal solution is identified, the process is terminated. A flowchart for the Artificial Gorilla Troops Optimization Algorithm (AGTOA) is presented in Fig. 2. The hyper parameters present in the GEBERT, such as learning rate and weight decay, were optimally selected using the Artificial Gorilla Troops Optimization Algorithm (AGTOA). This approach significantly enhanced model performance and ensured efficient convergence during training.

3.5 Query expansion Based on Manifold Ranking Algorithm

Query expansion is a technique used in information retrieval to improve search results by reformulating or augmenting the original query. This process often involves adding synonyms, related terms, or variations of keywords to capture more relevant documents.

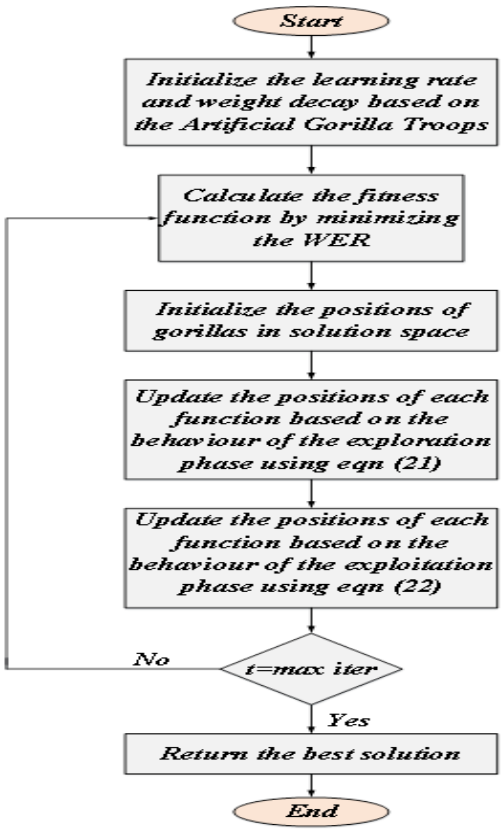

Fig. 2: Flow Chart for AGTO Algorithm

By expanding the query, it aims to address issues of vocabulary mismatch and to enhance retrieval effectiveness. Ultimately, query expansion helps users find more accurate and comprehensive information related to their search intent. Ranking-based query expansion is a technique that improves search results by expanding the original query with additional relevant terms ranked according to their importance [28]. This method leverages user interactions and document relevance to refine the search, increasing the likelihood of retrieving more pertinent information. After converting the words into vector form, a ranking-based query expansion approach is developed to enhance the effectiveness of document retrieval systems. This process consists of several key steps. The first step involves reranking the documents using a manifold ranking algorithm, which assesses the relationships between documents to identify the most relevant ones. In the second step, the highest-ranked documents are divided into text chunks of a predetermined length, and each individual chunk is evaluated for its relevance. Finally, the selected chunks are combined with the original query to score the document, determining its relevance based on cosine similarity. This systematic approach improves retrieval performance by ensuring that the most pertinent information is highlighted and presented to the user. Algorithm - 1 explains the pseudo code for the Effective retrieval of document as shown below.

Algorithm-1: Pseudo code for Effective Retrieval of Document

Input: User Query (Z) and Answer(M)
 # Text augmentation
{
AT= PAE (Z)// for data augmentation using equation (5)
}
 # Preprocessing (AT, M)
{
RM= Tokenization (AT, M) // splits text into individual words
HI= Lemmatization (RM)// reduces words to their base forms
DG= Spell_Correction (HI)// improves text accuracy by fixing graphical errors.
FT= hyperlinks_Removal (DG)// elimination of hyperlink
FG= Acronym_ expansion (FT)// enhances understanding.
SR= stop word removal (FG) // removing non-informative words
}
 # Keyword Extraction
{
AE= EBKE (SR) // identify important keywords from the document using equation (10)
}
 # Word Embedding
{
RT= HT-GEBERT (AE)// convert the word into vector form using equation (16)
TO=AGTOA (RT) //optimally selection of hyper parameter
{
Calculate fitness function
F=min(WER)

}
 # Query expansion
{
RH= Manifold Ranking(M)
UM= Chunk(RH)
ET= Cosine Similarity (UM) and (Z)
}
Output: Retrieved appropriate document
END

4. Result and discussion

The proposed model enhances query relevance using HT-GEBERT for expansion and ranks documents with the manifold ranking algorithm for retrieval. The user submits a query that interacts with the system. Then the user query undergoes pseudo adversarial embedding to enhance robustness by creating variations of the query, improving its adaptability for further stages. The proposed framework is implemented using Python 3.8, with the system running on an Intel Core i5 CPU, Nvidia GeForce GTX 1650 GPU, and 16GB of RAM. The simulation parameters for GEBERT and AGTOA are provided in Tables. 1 and Table. 2.

Table. 1: Simulation parameter for AGTOA

Parameter	Values
max_iterations	50
num_gorillas	10
search_space	[(0.00001, 0.01), (0.0, 0.3)]
Beta	3
W	0.8

Table. 2: Simulation parameter for GEBERT

Parameter	Values
Learning Rate	0.0002
Weight_decay	0.294
optimizer	Adam
Maximum sequence length	512
Epoch	40
dropout	0.1

Table. 1, presents the simulation parameters for AGTOA, including Max_iterations, num_gorillas, search_space, Beta and W. Table. 2 lists the simulation parameters for GEBERT, such as Learning Rate, Weight_decay, optimizer, Maximum sequence length, Epoch and dropout.

Table. 3: Preprocessing of Sample data

Original Text	Stopword	Removal Hyperlinks	Tokenization	Lemmatization
Steps to develop an entrepreneurial mindset?	steps develop entrepreneurial mindset	steps to develop an entrepreneurial mindset	['steps', 'to', 'develop', 'an', 'entrepreneurial', 'mindset']	step develop entrepreneurial midst
Identify the key species in marine biology contributing to ecosystem health?	identify key species marine biology contributing ecosystem health	identify the key species in marine biology contributing to ecosystem health	['identify', 'the', 'key', 'species', 'in', 'marine', 'biology', 'contributing', 'to', 'ecosystem', 'health']	identify key species marine biology contributing ecosystem health
Compile a list of traditional dishes from five different countries?	compile list traditional dishes five different countries	compile a list of traditional dishes from five different countries	['compile', 'a', 'list', 'of', 'traditional', 'dishes', 'from', 'five', 'different', 'countries']	compile list traditional dish five different country
Describe the evolution of classical music through different periods?	describe evolution classical music different periods	describe the evolution of classical music through different periods	['describe', 'the', 'evolution', 'of', 'classical', 'music', 'through', 'different', 'periods']	describe evolution classical music different period
What are the most common psychological disorders today?	common psychological disorders today	what are the most common psychological disorders today	['what', 'are', 'the', 'most', 'common', 'psychological', 'disorders', 'today']	common psychological disorder today

Table. 3, illustrates the original text from a sample of pre-processed data using the Hugging Face dataset. The original text is presented in the first column, while the stopword and removal hyperlink text are displayed in the second and third columns. The fourth and fifth columns demonstrate tokenization and lemmatization performed using proposed algorithms.

4.1 Dataset Description

The data is collected from the Hugging face website [29], [30] and [31]. The Book Corpus dataset originally included 11,038 books. However, the files obtained indicate that there are only 7,185 unique books, excluding romance-all.txt and adventure-all.txt, are noted. An analysis of the file names revealed the possibility of 2,930 duplicate books. Each book in Book Corpus contains the complete text of the ebook, which may include elements like the preamble and copyright information. However, researchers using Book Corpus have employed various encoding schemes that alter the definition of an "instance" (for example, GPT-N training utilizes byte-pair encoding for text). The data fields remain consistent across all splits. Overall, 80% of the data is allocated for training, while 20% is reserved for testing.

4.2 Comparison Analysis

The proposed HT-GEBERT model is compared against existing algorithms, including Graph Neural Networks (GNN), Bidirectional Long Short-Term Memory (BiLSTM), Gated Recurrent Units (GRU), and Bidirectional Encoder Representations from Transformers (BERT).

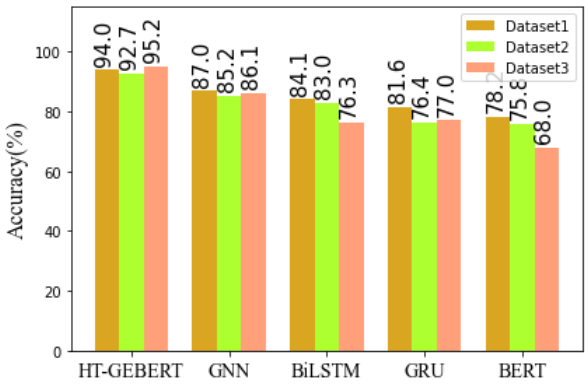

Fig. 3: Comparison of Accuracy with other existing algorithms

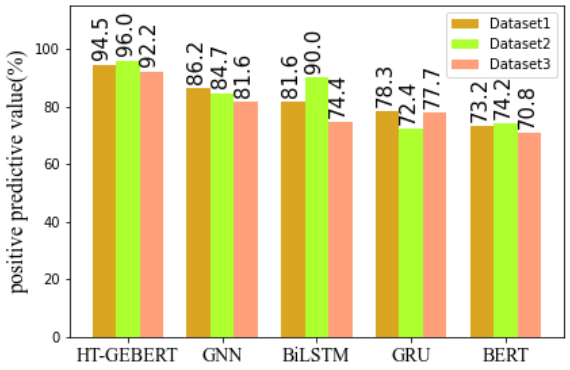

Fig. 4: Comparison of the positive predictive value of HT-GEBERT with other existing algorithms

Several performance metrics are used to analyze the effectiveness of these approaches, including: Accuracy, positive predictive value, Hit rate, Miss Rate, FDR, F1_Score, Error, Mean Absolute Percentage Error, Normalized Levenshtein Similarity, Bert Score, Bleu Score, Meteor Score, Character Error Rate and Word Error Rate. Fig. 3 and Fig. 4 provide a comparison of Accuracy and Positive Predictive Value (PPV). These metrics compare the proposed HT-GEBERT method with existing models are GNN, BiLSTM, GRU, and BERT. For Dataset -1, the Accuracy and PPV of the proposed approach are 94% and 94.5%, respectively, while the values for the existing models are 92.7% and 95.2% for Accuracy, and 96% and 92.2% for PPV. The HT-GEBERT algorithm outperforms the other models due to its graph structure, which enhances BERT's ability to capture long-range token dependencies, improving contextual representation, especially in complex multi-hop reasoning tasks.

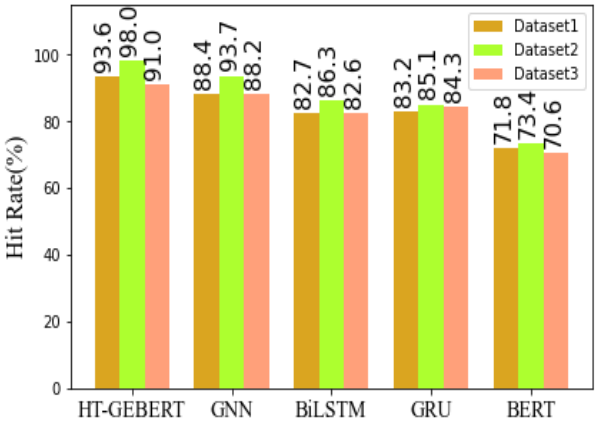

Fig. 5: Hit rate comparison between the proposed algorithm and other existing algorithms

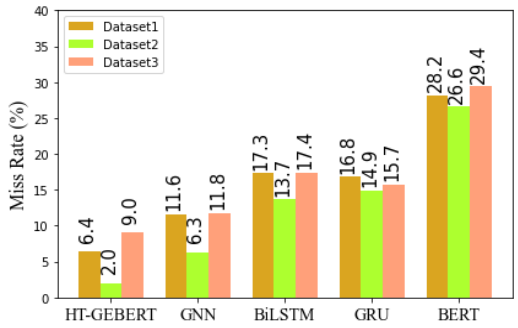

Fig. 6: Comparison of Miss Rate for HT-GEBERT with various existing algorithms.

The Hit Rate comparison is presented in Fig. 5. Using the proposed approach, the Hit Rate values for the three datasets are 93.6%, 98%, and 91%. Fig. 6 shows the Miss Rate comparison for various existing approaches. For the proposed approach, the Miss Rate values are 6.4%, 2%, and 9%. The HT-GEBERT algorithm outperforms other existing algorithms by effectively leveraging graph embeddings, which capture complex relationships between words, sentences, or entities, improving the model's understanding of semantic connections beyond simple token sequences.

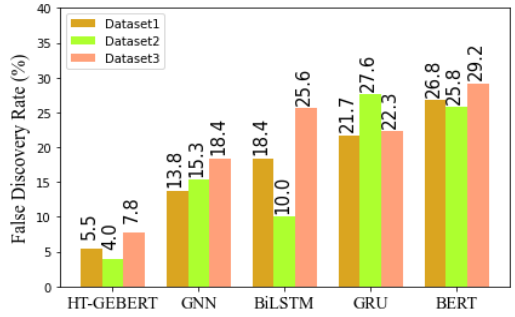

Fig. 7: Comparison of HT-GEBERT's FDR values with various existing techniques.

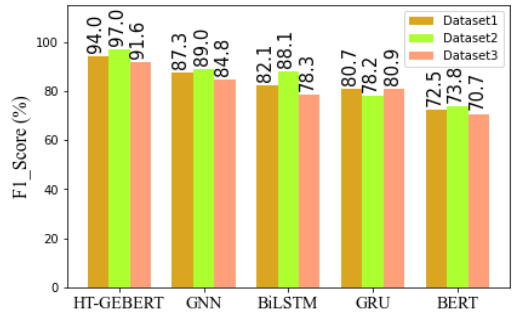

Fig. 8: HT-GEBERT of F1_Score value compared to other existing algorithms

Fig. 7 and Fig. 8 compare the FDR and F1_Score of the proposed HT-GEBERT method against existing models such as GNN, BiLSTM, GRU, and BERT. For Dataset 1, HT-GEBERT achieves an FDR of 5.5% and an F1_Score of 94%, whereas the existing models show FDR values of 4% and 7.8%, and F1_Scores of 97% and 91.6%. The HT-GEBERT algorithm, by leveraging graph-based structures, improves entity disambiguation and demonstrates better accuracy in NER and question-answering tasks compared to existing algorithms.

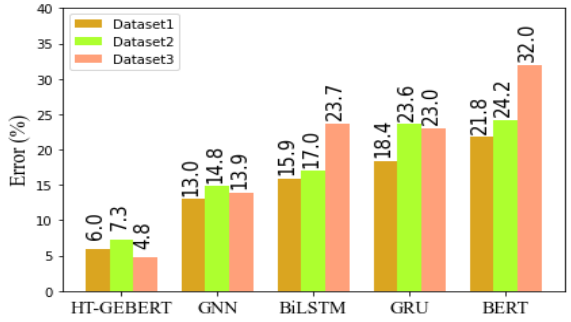

Fig. 9: Comparison of the Error value of HT-GEBERT with various existing techniques.

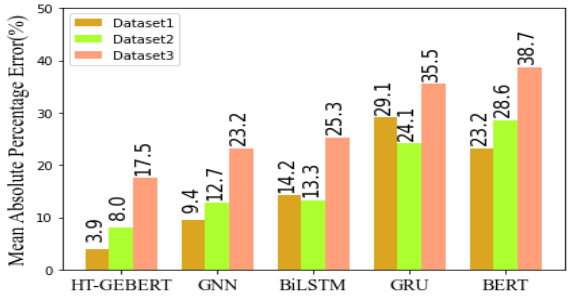

Fig. 10: Mean Absolute Percentage Error evaluation for HT-GEBERT with other existing techniques

Fig. 9 illustrates the comparison of error rates. The proposed approach yields error values of 6%, 7.3%, and 4% across the three datasets. Fig. 10 presents a comparison of Mean Absolute Percentage Error (MAPE), with the proposed method showing error rates of 3.9%, 8%, and 17.5% across the datasets. When compared to other existing algorithms, the HT-GEBERT algorithm demonstrates a notable improvement, particularly when handling data inherently structured as graphs. GEBERT effectively utilizes this structure to deliver enhanced performance compared to standard BERT.

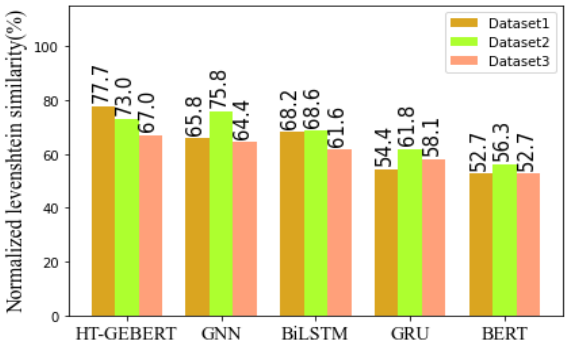

Fig. 11: Comparison of Normalized Levenshtein Similarity for HT-GEBERT with various existing algorithms.

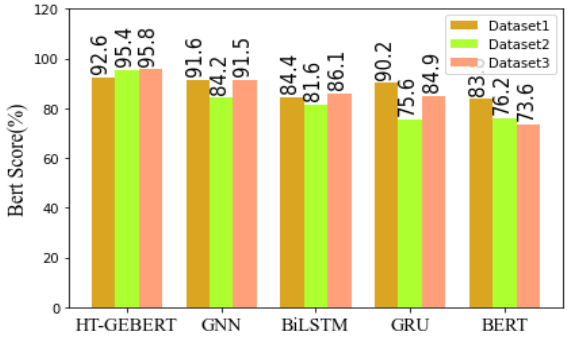

Fig. 12: Bert Score comparison between the proposed algorithm and other existing algorithms

Fig. 11 and 12 compare the NLS and BERT Score metrics, evaluating the proposed HT-GEBERT method against existing models, including GNN, BiLSTM, GRU, and BERT. For Dataset 1, HT-GEBERT achieves NLS and BERT Score values of 77.7% and 92.6%, respectively, while the existing models have NLS values of 73% and 67%, and BERT Score values of 95.4% and 95.8%.

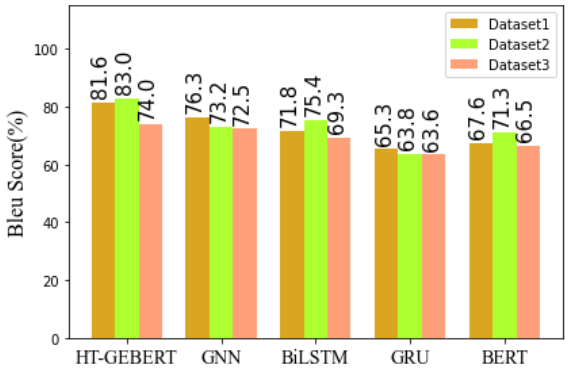

Fig. 13: Comparison of the Bleu Score of HT-GEBERT with various existing techniques.

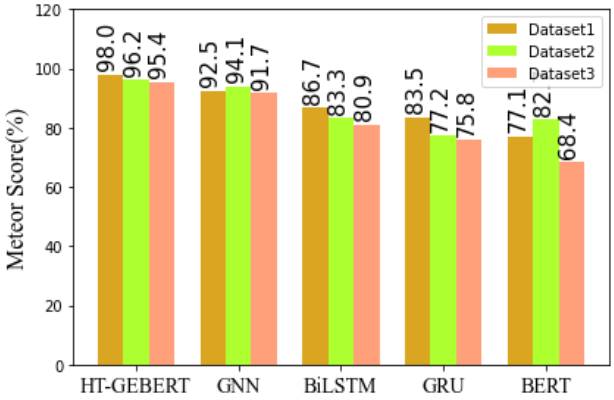

Fig. 14: HT-GEBERT of Meteor Score value compared to other existing algorithms

The HT-GEBERT algorithm demonstrates superior performance compared to other models. Additionally, hyper-tuning parameters such as learning rate, optimizer, and weight decay enhances convergence and reduces overfitting, improving generalization on unseen data. Fig. 13 presents a comparison of Bleu Scores, with the proposed approach achieving values of 81.6%, 83%, and 74% across the datasets. Fig. 14 shows the Meteor Score comparisons, where the proposed method records Meteor Scores of 98%, 96%, and 95% for the three datasets. The HT-GEBERT algorithm outperforms existing algorithms, as its graph-enhanced mechanisms improve model generalization on smaller datasets by leveraging relational information that standard sequence-based representations often miss. Fig. 15 shows the comparison of Character Error Rates (CER). For the proposed approach, the CER values for the datasets are 0.320%, 0.064%, and 0.170%, respectively.

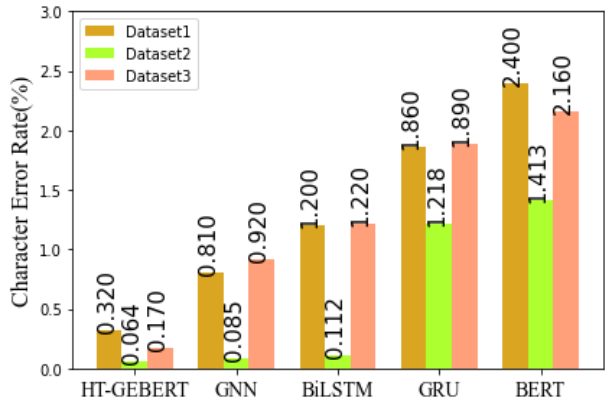

Fig. 15: Comparison of the Character Error Rate of HT-GEBERT with various existing techniques.

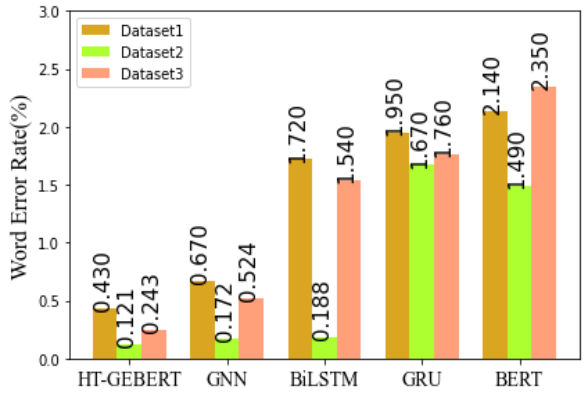

Fig. 16: HT-GEBERT of Word Error Rate value compared to other existing algorithms

Fig. 16 presents the comparison of Word Error Rates (WER), with the proposed approach yielding consistent values of 0.430%, 0.121%, and 0.243% across all datasets. The HT-GEBERT algorithm outperforms comparable existing algorithms, as graph structures capture hierarchical relationships between tokens, sentences, or documents, enabling GEBERT to process complex information more accurately in document classification tasks.

Table 4: Ablation Study for the Proposed Approach

Performance Metrics				
Methods	**Accuracy**	**PPV**	**F1 Score**	**FDR**
Pre-Processing EKE	94%	92.5%	94%	5.5%
EKE-GEBERT	92%	91.06%	91.08%	7.8%
Pre-Processing-HT-GEBERT	91%	92.86%	93.56%	9.67%
GEBERT-Manifold ranking	91.5%	92.78%	92.04%	11.97%

Table. 4, summarizes the findings by comparing the performance metrics of the proposed model under various scenarios involving Keyword Extraction and Word Embedding. The table demonstrates that the model consistently achieves superior results when using EBKE for preprocessing, Keyword Extraction, Word Embedding for optimization and query expansion. Improvements in accuracy, positive predictive value, F1 score, and false discovery rate (FDR) highlight the crucial role these techniques play in enhancing the model's effectiveness and robustness across the analyzed datasets.

Table. 5: Comparison of the proposed with Various State-of-the Art Methods

Performance Metrics	**Proposed (HT-GEBERT)**	**CNN [11]**	**BERT [12]**	**RNN [13]**	**DNN [14]**
Accuracy (%)	94	85	90	82	81
PPV (%)	94.5	80	80	75	75
Error (%)	6	15	10	18	19
F1_Score (%)	94	77	85.6	76.34	73
Hit Rate (%)	93.6	83.7	82	78.5	79.64

Table. 5, presents a comparison between the proposed method and various state-of-the-art techniques. The performance metrics include accuracy, positive predictive value (PPV), error rate, F1 score, and hit rate. The existing techniques used for evaluation comprise Convolutional Neural Networks (CNN), Bidirectional Encoder Representations from Transformers (BERT), Recurrent Neural Networks (RNN), and Deep Neural Networks (DNN). Based on these performance metrics, the proposed model outperforms the existing deep learning models, leading to an improved document retrieval system.

5. Conclusion

The proposed model improves query relevance by utilizing HT-GEBERT for query expansion and employs the manifold ranking algorithm for document retrieval. An automatic query expansion-based document retrieval system enhances user queries by adding relevant terms or phrases, improving the chances of retrieving more relevant documents. This technique helps users find information that they may not have initially considered, increasing the effectiveness of search results. However, it can introduce irrelevant terms, potentially leading to irrelevant results. Additionally, it may rely heavily on the quality of the expansion method, which can vary. Lastly, the system may require more processing time and resources to analyze and expand queries effectively. The user query from the corpus is initially enhanced using the Pseudo Adversarial Embedding (PAE) approach to increase data diversity for robust model training. Subsequently, the augmented text and answer are given to preprocessing techniques, including tokenization, lemmatization, acronym expansion, stop word removal, hyperlink removal, and spell correction, to ensure a consistent format for effective model analysis. Following this, keywords are

extracted using Eccentricity-Based Keyword Extraction (EKE) to identify significant terms. Once the keywords are extracted, the Hyper-Tuned Graph Enhanced Bidirectional Encoder Representation from Transformers (HT-GEBERT) model converts these words into vector form, with its hyper parameters optimally selected through the Artificial Gorilla Troops Optimization Algorithm (AGTOA). Finally, a ranking-based query expansion approach is developed to enhance document retrieval. Additionally, the efficiency of the proposed approach was evaluated against existing techniques such as GNN, BiLSTM, GRU, and BERT. The proposed method achieved an accuracy of 94%, a positive predictive value (PPV) of 94.5%, a hit rate of 93.6%, a miss rate of 6.4%, a false discovery rate (FDR) of 5.5%, and an F1 score of 94%. The results clearly indicate that the proposed method significantly outperforms existing approaches, as illustrated in a graphical comparison. These findings demonstrate that the proposed HT-GEBERT offers a more optimal solution than other existing methods.

Limitation and Future Scope

Hyper-tuned Graph Enhanced BERT-based models for automatic query expansion face several specific limitations. The complexity of hyper parameter tuning, along with the use of graph structures, sustains high computational costs, making them less suitable for large-scale applications. Scalability is another concern, as memory-intensive graph representations limit their practicality for large datasets. In the future, automatic query expansion can be implemented using deep learning models like BERT for contextual understanding, and graph-based approaches to leverage knowledge graphs for semantically related terms. Reinforcement learning will optimize query strategies based on user feedback, while federated learning can enhance personalization without compromising privacy. Additionally, Adaptive Decision Systems can dynamically select effective expansions, and multi-modal integration will enrich query understanding across diverse content types.

Conflict of Interest

The authors declared "No conflict of interest"

References

[1] P. N. R. Okhrati, S. Guan and V. Chang, "Knowledge Graph and Deep Learning-based Text-to-GraphQL Model for Intelligent Medical Consultation Chatbot", *Information systems Frontiers*, Vol. 26, pp. 137-156, 2024. https://doi.org/10.1007/s10796-022-10295-0

[2] G. Singh, N. Mittal and S. S. Chouhan "A deep learning framework for multi-document summarization using LSTM with improved Dingo Optimizer (IDO)", *Multimedia Tools and Applications*, Vol. 83, pp. 69669-69691, 2024. https://doi.org/10.1007/s11042-024-18248-2

[3] V. Deepak, and S. Kumar "Automatic Query Expansion for Enhancing Document Retrieval System in Healthcare application using GAN Based Embedding and Hyper-tuned DAEBERT Algorithm", *Data & Knowledge Engineering*, Vol. 160, art. no. 102468, 2025. https://doi.org/10.1016/j.datak.2025.102468

[4] M. Esposito, E. Damiano, A. Minutolo, G. D. Pietro and H. Fujita, "Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering", *Information Sciences*, Vol. 514, pp. 88-105, 2020. https://doi.org/10.1016/j.ins.2019.12.002

[5] I. Safder , S. U. Hassan , A. Visvizi , T. Noraset, R. Nawaz and S. Tuarob "Deep Learning-based Extraction of Algorithmic Metadata in Full-Text Scholarly Documents", *Information processing and management*, Vol. 57, No. 6, art. no. 102269, 2020. https://doi.org/10.1016/j.ipm.2020.102269

[6] M. Bidoki, M. R. Moosavi and M. Fakhrahmad, "A semantic approach to extractive multi-document summarization: Applying sentence expansion for tuning of conceptual densities", *Information processing and management*, Vol. 57, No. 6, art. no. 102341, 2020. https://doi.org/10.1016/j.ipm.2020.102341

[7] J. Guo, Y. Fan, L. Pang, L. Yang , Q. H. Zamani, C. Wu, W. B. Croft and X. Cheng "A Deep Look into neural ranking models for information retrieval", *Information Processing and Management*, Vol. 57, No. 6, art. no. 102067, 2020. https://doi.org/10.1016/j.ipm.2019.102067

[8] Z. Chu, J. Yu and A. Hamdulla, "A novel deep learning method for query task execution time

prediction in graph database", *Future Generation Computer Systems*, Vol. 112, pp. 534-548, 2020. https://doi.org/10.1016/j.future.2020.06.006

[9] K. Taha, P. D. Yoo, C. Yeun and A. Taha "Text Classification Techniques: A Holistic Review, Observational Analysis, and Experimental Investigation", *Big Data Mining and Analytics*, Vol. 8, No. 3, pp. 624-660, 2025. https://doi.org/10.26599/BDMA.2024.9020092

[10] K. Munir and M. S. Anjum, "The use of ontologies for effective knowledge modelling and information retrieval", *Applied Computing and Informatics*, Vol. 14, No. 2, pp. 116-126, 2018. https://doi.org/10.1016/j.aci.2017.07.003

[11] L. Massai, "Evaluation of semantic relations impact in query expansion-based retrieval systems", *Knowledge Based Systems*, Vol. 283, art. no. 111183, 2024. https://doi.org/10.1016/j.knosys.2023.111183

[12] K. Sugathadasa, B. Ayesha, N. de Silva, A. S. Perera, V. Jayawardana, D. Lakmal and M. Perera, "Legal Document Retrieval using Document Vector Embeddings and Deep Learning", *Computers Science and Engineering*, Vol. 176, pp. 160-175, 2018. https://doi.org/10.48550/arXiv.1805.10685

[13] X. Wang, C. MacDonald and N. Tonellotto, "ColBERT-PRF: Semantic Pseudo-Relevance Feedback for Dense Passage and Document Retrieval", *ACM Transactions on the Web*, Vol. 17, No. 1, art. no. 3, pp. 1-39, 2023. https://doi.org/10.1145/3572405

[14] W. Ou and V. N. Huynh "Conditional variational autoencoder for query expansion in ad-hoc information retrieval", *Information Sciences*, Vol. 653, art. no. 119764, 2024. https://doi.org/10.1016/j.ins.2023.119764

[15] L. M. D. Campos, J. M. F. Luna, J. F. Huete, F. J. R. and N. Bolaños "Information Retrieval and Machine Learning Methods for Academic Expert Finding", *Algorithms*, Vol. 17, No. 2, art. no. 51, 2023. https://doi.org/10.3390/a17020051

[16] H. Bolat and B. Şen "Document Retrieval System for Biomedical Question Answering", *Applied Sciences*, Vol. 14, No. 6, art no. 2613, 2024. https://doi.org/10.3390/app14062613

[17] A. P. Bhopale and A. Tiwari, "Transformer based contextual text representation framework for intelligent information retrieval", *Expect Systems with Applications*, Vol. 238, No. 15, art no. 121629, 2024. https://doi.org/10.1016/j.eswa.2023.121629

[18] M. Kim and P. Kang, "Text Embedding Augmentation Based on Retraining With Pseudo-Labeled Adversarial Embedding", *IEEE Access*, Vol. 10, pp. 8363-8376, 2024. https://doi.org/10.1109/ACCESS.2022.3142843

[19] K. Gurugubelli, S. Mohamed and R. Krishna "Comparative Study of Tokenization Algorithms for End to End Open Vocabulary Keyword Detection", *IEEE International conference on acoustics speech and signal processing*, Vol. 18, pp. 14-16, 2024. https://doi.org/10.1109/ICASSP48485.2024.10445876

[20] R. Hafeez, M. W. Anwar, M. H. Jamal, T. Fatima, J. C. M. Espinosa, L. A. D. López, E. B. Thompson and I. Ashraf "Contextual Urdu Lemmatization Using Recurrent Neural Network Models", *Mathematics*, Vol. 11, No. 2, art. no. 435, 2023. https://doi.org/10.3390/math11020435

[21] T. I. Amosa, L. I. B. Izhar, P. Sebastian, I. B. Ismail, O. Ibrahim and S. L. Ayinla "Clinical Errors From Acronym use in Electronic Health Record", *IEEE Access*, Vol. 11, pp. 59297-59316, 2023. https://doi.org/10.1109/access.2023.3284682

[22] S. M. Uma, O. Koleoso, I. Umoga, M. Alassad and N. Agarwal "The Multi-attribute impact of hyperlinks in blogs: an emotion-centric approach", *Social Network Analysis and Mining*, Vol. 14, art. no. 134, 2024. https://doi.org/10.1007/s13278-024-01295-w

[23] G. Song, Z. Wu, G. Pundak, A. Chandorkar, K. Joshi and X. Velez, "Contextual Spelling Correction with Large Language Models", *IEEE o Automatic Speech Recognition and understanding workshop*, Vol. 23, pp. 16-20, 2024. https://doi.org/10.1109/ASRU57964.2023.10389637

[24] N. Rajkumar, T. S. Subashini, K. Rajan and V. Ramalingam "Tamil Stop word Removal Based on Term Frequency", *Data Engineering and Communication Technology*, Vol. 1079, pp. 21-30, 2020. https://doi.org/10.1007/978-981-15-1097-7_3

[25] D. A. V. Oliveros, P. S. Gomes, E. E. Milios and L. Berton "A multi-centrality index for graph-based

keyword extraction", *Information Processing & Management*, Vol. 56, No. 6, art. no. 102063, 2019. https://doi.org/10.1016/j.ipm.2019.102063

[26] Y. Yang and X. Cui, "Bert-Enhanced Text Graph Neural Network for Classification", *Entropy*, Vol. 23, No. 11, art. no. 1536, 2019. https://doi.org/10.3390/e23111536

[27] M. A. E. Dabah, M. H. Hassan, S. Kamel and H. M. Zawbaa "Robust Parameters Tuning of Different Power System Stabilizers Using a Quantum Artificial Gorilla Troops Optimizer", *IEEE Access*, Vol. 10, pp. 82560-82579, 2022. https://doi.org/10.1109/ACCESS.2022.3195892

[28] A. Ahmed and S. J. Malebary "Query Expansion Based on Top-Ranked Images for Content-Based Medical Image Retrieval", *IEEE Access*, Vol. 8, pp. 194541-194550, 2020. https://doi.org/10.1109/ACCESS.2020.3033504

[29] https://huggingface.co/datasets/bookcorpus

[30] https://huggingface.co/datasets/lapp0/query_expansion

[31] https://ir-datasets.com/

***