International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V74I5P104

Volume 74 Issue 5, 20-35, May 2026

Original Article

An Autoregressive Secant Optimization-Based Dual-View Retrieval-Augmented LLM Framework for Question Answering

Amit Virmani[1], Alok Kumar[2], Vineeta Singh[3]

[1,2,3]*Department of Computer Science and Engineering, School of Engineering and Technology (Formerly known as UIET Kanpur), Chhatrapati Shahu Ji Maharaj University, Kalyanpur, Kanpur, Uttar Pradesh, India.*

[1]*Corresponding Author: amitvirmani@csjmu.ac.in*

Received: 18 March 2026 Revised: 23 April 2026 Accepted: 14 May 2026 Published: 29 May 2026

Abstract *- A Question Answering System (QAS) is an Artificial Intelligence (AI) system that automatically understands a user's question in natural language and provides a precise list of documents that are retrieved. These systems face several difficulties, like understanding context, handling ambiguity in language, and generalization issues. Therefore, a new model, named Autoregressive Secant Optimization-based Retrieval-Augmented Generation Dual View Network (AuRSO-RAGNet), is established for question answering. The proposed model is based on the Taylor-aware Query-RAG Dual View Network with Large Language Model (TQ-RAGNetLLM). The TQ-RAGNet is designed by modifying the Passage Question Answering-Retrieval Augmented Generation (PQA-RAG) approach's learning rule employing Taylor-Aware Neighbor Mean Square Loss (TANMSL). Answer generation is performed by processing the input passage and questions using PQA-RAG, which comprises two main components: a Question-Dual View Modeling (Q-DVM) and a Question-RAG (Q-RAG) Layer. The overall model is trained using the AuRSO for increasing the performance, where the AuRSO is designed by fusing a Conditional Autoregressive Value-at-Risk (CAViaR) as well as a Secant Optimization Algorithm (SOA). Moreover, experimental outcomes demonstrate that the AuRSO-RAGNet attained a superior recall of 98.248%, precision of 97.840%, and an F1-score of 98.043%.*

Keywords *- Deep Learning, Question Answering System, Large Language Modelling, Retrieval-Augmented Generation, Natural Language Processing.*

1. Introduction

Human beings are inherently curious, posing questions to acquire knowledge and seamlessly integrating that knowledge into everyday life. Questions serve as the foundation of learning, yet several learners struggle with posting inquiries [1]. With the rapid advancement of human-computer interaction and technology, interest in affective computing has grown meaningfully. The continuous generation of large-scale data requires effective incorporation and querying of heterogeneous information sources. Natural Language Processing (NLP) has developed as a hopeful solution for representing complex queries that often involve uncertainty. This progression has led to the development of QAS, where questions are mapped to equations and corresponding answers. Question answering itself is a rapidly expanding research domain and drawing upon procedures from Information Retrieval, Information Extraction, and NLP [2]. A question answering approach is established to extract relevant answers from a given context in response to a natural language query. The chief objective of such systems is to save effort and time by providing concise and summarized responses that address the user's question directly [3]. Automated question generation schemes can support learners in analyzing their level of expertise and expanding their knowledge by helping them clarify their queries. Beyond education, such systems have broad applications, like scholarly research and placement activities, where they can be utilized to produce meaningful questions [1]. QAS allows users to pose queries in natural language and receive direct, concise responses [4].

QAS schemes are widely combined into familiar interfaces and search engines, where they excel at providing answers to straightforward factual queries. Nevertheless, when confronted with more complex questions, systems often return only a sequence of text snippets, leaving users to sift through the information to trace the precise answer [5] [4]. Recently, the adoption of Machine Learning (ML) algorithms has grown rapidly, as they are capable of addressing complex issues across diverse domains. Deep Learning (DL) has proven particularly effective in automating intricate tasks. Unlike traditional approaches, DL methods attain strong

performance without relying on manual feature engineering or costly external resources [6]. DL methods have demonstrated remarkable success when applied to QAS. However, they require analysis of large-scale databases for effective training. For instance, Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) are commonly employed to classify questions within QAS techniques [7].

Meanwhile, Generative AI has recently attracted significant attention from both academia and industry, owing to its emergent capabilities as well as transformative potential across diverse domains. Among its wide-ranging applications, LLMs signify the most prominent example, demonstrating the ability to facilitate human-computer interaction, produce coherent text, and support complex tasks [8]. LLMs have emerged as powerful tools for enabling chat-based interactions and providing conversational responses to natural language inputs. Despite their success, these approaches face notable limitations, particularly when the objective is to deliver explainable and reliable answers [9].

1.1. Research Gap and Problem Statement

Traditional QASs predominantly depend on keyword matching and shallow retrieval strategies, generating responses based on text similarity rather than deeper semantic reasoning. The rapid growth of digital information demands systems that move beyond simple retrieval toward intelligent and context-aware reasoning. However, most existing methods still suffer from poor generalization and low reliability. Therefore, a new model, namely AuRSO-RAGNet, is introduced for question answering.

1.2. Contribution

- A novel model, namely AuRSO-RAGNet, for question answering. Here, the AuRSO-RAGNet is devised based on the TQ-RAGNet LLM. The TQ-RAGNet is established by adjusting PQA-RAG's learning rule by TANMSL.
- The training of the overall model is conducted using AuRSO, a method that combines SOA with CAViaR to optimize system accuracy and efficiency.

The paper is further organized in the ensuing sections: Section 2 reviews the competing approaches for question answering. Section 3 presents the devised AuRSO-RAGNet in detail. Section 4 provides experimental outcomes and an assessment of AuRSO-RAGNet. Section 5 concludes the study.

2. Literature Review

A model LLM and Bidirectional Encoder Representations from Transformers- Knowledge-Based-Question-and-Answer (LB-KBQA) model was designed by Zhao, Y., *et al.* [8]. This model reduced errors, generated more readable answers, balanced speed, and adaptability. However, this approach failed to expand the query library with unstructured and structured data and minimize reliance on manual updates. Zhang, J. *et al.* [10] established a Hierarchical Semantic Matching-Question Answering (HSM-QA) system. This approach minimized computational overhead, showed strong generalization ability, and had higher efficiency.

Meanwhile, this technique was unsuccessful in merging hierarchical semantic matching along external knowledge to tackle tasks that involve multi-hop reasoning, domain-specific expertise, or contextual understanding. Wang, H., [11] proposed an Automated Question and Answer (AQA) model for online education. This method enhanced semantic similarity, reduced redundancy, and improved relevance in retrieved answers. However, this technique failed to handle much larger question banks efficiently, ensuring performance remained high with increased data volume. A Distilled-BERT (DistilBERT) was proposed by Alzubi, J.A. *et al.* [12] for QAS. This technique had good accuracy, lowered computational cost, and had a faster response time. Nonetheless, this model failed to develop more interactive and user-friendly interfaces for healthcare professionals and researchers. Swathi, B.P. *et al.* [13] developed a Bidirectional Gated Recurrent Unit (Bi-GRU) for educational QAS. This technique had better generalization, reduced computational complexity, and robustness to query variability. However, this framework failed to handle ambiguous queries that overlap categories, improving flexibility and accuracy. Mengqi Li and Rufu Qin [22] implemented DualGraphRAG, a dual-view graph-enhanced Retrieval-Augmented Generation model for improving QA performance and computational efficiency. The model effectively supported complex multi-hop reasoning. However, the framework did not investigate domain-specific adaptations.

The existing QAS approaches achieved improvements in semantic understanding, retrieval efficiency, and answer generation. However, most existing models still face limitations in contextual reasoning, ambiguity handling, and generalization performance when processing complex queries. In addition, several methods exhibit higher computational complexity and reduced adaptability in large-scale question answering environments. Therefore, an efficient framework is required to improve retrieval accuracy, contextual learning, and answer generation performance. To provide a solution to these problems, the AuRSO-RAGNet has been developed for effective question answering.

3. Proposed Autoregressive Secant Optimization-Retrieval-Augmented Generation Dual View Network for Question Answering

This paper establishes a novel model, named AuRSO-RAGNet, for QAS. The devised AuRSO-RAGNet is based on TQ-RAGNet LLM. Additionally, TQ-RAGNet is established by adjusting PQA-RAG's learning rule by TANMSL. Then,

answer generation is accomplished by processing the input questions and passage utilizing PQA-RAG, which contains two main components: Q-DVM [14] for passage processing and Q-RAG Layer [14] for processing questions and passages for answer generation. The proposed framework jointly utilizes Q-DVM and Q-RAG layers to strengthen the interaction between query understanding and retrieval generation processes. Once the answer is generated, the generated answer, passage, and question are subjected as input to prompt generation. This is then refined employing LLM. Moreover, the training of the overall model is conducted using AuRSO, a method that combines SOA [15] with CAViaR [16] to optimize system accuracy and efficiency. In Fig. 1, the architecture view of the AuRSO-RAGNet is depicted.

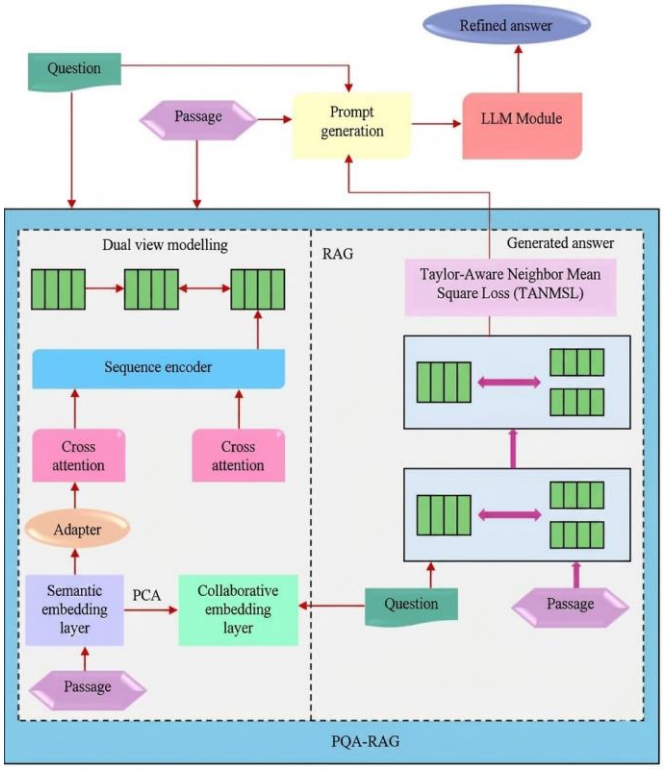


Fig. 1 Diagrammatic view of AuRSO-RAGNet for QAS

3.1. Question and passage data acquisition

Let us consider a database $N$with b_a number of records with question (R) and passage (D), and is expressed as,

$$N = \{R, D\} \tag{1}$$

The input question for question answering is obtained using a specified database, which is given by

$$R = \{x_1, x_2, \ldots\ldots, x_b\} \tag{2}$$

where $x_b$indicates the b^{th} question and $b$refers to the overall count of questions.

The input passage for the question and answering is attained from a corresponding dataset, expressed as,

$$D = \{y_1, y_2, \ldots\ldots, y_d\} \tag{3}$$

Here, $y_d$represents the d^{th} passage and $d$signifies the overall count of passages.

3.2. PQA-RAG Modelling

In the PQA-RAG model [14], generative modeling is augmented with retrieval to ensure that answers are supported by suitable passages. The model retrieves contextually relevant information from knowledge sources, reducing reliance on parametric memory. This design improves the accuracy of responses and delivers improved outcomes for complex question-answering applications. The question-text pair $P = \{y_d, x_b\}$is fed as input to the PQA-RAG, which contains two main components: Q-DVM [14] for passage processing and Q-RAG Layer [14] for processing questions and passages for answer generation. The procedure for generating answers is detailed as follows.

3.2.1. Q-DVM Module

Q-DVM comprises a semantic branch and a collaborative branch. The semantic branch leverages LLM embeddings, dimensional adapters, and cross-attention to encode question–passage content. The collaborative branch builds embeddings from interaction patterns and encodes them with cross-attention. Their results are incorporated to yield a comprehensive representation for generating answers.

Semantic View Modeling

Semantic-view modeling organizes questions and passages into prompts that leverage LLM semantic understanding. The resulting embeddings are stored as fixed vectors to reduce computational cost. These vectors are subsequently fed into semantic and collaborative branches, yielding question-aware representations optimized for accurate answer generation.

Collaborative View Modelling

The Collaborative-view branch model generates a trainable embedding layer from question-answer pairs, leveraging collaborative signals from historical question-passage contexts. These embeddings collect co-occurrence patterns, while Principal Component Analysis (PCA)-based dimensionality reduction is employed to stabilize and align the passage embeddings against pretrained vectors.

Two-Level Fusion

Collaborative and semantic perspectives are incorporated through a two-level fusion mechanism. At the sequence level, cross-attention gathers the interactions among collaborative and semantic sequences, while weight matrices define the alignment procedure.

3.2.2. Retrieval-Integrated Self-Distillation

It improves the creation of answers by channeling semantic knowledge from informative passages into the sequence encoder. As a variant of knowledge distillation, self-distillation boosts the model's precision while maintaining efficiency, since it requires no auxiliary networks. The procedure of retrieval augmented self-distillation includes retrieval of answers as well as the self-distillation module and is described as follows.

Retrieval of Answers

Answer generation benefits greatly from recognizing prior responses that exhibit semantic resemblance to the current query. Every answer is encoded as a semantic embedding vector based on its question–passage pair. By applying adjusted cosine similarity, the model retrieves those answers that most closely match the target, forming a pool of semantically consistent knowledge for transfer.

Self-Distillation

Self-distillation is designed for identifying semantically similar answers and uses them as a source of guidance. The target passage itself is modeled using a dual-view approach, which integrates semantic embeddings from LLMs with collaborative embeddings using interactions between users and items. Finally, to form the mediator, the system applies mean pooling over the retrieved similar passages, producing a distilled representation that guides the answer generation process with balanced semantic and collaborative knowledge. The answer produced by PQA-RAG is signified as B.

3.2.3. Taylor-Aware Neighbor Mean Squared Loss

TQ-RAGNet LLM is established by modifying PQA-RAG's learning rule using TANMSL. Moreover, TANMSL is developed by merging Neighbour loss [17], Mean Squared Error (MSE) [18], and Taylor series [19]. MSE evaluates the mean squared error between actual and predicted outcomes, with the squaring step ensuring that larger errors are penalized.

In contrast, neighbor measures discrepancies among neighboring predictions, which encourages local consistency and smoothness. TANMSL loss function integrates the Taylor-series-based mechanism to simplify function approximation and maintain consistent gradient updates during tuning. The TANMSL loss-based learning rule is represented as:

$$T_S^{H+1} \leftarrow T_S - I\left\{-\frac{1}{F}\sum_{a=1}^{F} D * \begin{bmatrix} \frac{5}{2}\left(\frac{B_a^*}{B_a(P_a,T_{H-1})} - \frac{(1-B_a^*)}{1-B_a(P_a,T_{H-1})}\right) - \\ 2\left(\frac{B_a^*}{B_a(P_a,T_{H-2})} - \frac{(1-B_a^*)}{1-B_a(P_a,T_{H-2})}\right) \\ +\frac{1}{2}\left(\frac{B_a^*}{B_a(P_a,T_{H-3})} - \frac{(1-B_a^*)}{1-B_a(P_a,T_{H-3})}\right) \end{bmatrix} + \frac{V}{F}\sum_{a=1}^{F} \begin{bmatrix} -5\left(B_a^* - B_a(P_a,T_{H-1})\right) \\ +4\left(B_a^* - B_a(P_a,T_{H-2})\right) \\ -\left(B_a^* - B_a(P_a,T_{H-3})\right) \end{bmatrix}\right\} \quad (4)$$

where F refers to observation count, B_a specifies the true output of PQA-RAG, $\hat{B}_a$ represents the expected output of PQA-RAG, a specifies the index of every observation, T represents the weight, S indicates the weight index, D is the weight assigned to the neighbor loss, I refers to the learning rate, and V designates a constant.

3.2.4. Training Using Autoregressive Secant Optimization

Here, the training of the overall LLM model is accomplished by using AuRSO, which is established by incorporating CAViaR [16] and SOA [15]. SOA is a mathematics-inspired metaheuristic designed for efficient global optimization and is derived from the classical Secant Method, which adapts this principle into a population-based optimization. SOA converges quickly, balances exploration and exploitation, avoids local optima, and is robust for high-dimensional problems. CAViaR is a financial risk model that is designed to estimate the value of risk utilizing quantile regression, without relying on restrictive assumptions like normality or independent returns. It provides a dynamic, data-driven way to measure market risk. CAViaR is merged with SOA to enhance convergence and stability when training complex risk models. This pairing leads to better performance improvement in question answering. The algorithmic steps of AuRSO are detailed below.

Initialization

Initialization involves creating random candidate solutions within the defined search space and is expressed as,

$$A_g = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_E \end{bmatrix}, g = 1,2,\ldots,E \quad (5)$$

Here, A_g represents the solution vector, E denotes the dimension, and g refers to the candidate solution. The algorithm is initialized by scattering random solution vectors within the domain, where the components of each vector are articulated as,

$$A_g = h + k * (j - h) \quad (6)$$

Here, h indicates lower limit, j refers to upper limit, and k represents a random number generated within the range $[0,1]$.

Fitness Calculation

The fitness is computed using the TANMSL loss function and is derived in expression (4).

Exploration (Updating Rule Phase)

The first operator in SOA directs the movement of vectors, aiming to enhance search effectiveness and guide them toward optimal locations within a solution space. The procedure initiates with two initial candidate solutions, which act as starting points. From there, SOA iteratively advances to

new locations with a constrained trajectory, refining the search path and progressively steering the population toward better solutions. The new updated position (A_{new}) after applying the operator is given by,

$$A_{new} = A_{best} - \frac{\left(\left(\frac{m(A_{best})-m(A_g)}{2\Delta A}\right)*(A_g - A_{best})\right)}{\left(\left(\frac{m(A_p)-m(A_g)}{2\Delta A}\right)-\left(\frac{m(A_{best})-m(A_g)}{2\Delta A}\right)\right)} \quad (7)$$

Here, m denotes a quadratic equation, L refers to the current candidate solution, Crepresents another candidate solution in the search space, ΔAsignifies a small perturbation utilized to approximate the derivative of m, A_{best}indicates the current best solution, $A_g$represents the g^{th} candidate solution in the population, and$A_p$denotes another candidate solution used to provide diversity in the update.

Stochastic Exploitation Stage

An expansion factor is introduced to encourage promising and diverse solutions within a search space, while also expanding into new areas and minimizing the risk of stagnation in local optima. This mechanism broadens the exploration procedure and ensures the algorithm maintains variety and continues to search effectively for global optima. The new location in a search space is expressed as,

$$A_g(w+1) = A_g(w) + G * (A_{closest} - A_k) + p * \left(A_{farthest} - A_{Q_1}\right) \quad (8)$$

Here, Gspecifies a scaling factor that influences the difference between $A_{closest}$ and the random candidateA_k, the solution closest to $A_g$is represented as$A_{closest}$, $p$is a random number, $w$is the iteration count, k symbolizes a random number ranging from [0,1], and $A_{farthest}$ specifies the solution farthest from A_g.

CAViaR [16] improves risk measurement by modeling quantiles directly, adapting to volatility, avoiding distributional assumptions, and offering robust statistical validation. From caviar,

$$A_g(w) = \beta_0 + \sum_{e=1}^{l} \beta_c\, A(w-e) + \sum_{q=1}^{i} \beta_q\, f\big(A(w-q)\big) \quad (9)$$

Here, $\beta_0$represents a bias term, $\beta_c$designates a vector of unknown parameters, $\beta_q$specifies learnable weights, and $f(.)$indicates a non-linear transformation function.

Let $l = i = 2$, then

$$A_g(w) = \beta_0 + \beta_1 A_g(w-1) + \beta_2 A_g(w-2) + \beta_1 f\left(A_g(w-1)\right) + \beta_2 f\left(A_g(w-2)\right) \quad (10)$$

Substituting expression (11) in expression (9),

$$A_g(w) = \begin{pmatrix} \beta_0 + \beta_1 A_g(w-1) + \beta_2 A_g(w-2) \\ +\beta_1 f\left(A_g(w-1)\right) + \beta_2 f\left(A_g(w-2)\right) \end{pmatrix} + G * (A_{closest} - A_k) + p * \left(A_{farthest} - A_{Q_1}\right) \quad (11)$$

The expression (11) represents the updated expression of AuRSO.

After assessing solutions, the search space is explored more efficiently, balancing exploitation of the ideal solution with exploration of alternative directions. The updated point in a search space A_{new} is given by,

$$A_{new} = A_{best} + G * \left(A_{new} - A_g\right) + \mathbb{R} * \left(A_g - A_{Q_2}\right) \quad (12)$$

Here, $A_{Q_1}$and A_{Q_2}refers to randomly chosen solutions from a present population and $\mathbb{R}$ is a random search function.

Fitness Re-Calculation

After updating, the fitness is assessed using the loss function. The strongest solution is retained, while weaker ones may revert to earlier states or undergo corrective adjustments based on boundary rules.

Termination

Following the update of each Secant's position, fitness is evaluated, and ideal solutions are graded. The strongest performers are retained to approximate the optimal solution, while the iteration counter advances. At the end of the loop, the best solution is identified. AuRSO improves answer generation by refining solutions iteratively, accelerating convergence, balancing exploitation and exploration, improving accuracy, and ensuring precise outputs.

3.3. Prompt Generation

The prompt generator in PQA-RAG takes three inputs: the question (x_b), the supporting passage (y_d), and the preliminary answer (B), and synthesizes them into a refined prompt. This structured prompt is then provided to the LLM, which uses it to produce a final answer that is both precise and contextually grounded. The purpose of this step is to optimize the input format for the LLM, ensuring that the model can generate more accurate responses. The produced prompt is indicated by J.

3.4. LLM-Guided Refined Response Generation

The produced prompt J serves as LLM's input, GPT-I, guiding the model to produce precise, context-aware answers. This refinement step ensures the LLM concentrates on critical information. GPT-I is first pre-trained on large-scale unlabeled text, later fine-tuned with a small supervised dataset for task-specific accuracy, and to avoid data labeling costs. The refined answer generated from the LLM model is denoted as K.

4. Result and Discussion

Here, a detailed assessment of the established AuRSO-RAGNet is presented in comparison with prevailing methods. The supremacy of AuRSO-RAGNet is assessed utilizing key metrics, namely recall, F1-score, and precision.

4.1. Experimental Environment

The AuRSO-RAGNet for generating question answering is accomplished via the Python tool. Table 1 lists the chosen parameter values of PQA-RAG as well as AuRSO.

Table 1. Experimental parameters of question answering and AuRSO

PQA-RAG		AuRSO	
Parameters	**Ranges**	**Parameters**	**Ranges**
Epoch	120	Population size (g)	50
Batch size	16	Decision variable (E)	30
Optimizer	AuRSO	Maximum Iterations	200
kernel size	(3,3)	k	Random value between (0,1)
Dropout	0.2	β (Constant)	1.5
Loss	TANMSL	-	-
Activation	ReLU	-	-

4.2. Dataset Description

Here, the passages and questions from two databases, namely XQuAD Dataset (dataset 1) [20] and MultiLingual Question Answering (MLQA) database (dataset 2) [21], are used for question answering, and the descriptions are illustrated beneath.

4.2.1. Dataset 1

Dataset 1 [20] serves as a benchmark dataset designed to evaluate cross-lingual question answering systems. It comprises 1,190 question-answer pairs and 240 paragraphs drawn using samples from the SQuAD v1.1 dev set, professionally translated into several languages. This provides fully parallel data across eleven languages. Furthermore, this dataset is distributed in the standard SQuAD JSON format.

4.2.2. Dataset 2

Database 2 [21] is a benchmark dataset developed to evaluate cross-lingual question answering performance. It comprises 5,000+ SQuAD-format extractive QA pairs covering seven different languages: Simplified Chinese, Vietnamese, Spanish, Hindi, Arabic, English, and German. The dataset is extremely parallel, with most question-answer pairs aligned across an average of four languages. MLQA is released in both development and test partitions, and all files comply with the SQuAD JSON format standard.

4.5. Experimental Outputs

The sample passages and questions of AuRSO-RAGNet for question answering are revealed in Figure 2. The sample question, passage, and its respective answer generated from dataset 1 are shown in Figure 2a. Likewise, in Figure 2b), the sample question, passage, and answer produced by them utilizing dataset 2 are portrayed.

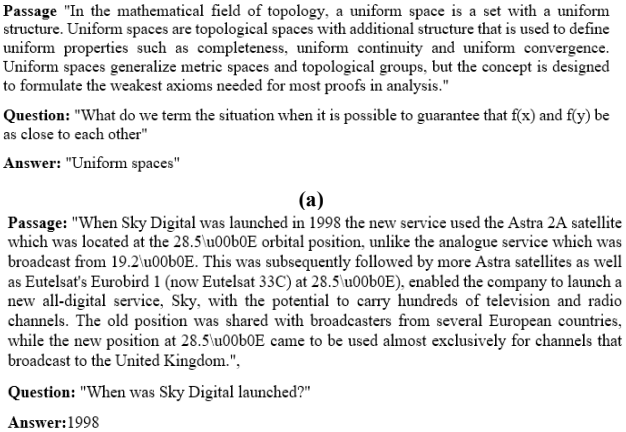


Fig. 2 Experimental outputs of AuRSO-RAGNet for question answering, a) Dataset-1, b) Dataset-2

4.5.1. Experimental Output of Failure Case

The sample failure case result is given in Figure 3, and the sample failure case result using dataset-1 is provided in Figure 3a), and dataset-2 is in Figure 3b).

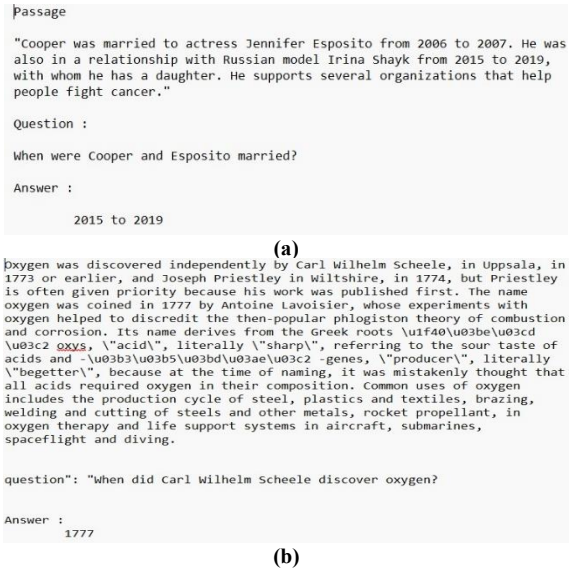


Fig. 3 Experimental outputs of failure case, a) Dataset-1, b) Dataset-2

4.6. Assessment

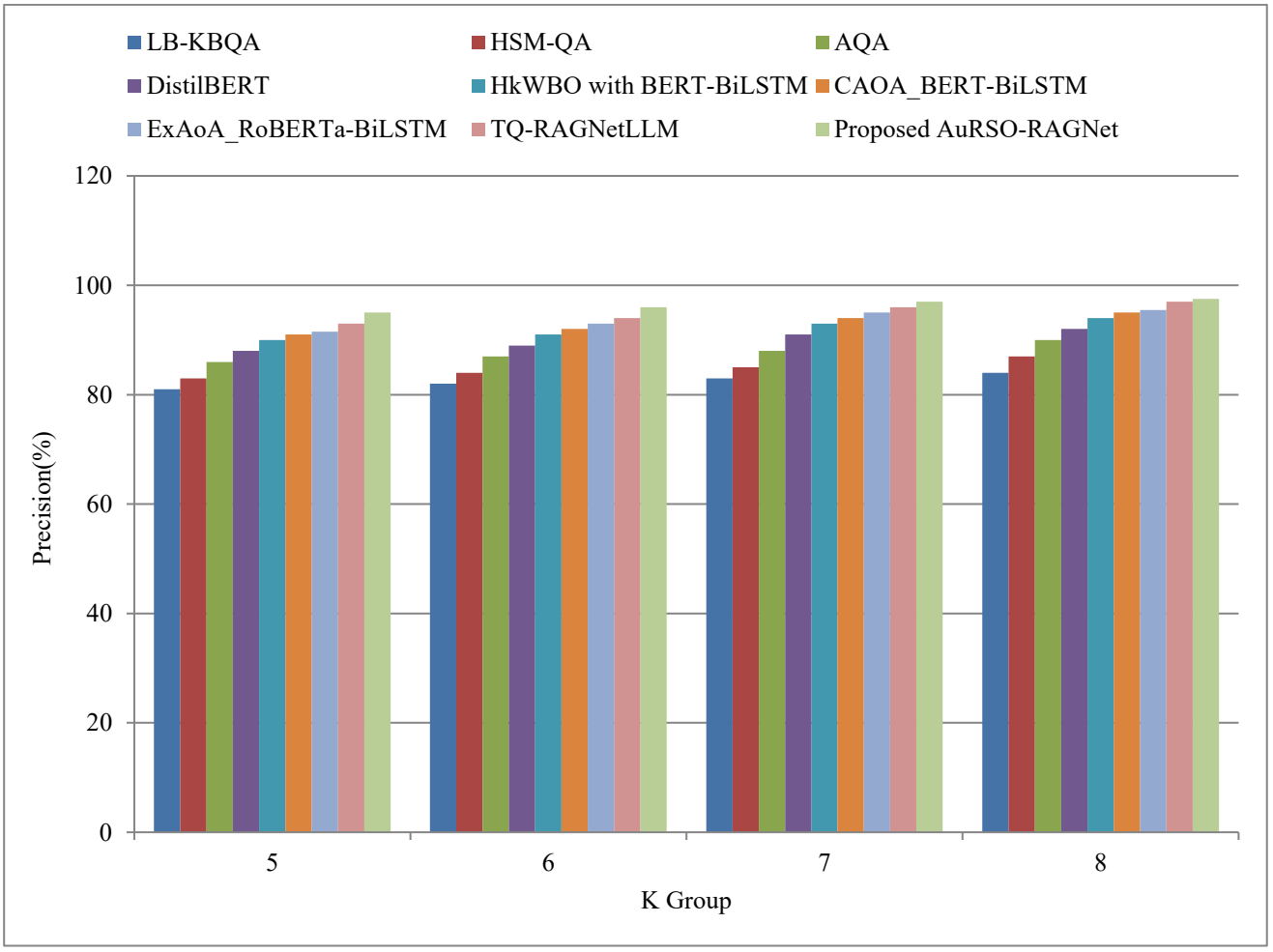
The efficacy of AuRSO-RAGNet for generating question answering is assessed using databases 1 and 2 through the K group, and is detailed below. Moreover, the developed AuRSO-RAGNet is compared against prevailing techniques,

namely LB-KBQA [8], HSM-QA [10], AQA [11], DistilBERT [12], Hiking Wolf Bird Optimization with Bidirectional Encoder Representations from Transformers-Bidirectional Long Short-Term Memory (HkWBO with BERT-BiLSTM), Caviar Addax Optimization Algorithm_BiLSTM (CAOA_BERT-BiLSTM), Exponential-Driven Aphids Algorithm_Robustly Optimized Bidirectional Encoder Representations from Transformers Pre-training Approach-BiLSTM (ExAoA_RoBERTa-BiLSTM), and TQ-RAGNetLLM.

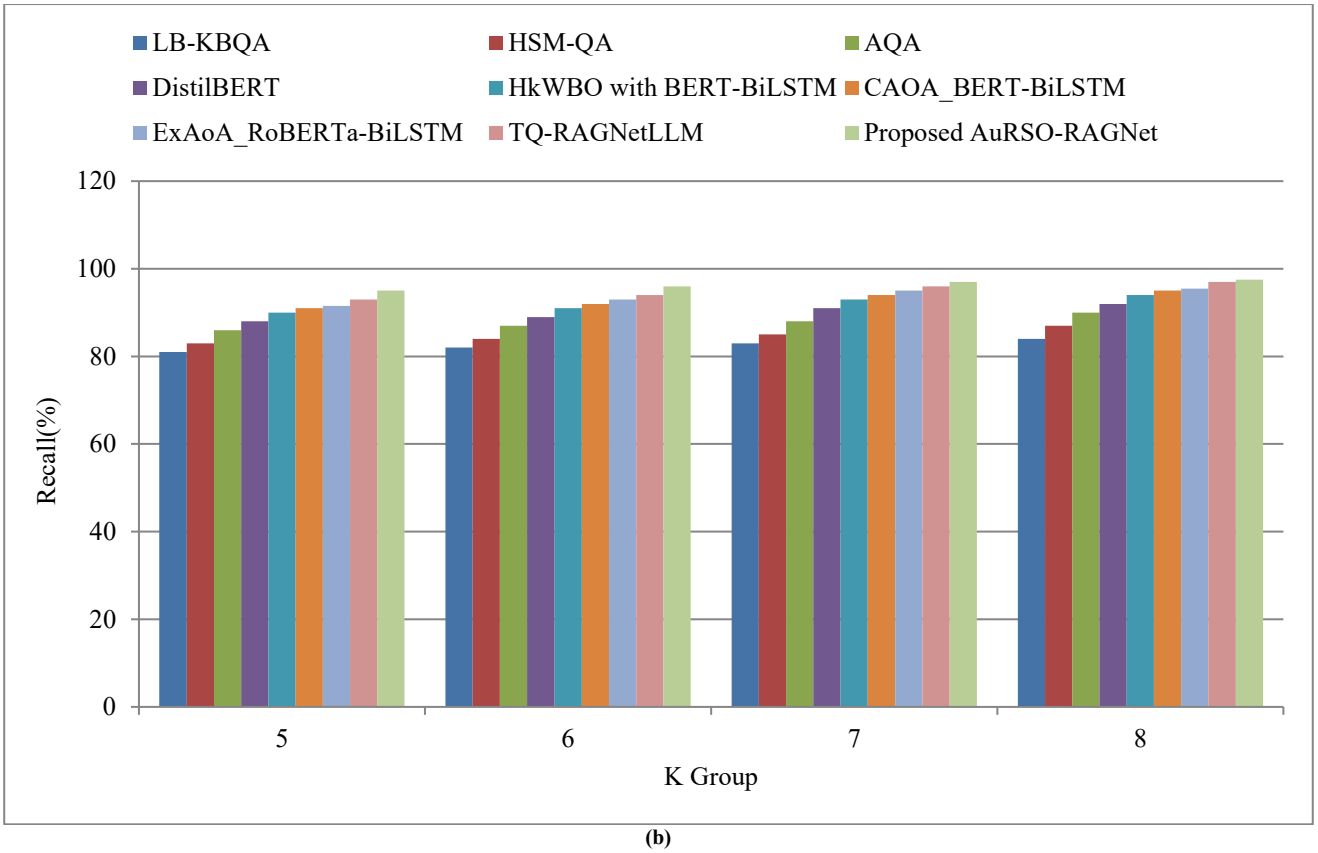
4.6.1. Dataset 1

Fig. 3 depicts the K group estimation of AuRSO-RAGNet for question answering for dataset 1. In Fig. 3a), the K group evaluation of AuRSO-RAGNet with precision is presented. With the K group of 5, the precision recorded by AuRSO-RAGNet is 95.138%, whereas competing models, like HSM-QA, is 83.295%, LB-KBQA is 81.162%, DistilBERT is 88.273%, AQA is 86.183%, HkWBO with BERT-BiLSTM is 90.273%, CAOA_BERT-BiLSTM is 91.628%, TQ-RAGNetLLM is 93.249%, and ExAoA_RoBERTa-BiLSTM is 91.910%. The precision of AuRSO-RAGNet is improved by 6.71% than HSM-QA. The K group estimation of AuRSO-RAGNet with recall is depicted in Fig. 3b. The recall recorded by the AuRSO-RAGNet is 97.323%, and competing techniques, such as AQA, is 88.994%, HSM-QA is 86.488%, DistilBERT is 91.333%, CAOA_BERT-BiLSTM is 95.159%, LB-KBQA is 84.044%, ExAoA_RoBERTa-BiLSTM is 95.624%, HkWBO with BERT-BiLSTM is 93.988%, TQ-RAGNetLLM is 96.631%, when K group is 10. In comparison to AQA, the recall of AuRSO-RAGNet is improved by 5.23%.

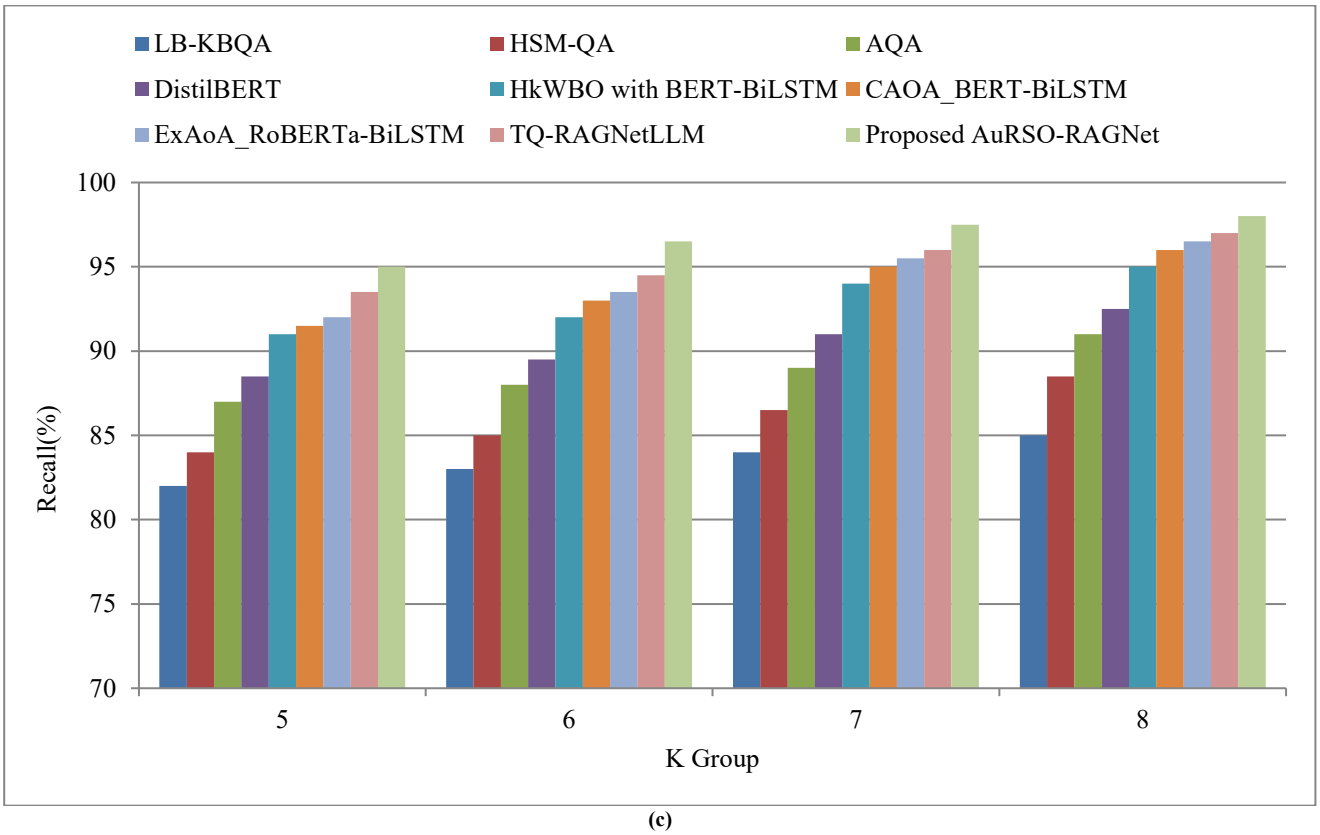
In Fig. 3c), the K group analysis of AuRSO-RAGNet with F1-score is represented. By assuming a K group of 7, the F1-score obtained by traditional frameworks, like HSM-QA, is 86.488%, AQA is 88.994%, CAOA_BERT-BiLSTM is 95.159%, DistilBERT is 91.333%, LB-KBQA is 84.044%, HkWBO with BERT-BiLSTM is 93.988%, ExAoA_RoBERTa-BiLSTM is 95.624%, TQ-RAGNetLLM is 96.631%, and the proposed AuRSO-RAGNet is 97.323%. The AuRSO-RAGNet demonstrates a higher F1-score by 6.12% than LB-KBQA.



(a)



(b)

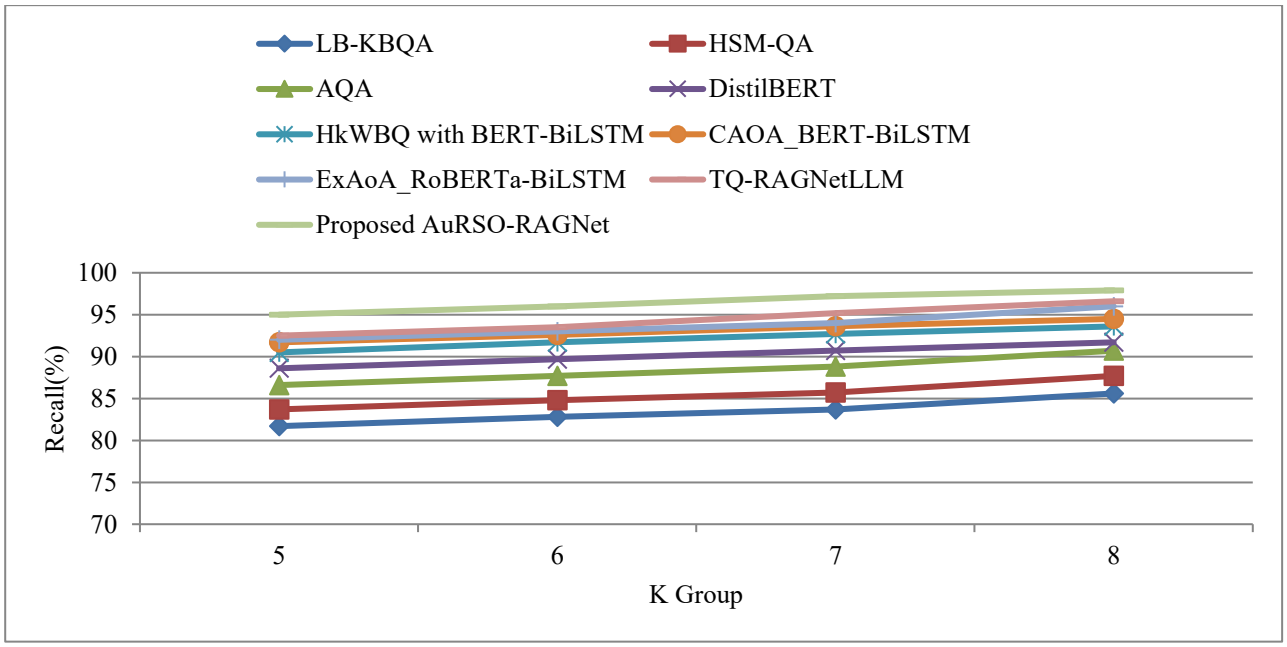


(c)

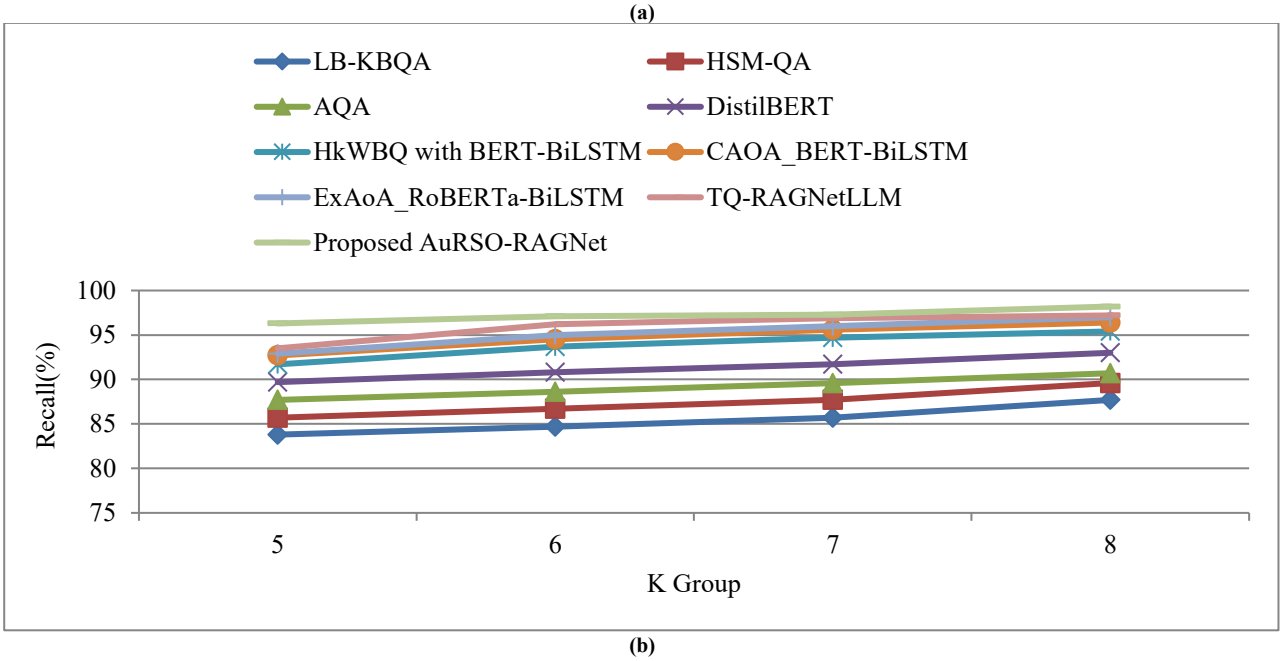
Fig. 3 K group valuation of AuRSO-RAGNet for generating question answering for dataset 1: a) precision, b) recall, and c) F1-score

4.6.2. Dataset 2

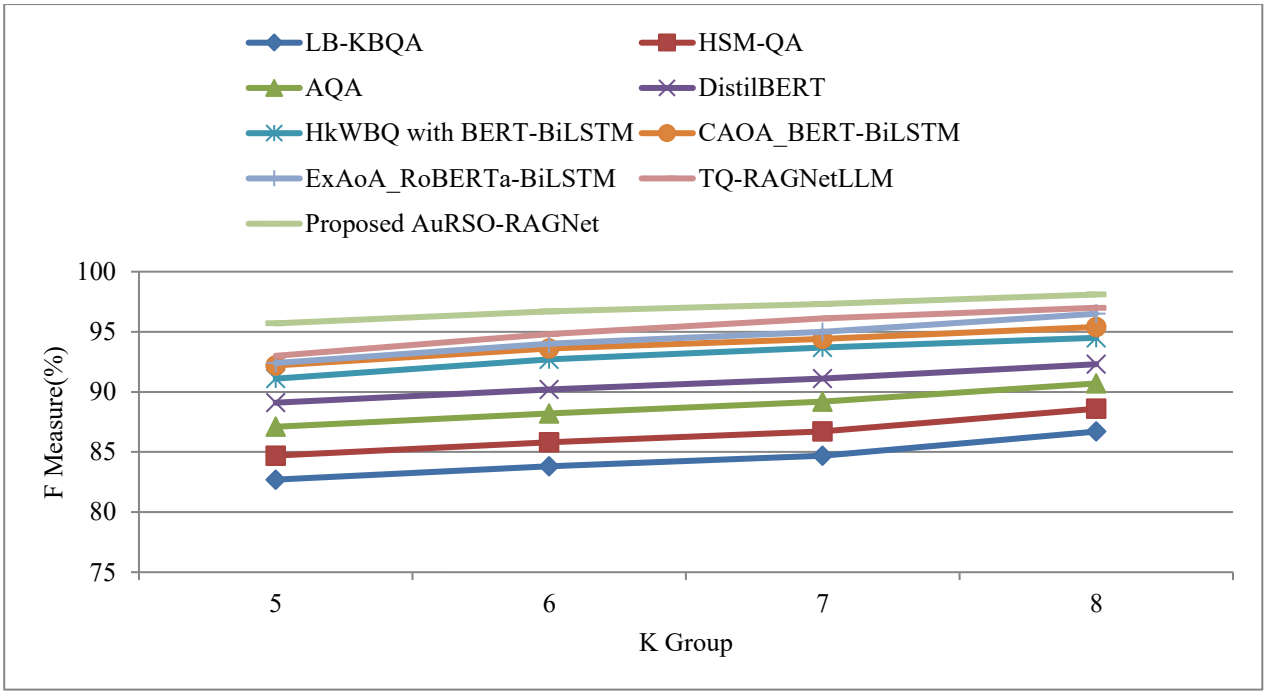
The K group assessment of AuRSO-RAGNet for question answering employing dataset 2 is presented in Fig. 4. The K group estimation of AuRSO-RAGNet with precision is visualized in Fig. 4a). At K group is 8, precision achieved by AuRSO-RAGNet is 97.840% and competing techniques is 85.595%, 87.704%, 91.704%, 93.595%, 90.707%, 94.483%, 96.619%, and 95.916%. Here, the precision attained by AuRSO-RAGNet is improved over that of LB-KBQA by 4.78%. Fig. 4b) depicts the K group evaluation of AuRSO-RAGNet regarding recall. At the K group is 5, the recall measured by competing methods, including AQA is 87.705%, HSM-QA is 85.705%, CAOA_BERT-BiLSTM is 92.688%, LB-KBQA is 83.726%, DistilBERT is 89.705%, HkWBO with BERT-BiLSTM is 91.704%, ExAoA_RoBERTa-BiLSTM is 92.917%, TQ-RAGNetLLM is 93.473% and AuRSO-RAGNet is 96.239%. AuRSO-RAGNet delivers better recall by 4.06% than AQA. In Fig. 4c), the K group assessment of the F1-score is portrayed. The F1-score measured by conventional schemes is 94.016%, 85.745%, 83.750%, 90.257%, 92.694%, 93.546%, 88.158%, and 94.826%, and the devised AuRSO-RAGNet is 96.635% by using a K group of 6. Relative to HSM-QA, the AuRSO-RAGNet attained a better F1-score by 5.64%.



(a)



(b)

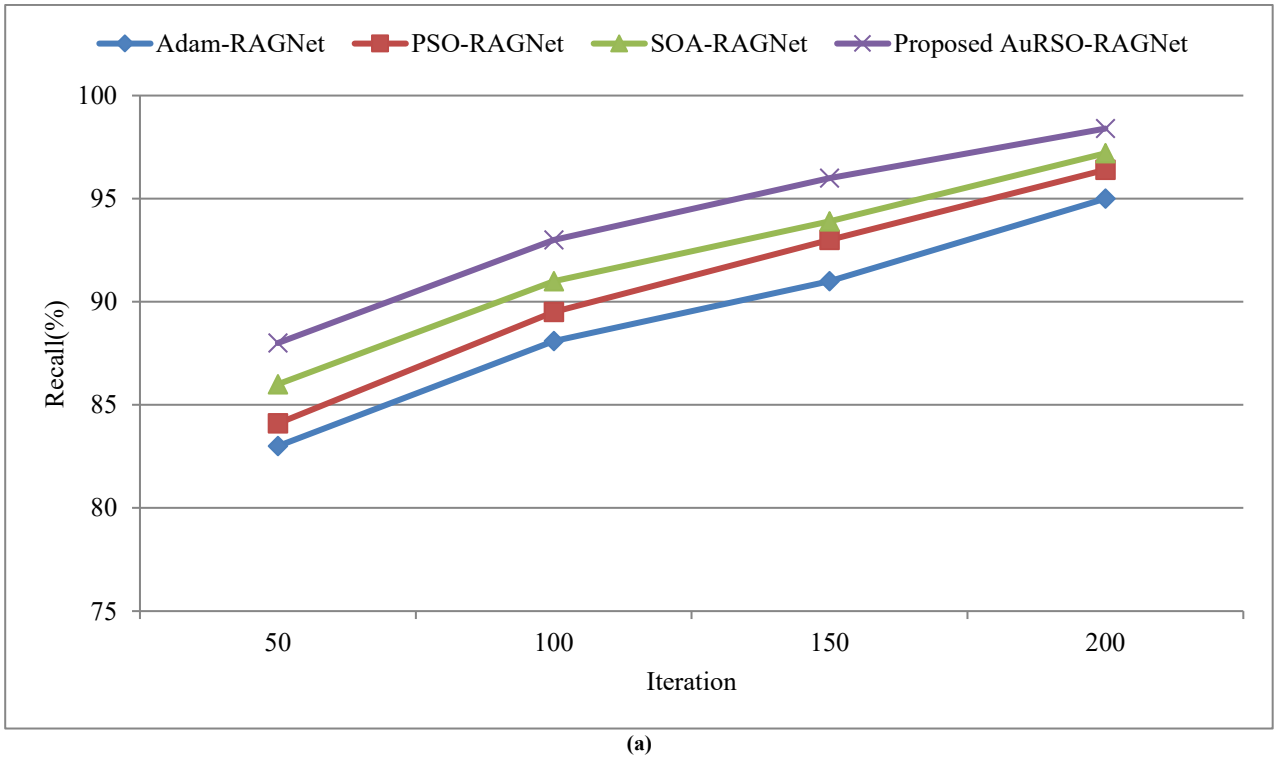


(c)

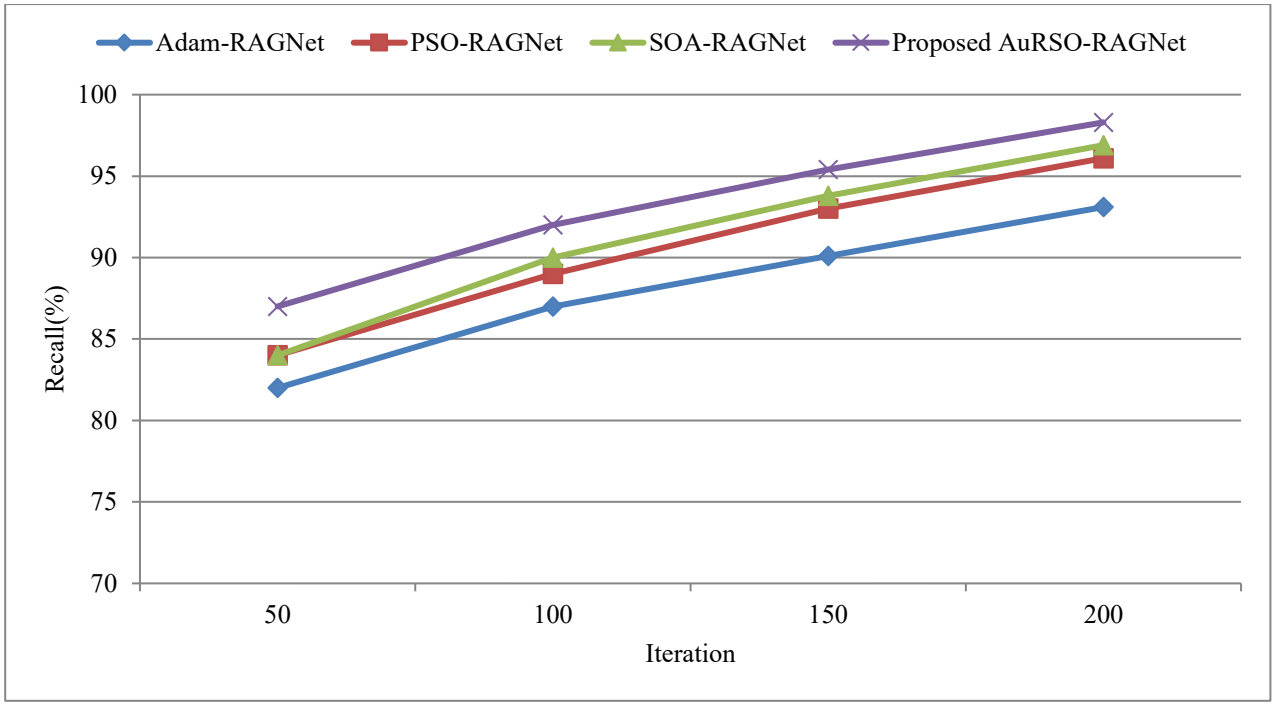
Fig. 4 K group assessment of AuRSO-RAGNet for generating question answering utilizing database 2: a) precision, b) recall, and c) F1-score

4.7. Algorithmic Analysis

Figure 5 shows the algorithmic analysis. Here, the performance of the proposed AuRSO-RAGNet is against Adam-RAGNet, Particle Swarm Optimization (PSO)-RAGNet, and SOA-RAGNet. The algorithmic analysis using dataset-1 is provided in Figure 5a) and dataset-2 is given in Figure 5b). When the iteration is 200, the recall obtained by the Adam-RAGNet, PSO-RAGNet, SOA-RAGNet, and AuRSO-RAGNet is 95.00%, 96.33%, 97.22%, and 98.35% using dataset-1 and 93.10%, 96.00%, 96.90%, and 98.25% using dataset-2. This analysis verifies that the performance of AuRSO exceeds that of other optimization algorithms.



(a)

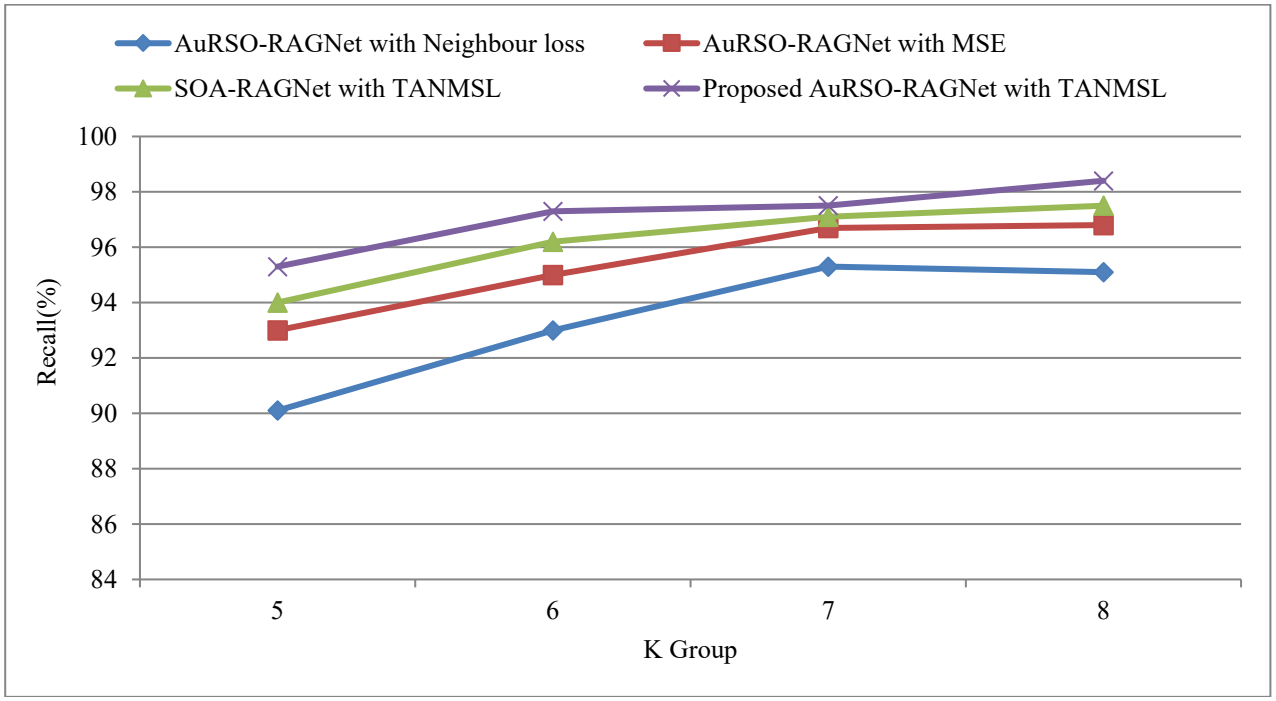


(b)

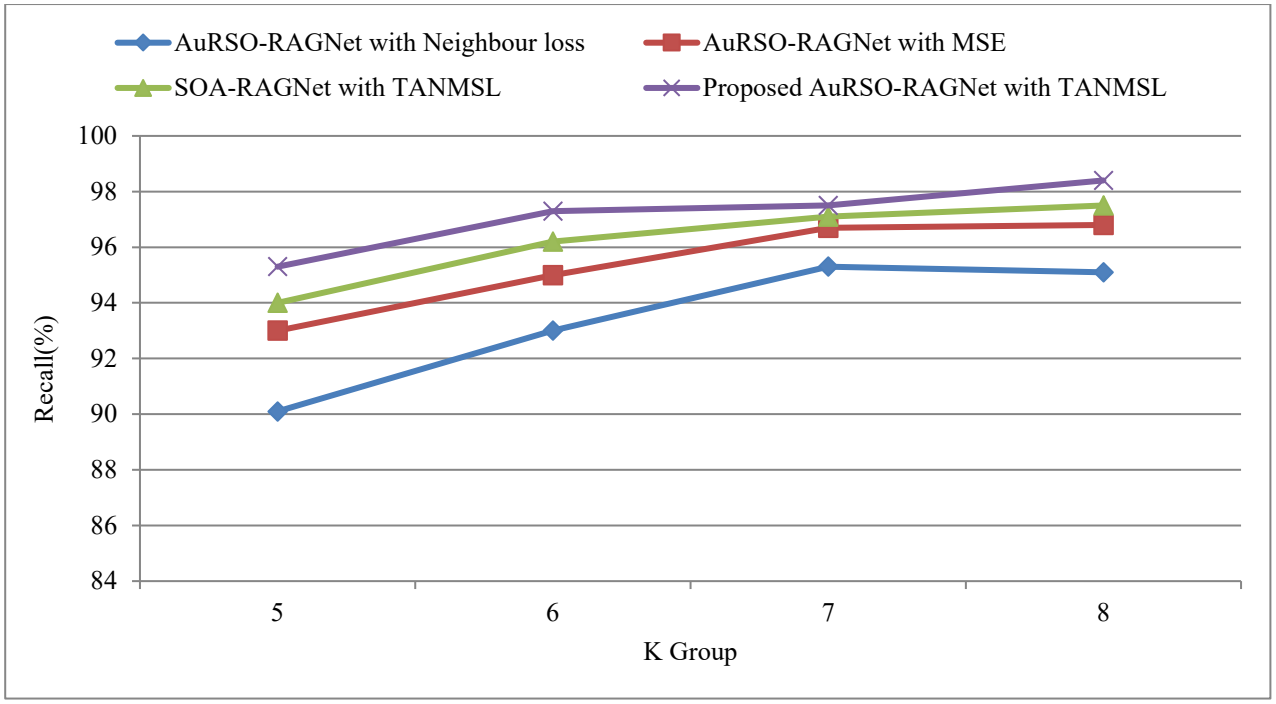
Fig. 5 Algorithmic analysis: a) Dataset 1, b) Dataset 2

4.8. Ablation Study

The ablation study is provided in Figure 6. The ablation analysis results of dataset-1 and dataset-2 are illustrated in Figure 6a and Figure 6b, respectively. When the k-value is 8, the recall obtained by the AuRSO-RAGNet with Neighbour loss, AuRSO-RAGNet with MSE, SOA-RAGNet with TANMSL, and the proposed AuRSO-RAGNet with TANMSL is 95.12%, 96.79%, 97.45%, and 98.35% using dataset-1 and 96.27%, 97.35%, 97.89%, and 98.25% using dataset-2. The performance of the AuRSO-RAGNet with TANMSL is higher due to the contribution of all components.



(a)

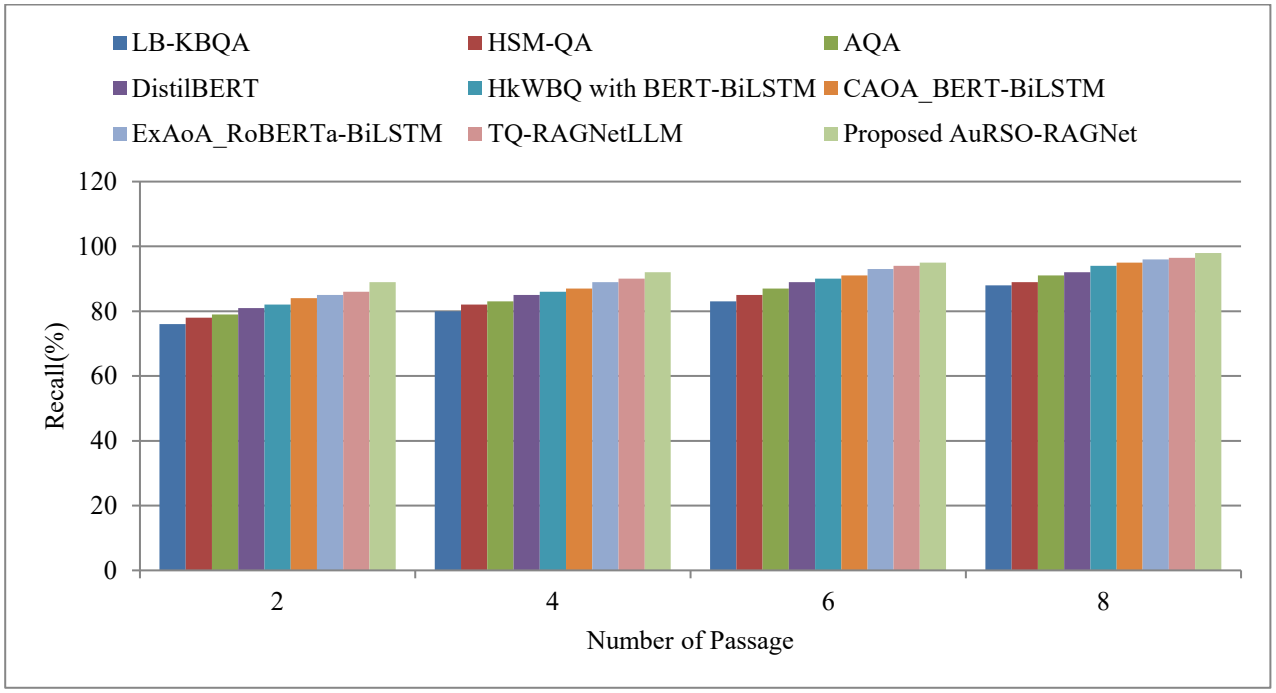


(b)

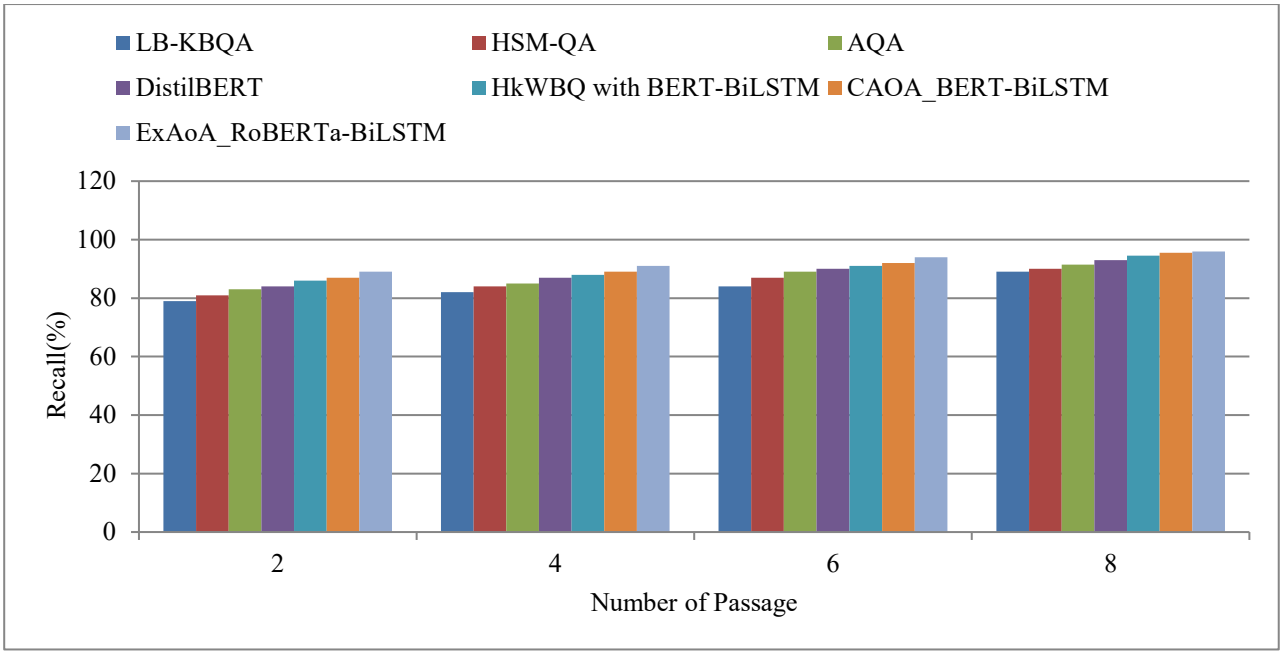
Fig. 6 Ablation study: a) Dataset 1, b) Dataset 2

4.9. Scalability Analysis

The scalability analysis is shown in Figure 7, where the analysis using dataset-1 is given in Figure 7a) and dataset-2 is depicted in Figure 7b). When the number of passages=8, the recall recorded by the LB-KBQA, HSM-QA, AQA, DistilBERT, HkWBO with BERT-BiLSTM, CAOA_BERT-BiLSTM, ExAoA_RoBERTa-BiLSTM, TQ-RAGNetLLM, and AuRSO-RAGNet is 87.89%, 88.99%, 90.99%, 91.99%, 94.00%, 95.00%, 95.89%, 96.36%, and 97.80% using dataset-1 and 89.67%, 90.67%, 91.78%, 93.00%, 94.78%, 96.00%, 96.57%, 97.68%, and 98.32% using dataset-2. It is clear that the recall of the AuRSO-RAGNet is higher than that of the existing methods.



(a)



(b)

Fig. 7 Scalability analysis: a) Dataset 1, b) Dataset 2

4.10. Analysis based on Computational Complexity

Table 2 compares computational complexity in terms of computational time and memory usage of different QA models based on dataset-1 and dataset-2. Traditional methods show higher time and memory consumption. The proposed AuRSO-RAGNet attains the lowest computational time of 23.99 sec and 24.99 sec and memory usage of 15.4 MB and 16.4 MB for datasets 1 and 2, due to the AuRSO optimization and dual-view architecture.

Table 2. Computational time and memory analysis

Methods	Computational time (Sec)		Memory (MB)	
	Dataset-1	Dataset-2	Dataset-1	Dataset-1
LB-KBQA	54.99	56.99	26.8	27.4
HSM-QA	49.91	51.01	24.6	25.3
AQA	45.99	47.00	22.5	24.4
DistilBERT	39.10	41.99	20.8	23.8
HkWBO with BERT-BiLSTM	37.00	38.99	20.5	23.5
CAOA_BERT-BiLSTM	32.09	33.99	19.5	21.5
ExAoA_RoBERTa-BiLSTM	30.00	30.99	17.9	19.7
TQ-RAGNetLLM	26.10	28.99	17.4	18.7
Proposed AuRSO-RAGNet	23.99	24.99	15.4	16.4

4.11. Comparative Discussion

Table 3 displays the comparative discussion of the proposed AuRSO-RAGNet for question answering, based on evaluation parameters, like recall, F1-score, and precision, using a K-group of 8. By using dataset 2, the precision gained by the AuRSO-RAGNet is 97.840%, the recall is 89.594%, and the F1-score is 98.043%. The proposed AuRSO-RAGNet achieved superior performance compared with existing methods due to the effective integration of dual-view retrieval learning and optimization strategies. The Q-DVM and Q-RAG layers improved contextual understanding by strengthening the interaction between query representation and retrieval generation processes. In addition, the TANMSL-based learning mechanism enhanced semantic learning capability and reduced retrieval inconsistencies during answer generation. Furthermore, the AuRSO optimization strategy improved parameter tuning, convergence efficiency, and overall model stability, which contributed to higher precision, recall, and F1-score values. As a result, the proposed framework demonstrated better generalization capability and answer prediction performance than conventional state-of-the-art approaches.

Table 3. Comparative Discussion

Metrics	LB-KBQA	HSM-QA	AQA	DistilBERT	HkWBO with BERT-BiLSTM	CAOA_BERT-BiLSTM	ExAoA_RoBERTa-BiLSTM	TQ-RAGNetLLM	Proposed AuRSO-RAGNet
Dataset 1									
Precision (%)	84.383	87.383	90.273	92.275	94.273	95.539	96.016	97.119	97.729
Recall (%)	85.717	89.718	91.715	92.615	95.707	96.516	97.018	97.508	98.348
F1-score (%)	85.045	88.535	90.988	92.445	94.984	96.025	96.514	97.313	98.037
Dataset 2									
Precision (%)	85.595	87.704	90.707	91.704	93.595	94.483	95.916	96.619	**97.840**
Recall (%)	87.694	89.594	90.704	93.004	95.372	96.405	97.027	97.259	**98.248**
F1-score (%)	86.631	88.639	90.705	92.349	94.475	95.434	96.468	96.938	**98.043**

4.12. Statistical Analysis

Table 4 presents the T-test results comparing the proposed AuRSO-RAGNet with existing QA models across precision, recall, and F1-score for both datasets based on k-Fold cross-validation results. The obtained T-statistic values are steadily high, while the equivalent p-values are all under 0.05, indicating that the improvements achieved by the proposed model are statistically significant. For Dataset-1, AuRSO-RAGNet shows stronger statistical significance when compared to traditional methods. A similar trend is observed for Dataset-2, where higher T-statistics and lower p-values further confirm the robustness of the AuRSO-RAGNet. This statistical analysis validates that the performance improvements of AuRSO-RAGNet are not due to random variation but are significant and reliable across both datasets.

Table 4. T-Test

Proposed method	Traditional methods	T-statics			P-value		
		Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
		Dataset-1					
AuRSO-RAGNet	LB-KBQA	2.879	3.489	3.190	0.020	0.016	0.018
	HSM-QA	2.689	3.229	2.888	0.024	0.019	0.022
	AQA	2.589	2.988	2.590	0.027	0.025	0.027
	DistilBERT	2.368	2.777	2.379	0.032	0.029	0.030
	HkWBO with BERT-BiLSTM	2.099	2.589	2.099	0.036	0.033	0.036
	CAOA_BERT-BiLSTM	1.988	2.481	1.888	0.039	0.037	0.040
	ExAoA_RoBERTa-BiLSTM	1.790	2.390	1.779	0.042	0.040	0.043
	TQ-RAGNetLLM	1.589	2.098	1.589	0.046	0.042	0.045
		Dataset-2					
AuRSO-RAGNet	LB-KBQA	3.125	3.735	3.436	0.018	0.014	0.016
	HSM-QA	2.935	3.475	3.134	0.022	0.017	0.020
	AQA	2.835	3.234	2.836	0.025	0.023	0.024
	DistilBERT	2.614	3.023	2.625	0.030	0.027	0.028
	HkWBO with BERT-BiLSTM	2.345	2.835	2.345	0.034	0.031	0.034
	CAOA_BERT-BiLSTM	2.234	2.727	2.134	0.037	0.035	0.038
	ExAoA_RoBERTa-BiLSTM	2.036	2.636	2.025	0.040	0.038	0.041
	TQ-RAGNetLLM	1.835	2.344	1.835	0.044	0.040	0.042

5. Conclusion

QAS has emerged as a widely adopted and often efficient technique for information retrieval. These systems allow users to pose queries in natural language and attain relevant responses to their inquiries. However, the existing systems suffer from poor generalization and low reliability. To address these limitations, a new approach, namely AuRSO-RAGNet, is proposed for question answering. Firstly, the proposed AuRSO-RAGNet is based on TQ-RAGNetLLM. Moreover, TQ-RAGNet is developed by adjusting PQA-RAG's learning rule by using TANMSL. Here, answer generation is performed by processing the input questions and passage utilizing PQA-RAG, which contains two main components: the Q-DVM layer as well as the Q-RAG Layer. The overall LLM is optimized using AuRSO, which is developed by the combination of SOA with CAViaR to enhance performance. In addition, experimental outcomes demonstrate that the devised AuRSO-RAGNet attained a better precision of 97.840%, recall of 98.248%, and an F1-score of 98.043%, outperforming conventional methods. However, the proposed framework is evaluated using limited benchmark datasets, which may affect its adaptability to highly domain-specific and multilingual question answering tasks. In the future, multimodal information such as video, audio, images, and text will be integrated to handle complex cross-media queries more effectively. Also, the model will be tested in practical deployment environments.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Funding Statement

No external funding was associated with the research study.

Authors' Contributions

Amit Virmani conceived and designed the research study, collected and analyzed the data, and prepared the overall manuscript. Alok Kumar provided overall supervision, guidance, and critical review of the manuscript. Vineeta Singh assisted in reviewing and improving the manuscript. All authors read and approved the final version of the paper.

Data Availability Statement

The data that support the findings of this study are openly available in the XQuAD Dataset at https://github.com/google-deepmind/xquad and MultiLingual Question Answering (MLQA) dataset at https://github.com/google-deepmind/xquad.

Code Availability Statement

The source code of this research is available at: https://github.com/amitvirmanicsjmu/Research_Work

References

[1] Priti Gumaste et al., "Automated Question Generator System using NLP Libraries," *International Research Journal of Engineering and Technology*, vol. 7, no. 6, pp. 4568-4572, 2020. [Google Scholar] [Publisher Link]

[2] Eman Mohamed Nabil Alkholy, Mohamed Hassan Haggag, and Amal Aboutabl, "Question Answering Systems: Analysis and Survey," *International Journal of Computer Science & Engineering Survey*, vol. 9, no. 6, pp. 1-13, 2018. [Google Scholar] [Publisher Link]

[3] Rohit Arora et al., "Comparative Question Answering System Based on Natural Language Processing and Machine Learning," *International Conference on Artificial Intelligence and Smart Systems*, pp. 373-378, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Kushwanth Sai Lalam et al., "Question Answering System Using NLP," *International Research Journal of Engineering and Technology*, pp. 1658-1661, 2021. [Publisher Link]

[5] Y. Sharma and S. Gupta, "Deep Learning Approaches for Question Answering System," *Procedia Computer Science*, vol. 132, pp. 785-794, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Vaishali Fulma, K.P. Moholkar, and S.H. Patil, "The Implementation of Question Answer System using Deep Learning," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 15, pp. 176-182, 2021. [Google Scholar] [Publisher Link]

[7] Arefeh Kazemi et al., "FarsNewsQA: A Deep Learning-Based Question Answering System For The Persian News Articles," *Information Retrieval Journal*, vol. 26, no. 1, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yan Zhao et al., "LB-KBQA: Large-Language-Model and BERT based Knowledge-Based Question and Answering System," *IEEE 22nd International Conference on Industrial Informatics*, pp. 1-5, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Juan Sequeda, Dean Allemang, and Bryon Jacob, "Knowledge Graphs as a Source of Trust for LLM-Powered Enterprise Question Answering," *Journal of Web Semantics*, vol. 85, pp. 1-6, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jinlu Zhang et al., "HSM-QA: Question Answering System Based on Hierarchical Semantic Matching," *IEEE Access*, vol. 11, pp. 77826-77839, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Hainan Wang, "Automatic Question-Answering Modeling in English by Integrating TF-IDF and Segmentation Algorithms," *Systems and Soft Computing*, vol. 6, pp. 1-10, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Jafar A. Alzubi et al., "COBERT: COVID-19 Question Answering System using BERT," *Arabian Journal for Science and Engineering*, vol. 48, no. 8, pp. 11003-11013, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] B.P. Swathi, M. Geetha, and M.V. Suhas, "Enhancing Query Understanding in Educational Question Answering Systems Through Neural Models," *IEEE Access*, vol. 14, pp. 19920-19933, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[14] Qidong Liu et al., "LLM-ESR: Large Language Models Enhancement for Long-Tailed Sequential Recommendation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 26701-26727, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Mohammed Q. Ibrahim et al., "Secant Optimization Algorithm for Efficient Global Optimization," *Scientific Reports*, pp. 1-50, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[16] Robert F. Engle, and Simone Manganelli, "CAViaR: Conditional Autoregressive Value at Risk by Regression Quantiles," *Journal of Business & Economic Statistics*, vol. 22, no. 4, pp. 367-381, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[17] Wei Yuan, and Wenbo Xu, "Neighborloss: A Loss Function Considering Spatial Correlation for Semantic Segmentation of Remote Sensing Image," *IEEE Access*, vol. 9, pp. 75641-75649, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ming-Chun Lee, and Wei-Ho Chung, "Mean Square Error (MSE) Based Hybrid Analog and Digital Combining for Systems with Large Receive Antenna Arrays," *IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications*, pp. 1-6, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[19] R. Barrio et al., "Breaking the Limits: The Taylor Series Method," *Applied Mathematics and Computation*, vol. 217, no. 20, pp. 7940-7954, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[20] XQuAD Dataset 2026. [Online]. Available: https://github.com/google-deepmind/xquad

[21] MultiLingual Question Answering (MLQA) Dataset, 2026. [Online]. Available: https://github.com/google-deepmind/xquad

[22] Mengqi Li, and Rufu Qin, "DualGraphRAG: A Dual-View Graph-Enhanced Retrieval-Augmented Generation Framework for Reliable and Efficient Question Answering," *Applied Sciences*, vol. 16, no. 5, pp. 1-2026. [CrossRef] [Google Scholar] [Publisher Link]