Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Hybrid LLM and RAG Framework for Generating Questions Aligned with Bloom's Taxonomy

Senthilkumar P

way2sen@gmail.com

Central University of Tamil Nadu

Nandhini K

Central University of Tamil Nadu

Research Article

Keywords: Question Generation, Bloom Taxonomy, NLP, Large Language Model, Retrieval Augmented generation

Posted Date: May 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9162087/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Hybrid LLM and RAG Framework for Generating Questions Aligned with Bloom's Taxonomy

Senthilkumar P[1]* and Nandhini K[2]†

[1, 2] Department of Computer Science,

Central University of Tamil Nadu,

Thiruvarur, India.

*Corresponding author(s). E-mail(s): way2sen@gmail.com;

Contributing authors: nandhinikumaresh@cutn.ac.in;

†These authors contributed equally to this work.

Abstract

In today's educational paradigm, automated question generation plays a pivotal role as it helps in scalable material production, continuous testing, and personalized learning. However, the task of generating academically relevant questions, which are also categorized according to the different levels of cognitive Bloom's taxonomy, remains a challenge. Although state-of-the-art LLM-based approaches have improved the quality of questions for examination purposes, they are also marred by problems such as hallucinations, lack of subject matter knowledge, and lack of syllabus relevance. In contrast, conventional rule-based and template-based approaches are limited to lower-order cognitive tasks. To address these issues, this paper proposes a Delimiting LLM framework with the concept of Retrieval-Augmented generation (RAG) for syllabus-relevant question generation, which is also in accordance with the different levels of cognitive Bloom's Taxonomy. The proposed approach can generate a variety of novel question formats by utilizing RAG-based examples as few-shot learning, semantic reference context, and exact textbook content as contextual grounding. The proposed approach was validated through automated and human evaluations to measure the quality of questions, cognitive relevance, and pedagogical significance. Based on the experimental findings, the proposed framework outperformed conventional and non-retrieval-based baselines in generating cognitively diverse, syllabus-relevant, and pedagogically significant questions. Additionally, it also offers a scalable solution for smart assessment question generation.

Keywords: Question Generation, Bloom Taxonomy, NLP, Large Language Model, Retrieval Augmented generation

1 INTRODUCTION

Exams are the primary method to assess the academic value of students, which are an essential instructional technique for improving critical thinking, active learning, and comprehension in educational settings [1], [2], [3]. Exams are one of the most popular ways to evaluate students' knowledge, and they call for well-written question papers. However, it takes a lot of effort and repetition for teachers to create these questions by hand. Teachers often spend a lot of time creating question papers because there are so many exams in the academic field, and every exam requires a different set of questions.

Table 1: Question types in higher education: Mapping with Bloom's Taxonomy.

S.No	Levels	Question Type	Bloom Taxonomy	Focus on
1	L1	True or false	Remember	Recall facts
2	L2	Fill in the blanks	Understand	Explain concepts
3	L3	Fill in the blanks with a hint	Apply	Use knowledge
4	L4	Multiple choice questions	Analyze	Examine structures
5	L5	Multi select questions	Evaluate	Judge effectiveness
6	L6	Descriptive	Create	Generate new ideas

Automatic Question Generation (AQG) plays a vital role in teaching and learning through technology to develop individualized [2], context-specific questions that promote student engagement and understanding [4], [5], [6]. Automating this procedure using a machine-based approach could assist save significant time, which can then be focused towards more beneficial instructional initiatives. However, machine-generated questions cannot be used blindly. Automation can speed up the process, but it does not ensure quality or exam-specific appropriateness, and many may be unclear, inaccurate, or irrelevant. Advanced techniques like large language models (LLMs) can assist in question generation[7], [8], but they are often fail to produce questions that correspond with human framing styles—especially when not fine-tuned for a given educational subject and syllabus.

A well-known system for categorizing concerns based on their level of difficulty and cognitive complexity is Bloom's Taxonomy [9], [10]. Certain question formats can be matched to particular Bloom's Taxonomy levels based on earlier research on question kinds and their corresponding difficulty levels. Table 1 illustrates this mapping, which is developed from an exhaustive study of current material as well as survey responses collected from college-level students.

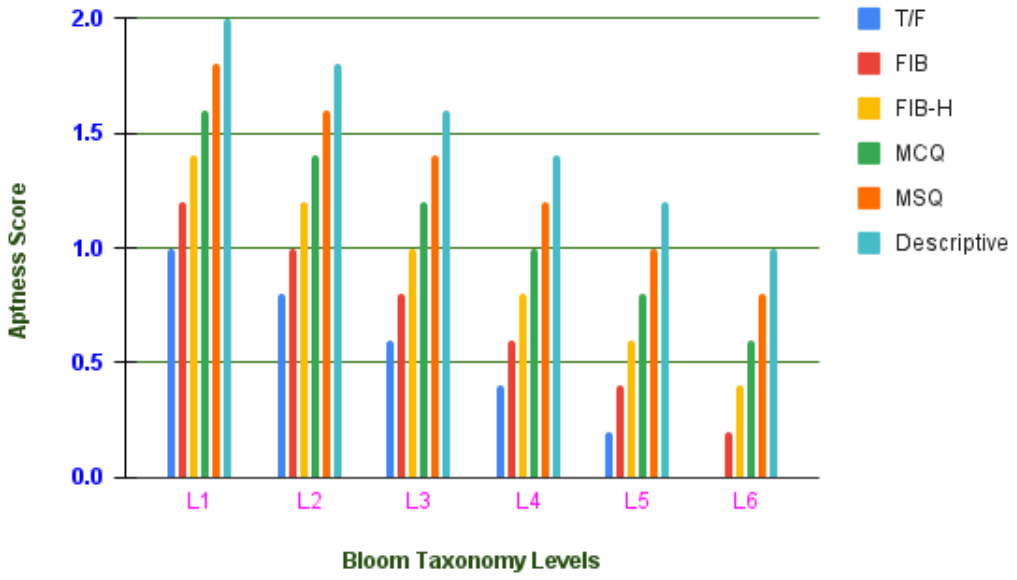


Fig. 2. Literature Search: a pictorial representation of the procedure

The aptness scores of six distinct question types—True/False (T/F) [3], Fill-in-the-Blank (FIB) [11], Higher-level Fill-in-the-Blank (FIB-H), Multiple Choice Questions (MCQ) [12], Multiple-Select Questions (MSQ), and Descriptive questions—are mapped across the six Bloom's Taxonomy cognitive levels (L1 to L6) in the diagram. The aptness score reflects how suited or aligned each question type is for assessing the corresponding cognitive level.

Nevertheless it's not the best idea to strictly link a specific question type to a set Bloom's Taxonomy level. Large language model (LLM) technologies have made it possible to design a single question type spanning several Blooms' Taxonomy cognitive tiers. Therefore, the main goal of this study is to show how even basic questions may be constructed inside an exam-specific framework to target various Bloom's Taxonomy levels.

Using LLM to generate questions is similar to a collaborative method between humans and computers, which is crucial—using the human for exam-specific guidance and the machine for speed. In the proposed method, the machine system is replaced with a Large Language Model (LLM), while the human role is simulated using the Retrieval-Augmented Generation (RAG) approach. This combination permits more effective prompt design, generating automatically generated questions that closely mirror those crafted by people using an exam-specific manner.

Table 2: Question Aptness Scoring [QAS]: Algorithm.

QCS	Bloom Taxonomy						
Question Types		L1	L2	L3	L4	L5	L6
	L1	1.0	1.2	1.4	1.6	1.8	2.0
	L2	0.8	1.0	1.2	1.4	1.6	1.8
	L3	0.6	0.8	1.0	1.2	1.4	1.6
	L4	0.4	0.6	0.8	1.0	1.2	1.4
	L5	0.2	0.4	0.6	0.8	1.0	1.2
	L6	0.0	0.2	0.4	0.6	0.8	1.0

This study primarily analyzes the challenges associated with creating a variety of exam-specific questions linked with different levels of Bloom's Taxonomy, with a special emphasis on questions targeting higher-order cognitive skills. In particular, we address the following three research studies:

1. How can a Large Language Model (LLM) be effectively limited to generate questions across different levels of Bloom's Taxonomy for practical use inside a certain examination framework?
2. To what extent does the integration of a Retrieval-Augmented Generation (RAG) technique, paired with few-shot learning instances, effect the performance of an LLM during prompt-based question generation?
3. To what extent does a reference-based learning approach enhance the exam alignment, structural coherence, and quality of LLM-generated questions?

The LLM generates questions aligned with Bloom's taxonomy levels based on the input context. The challenge includes instructions such as: 'You are a college teacher writing examination questions—generate academically relevant questions from the given context.' We are able to assess the model's highest standalone capacity because these instructions direct the LLM in question generation without depending on any outside sources. Additionally, exam-specific sample questions, extracted via RAG (dynamic few-shot learning from a curated dataset), are supplied alongside user inputs. This helps the LLM produce questions that follow the style and structure of exam-specific RAG examples [13], [14]. Once the LLM creates the questions, each one is evaluated

to determine its cognitive level within Bloom's taxonomy, helping to assess the appropriateness of the question type for each level.

In the proposed method, our contributions are,

RAG File Construction: The T5 model is used to generate an exam-specific RAG file that includes questions from prior years' exams, contextual information, question types, and matching Bloom's Taxonomy levels. In the RAG framework, this file serves as the foundation for creating dynamic few-shot learning instances. Context-Aware Retrieval Using ANN: The RAG pipeline uses an Artificial Neural Network (ANN)-based retrieval mechanism to find and extract contextually aligned content in order to improve the relevance of recovered information. Reference-Based Learning Strategy: To enhance semantic grounding and the general caliber of LLM-generated questions, other reference texts, such as preceding and succeeding contextual sections are integrated into rapid formulation in addition to the primary context.

The rest of the paper is organized as follows: Part 2 discusses the relevant literature, and Part 3 of our discussion covers our suggested approach for exam-specific question generation. Part 4 presents the outcome of our suggested approach and a discussion of its benefits and drawbacks. In Part 5, we finally bring our paper to a close by discussing our future plans to expand on it.

2 RELATED WORKS

Automatic question generation (AQG) began with rule-based approaches [15], [16], particularly geared at speeding up question paper creation in the education sector. These approaches depended largely on predetermined rules, allowing minimal flexibility in format modifications and lacked exam-specific adaptation. With the advent of NLP, AQG added capabilities such as generating questions based on key phrases and part-of-speech (POS) patterns from the provided context[5], [17]. Nevertheless, this method frequently generated semantically redundant questions, had limited practical usefulness, and required manual feature selection. By utilizing massive datasets, deep learning models—in particular, T5-based architectures—addressed some of these issues [18]. However, the dataset had significant effects on the generated questions' quality, and biased or irrelevant data frequently produced useless results.

Large Language Models (LLMs) represents a major advancement, using decoder-based architectures trained on enormous corpora to generate high-quality queries [8], [19], [20]. LLM-based AQG eliminates the requirement for manual feature selection or dedicated datasets and may create questions of any type at multiple cognitive levels. However, these algorithms are unable to consistently generate fully aligned questions because there are no datasets particularly developed for generating questions across various Bloom's taxonomy levels in academic settings [21], [22]. Retrieval-Augmented Generation (RAG), which improves LLM performance by providing pertinent context and sample formats, can get around this limitation [23]. RAG uses dynamic few-shot learning [24] to give pattern-like datasets that help the model learn on its own and create exam-specific, cognitively aligned questions [25].

The RAG database is populated exclusively with computer science–related text samples to ensure the generation of high-quality example questions. By using an exam-specific RAG file, the system provides only relevant sample questions as the reference set. These, combined with user input, are used to produce more accurate and context-appropriate results.

3 PROPOSED METHOD

The proposed method includes three core components:

1. LLM Pre-Processing: Creating a RAG-based vector database.
2. LLM Processing: Utilizing exam-specific few-shot learning through RAG.
3. LLM Post-Processing: Validating and refining the generated questions for quality and alignment with Bloom's Taxonomy.

In the proposed approach, the user provides three inputs: the context (used as the basis for question generation), the question type (defining the format of the question), and the cognitive level based on Bloom's Taxonomy (indicated as A, B, and C in the diagram) and the workflow of the proposed method is depicted in Figure 1. These inputs are used to build a prompt, coupled with additional reference context and some examples (D) collected via the RAG-based technique. The RAG system offers dynamic few-shot learning by exploiting an exam-specific dataset optimized for generating questions aligned with current trends.

An exam-specific dataset is built using syllabus-aligned textbooks, documents, and notes, which are scraped and pre-processed into individual sentences based on related keywords. These sentence–keyword pairs are subsequently fed into a T5 model[26], fine-tuned on the SQuAD dataset, to generate contextually relevant questions.

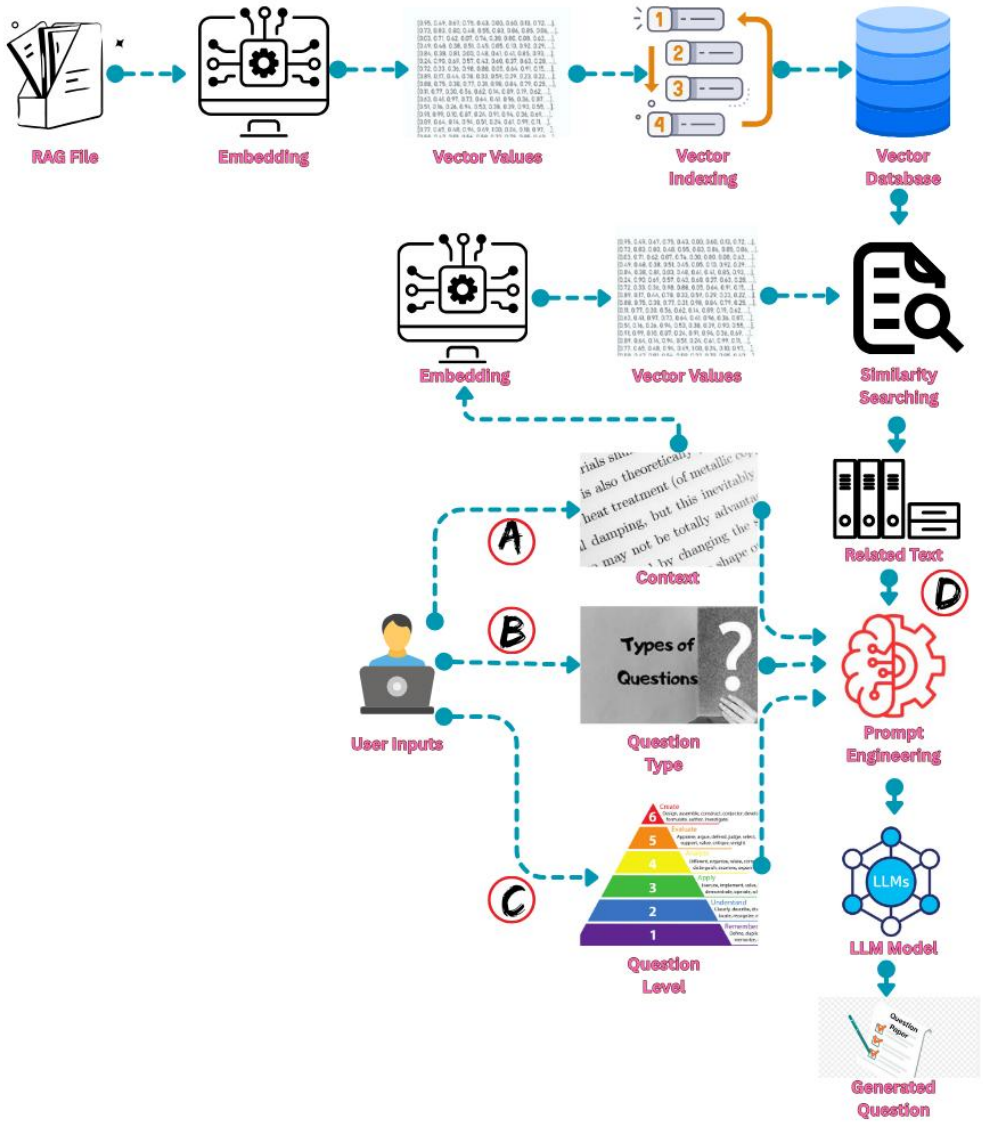


Fig. 1.Question Generation: Proposed method flow graph representation

As seen in Table 4, the RAG-based input process starts with a file containing four columns: context, question type, Bloom's level, and the related generated questions. The BGE embedding technique, which is recommended for its strong semantic similarity capabilities, effective retrieval performance, high performance in the MTED (Massive Text Embedding Benchmark), and capacity to lessen hallucinations in large language model (LLM) outputs, is used to embed the context column.

Once the embedding is accomplished, the other columns—question type, Bloom's taxonomy level, keywords, and generated question—are paired with their relevant context embedding to form chunk values. After that, vector representations are created from these chunk values. Lastly, the Pinecone approach is used to index and store the vectors in a vector database.
With Pinecone, the most relevant chunks are retrieved quickly, thanks to its support for ANN-based search and hybrid search methods.

Using Approximate Nearest Neighbour (ANN) search alongside with cosine similarity, the user-provided context is first converted into an embedding, which is subsequently compared with the pre-stored embedding in the vector database of the RAG file. This similarity-based retrieval delivers the most relevant text chunks from the Pinecone vector database. To enhance the precision of retrieval, further filtering can be done using metadata values—such as selecting Bloom Level = Apply and Question Type = MCQ—in conjunction with the user's inputs. This combined method of semantic similarity search and metadata-based filtering ensures more accurate and contextually relevant results from the vector database.

The LLM is prompted to create new, contextually relevant questions using the obtained data (row-based). The LLM models, which are well-suited for RAG-based systems because of their powerful few-shot learning capabilities, receive these chunks as input and use them as dynamic few-shot learning examples. The RAG-retrieved sample content and user inputs are used by the LLM to produce questions. After the question is created, it is filtered and assessed according to its relevance score.

Prompting Working Principle:

$$f(P,T,L)=Q \quad (1)$$

This overall problem statement is given in the Equation 1 as,

P: Primary input context

T: Type of question

L: Bloom Taxonomy level

Q: the output (the generated question according to P,T and L)

In LLM:

The LLM generates question purely on the given text (P), given question type (T) and given Taxonomy level(L) is given in the Equation 2, it possible generated question ranges from 1 to n according to the context availability in the input.

- Generate meaningful questions according to the question type and Bloom taxonomy question level from the following context: PRIMARY CONTEXT.

Context: {context}

- Instructions:
  - All questions must be answerable strictly from the PRIMARY CONTEXT
  - Avoid copying sentences from either input

$$\boldsymbol{f}_{\{\boldsymbol{LLM}\}:} \rightarrow \{(\boldsymbol{P_i}, \boldsymbol{T_i}, \boldsymbol{L_i} \rightarrow \boldsymbol{Q_i})\}_{\{\boldsymbol{i=1}\}}^{\{\boldsymbol{n}\}} \quad \textbf{(2)}$$

In Reference Context Learning:

The Reference Context Learning approach gives a sample context relevant to the real context for the question generation) is given in the Equation 3, it increases the quality of the question as well as the chances of framing the more number of questions. Here the P means the original context and Pi means to have a sample contexts.

- Generate meaningful questions according to the question type and Bloom taxonomy question level from the following context: PRIMARY CONTEXT.
- Use the REFERENCE TEXT only as guidance for question style, depth, and structure.

Context: {context}

Reference :{ reference context}

- Instructions:
  - All questions must be answerable strictly from the PRIMARY CONTEXT
  - Do NOT introduce concepts that appear only in the reference text
  - Avoid copying sentences from either input

$$\boldsymbol{f}_{\{\boldsymbol{RCL}\}}: \rightarrow \boldsymbol{R_i}, \{(\boldsymbol{f}_{\{\boldsymbol{RCL}\}})_{\{\boldsymbol{i=1}\}}^{\{\boldsymbol{k}\}}\} \quad \textbf{(3)}$$

From a student examination perspective, the primary requirement is to study subjects using a syllabus-oriented approach. Students are not expected to develop extensive knowledge beyond the prescribed syllabus; instead, their focus is on performing well in the examination. This approach aligns with the *remember* level of Bloom's taxonomy. Although this may appear limited, at the foundational stage of education it helps students build strong basic knowledge.

To ensure that the LLM concentrates strictly on syllabus-specific content, it is constrained to operate only within predefined boundaries. Accordingly, textbook-based context—closely aligned with the syllabus—is provided while generating questions. This reference content is incorporated using a few-shot learning approach to guide syllabus-oriented question is given in the Equation 4,

In RAG:

- Generate {number of questions} meaningful questions from the following context:
- Generate questions from the PRIMARY CONTEXT.
- Use the REFERENCE TEXT only as guidance for question creation.
- Use the SAMPLE QUESTION only as guidance for question style, depth, and structure.

Context: {context}

Reference :{ reference context}

Sample Question :{ questions}

- Instructions:
  - All questions must be answerable strictly from the PRIMARY CONTEXT
  - Do NOT introduce concepts that appear only in the reference text
  - All questions must be framed strictly from any of the SAMPLE QUESTION format
  - Avoid copying sentences from either input

$$f_{\{RAG\}} : \rightarrow \left(S_i, \{(f_{\{RCL\}})_{\{i=1\}}^{\{k\}}\}\right) \quad \textbf{(4)}$$

The RAG gives a sample reference questions, that boosts up the LLM for framing the relevant question as well it follows the question format. The ranges of sample question decided by the situation and needs.

The second aspect is that question papers should be aligned with patterns observed in previous years to avoid ambiguity for students. While the questions need not be identical, those with similar structural formats should be treated equivalently. To achieve this, the proposed method employs a Retrieval-Augmented Generation (RAG) framework, where previous years' questions serve as reference material for generating new questions across different Bloom's taxonomy levels and multiple question types.

In Combined Form:

$$Q = f_{\{PRS\}(P_i,(R_i(S_i)))_{\{i=1\}}^{\{k\},}} \quad \textbf{(5)}$$

The final hybrid approach is given in the Equation 5, LLM generates question with the help of RCL and RAG approach, it produces very good questions as expected and has good human evaluation score too.

In general, the results clearly show that there is a synergistic effect between RAG and RCL in improving the performance of the LLM, and that the combination of the two yields the best results. The model that performs best in all experimental conditions is Mistral, which adapts and improves exceptionally well when prompted with both RAG and RCL. These results clearly indicate that hybrid approaches to prompting are necessary for maximizing the interpretability, correctness, and usefulness of the generated responses, especially in complex tasks that require a lot of information.

When the LLM is used alone, the question that is generated is very similar to the source sentence, which results in a direct recall-based item with very little linguistic variation. The use of example formats to guide the LLM is improved by Reference-based Learning Constraints (RLC), which improve the quality of the questions by allowing for more varied and pedagogically formatted questions without deviating from the desired cognitive level.

Retrieval-Augmented Generation (RAG) is another improvement that provides more contextual grounding, which allows the model to generate questions that are both semantically accurate and faithful to the original content.

Overall, the combined LLM + RLC + RAG framework achieves superior balance among accuracy, diversity, and educational relevance**,** making it well-suited for automated question generation in academic assessment systems[27]. This layered approach ensures Bloom-level alignment while reducing redundancy and improving the naturalness of generated questions.

I. Source Context: 'Definition of a byte in computer science'
II. 'Question Type': 'L2' and 'Bloom-Taxonomy Level': 'L1'
III. Generated Question:
   a. LLM: - A byte consists of _____ bits.
   b. LLM + RLC: - The basic unit of digital information in computer science is called a _____.
   c. LLM + RLC + RAG: - A byte is made up of _____ individual bits.

Table 4: An approaches comparison.

Approach	Strength	Limitation
LLM	Simple, direct recall	Tends to copy context
LLM + RLC	Better variation and format control	Limited external grounding
LLM + RLC + RAG	Most diverse, accurate, exam-oriented	Slightly higher complexity

As presented in Table 4, while the differences between various LLM models are not statistically significant, the employment of Reference Context Learning (RCL) and Retrieval-Augmented Generation (RAG) produces substantial performance enhancements. The study finds that the synergistic application of RCL and RAG leads to superior results compared to either technique used in isolation, establishing the combined approach as the most effective.

4 RESULTS AND DISCUSSION

Table 5 presents a comparative assessment of the question generation capability of three large language model platforms, namely OpenAI, Ollama, and Gemini, for three different experimental setups: baseline LLM, Reference Context Learning (RCL), and Retrieval-Augmented generation (RAG). The automatic evaluation criteria, such as BLEU [28], ROUGE (ROUGE-1, ROUGE-2, ROUGE-3) [29], METEOR [30], and F1 score, are used to assess the performance of the models. These criteria collectively assess the lexical overlap, semantic similarity, and content coverage of the generated questions and the reference questions.

The automatic evaluation criteria clearly indicate the improvement trajectory of the OpenAI model. The baseline setup showed very low scores for BLEU (0.015) and ROUGE, which clearly indicated the lack of n-gram overlap between the generated questions and the reference questions. However, the RCL setup showed a significant improvement in the scores, with ROUGE-1 and METEOR scores increasing to 0.327 and 0.257, respectively. The best performance was achieved by the RAG setup, which showed the highest scores for ROUGE-2 (0.318), ROUGE-3 (0.286), and F1 (0.432). This demonstrates that retrieving and incorporating external knowledge is most effective for improving contextual relevance and lexical coverage.

A similar pattern can be observed in the Ollama-based model. The baseline LLM arrangement performs mediocrely, especially in ROUGE-1 (0.222) and BLEU (0.003). Significant improvements are gained with the Reference Context Learning approach, particularly in METEOR (0.327) and F1 score (0.419), which suggest enhanced semantic alignment. Competitive ROUGE-2 (0.279) and ROUGE-3 (0.244) scores are produced by the RAG design, demonstrating the effectiveness of retrieval processes in increasing phrase-level similarity and overall completeness.

With ROUGE-2 and ROUGE-3 scores of zero, the baseline LLM results for the Gemini-based model are rather poor and show little higher-order n-gram overlap. However, the RAG-based technique produces additional improvements in ROUGE-2 (0.238) and ROUGE-3 (0.200), while the Reference Context Learning setting raises ROUGE-1 to 0.291 and F1 to 0.318.

Table 3: Sample Content: taken from RAG file

S.No	Context	QT	BT	Keywords	Sample Questions
1	Definition of a byte in computer science..	L1	L1	define	A byte consists of 8 bits.
2	Explanation of Python's typing system..	L1	L2	describe	Python uses dynamic typing.
3	Comparison between HTTP and HTTPS protocols..	L1	L3	compare	HTTP and HTTPS are the same protocol.
4	Usage of Java's 'final' keyword in inheritance..	L1	L4	use	The 'final' keyword in Java prevents method overriding.
5	Experiment with Dijkstra's algorithm limitations..	L1	L5	experiment	Dijkstra's algorithm works for graphs with negative weights.
6	Design consideration for REST API methods..	L1	L6	design	In REST APIs, PUT and PATCH perform identical operations.
7	Identifying sorting algorithms by time complexity..	L2	L1	define	The _____ sorting algorithm has a worst-case time complexity of O(n log n).
8	SQL clause for filtering grouped data..	L2	L2	describe	In SQL, _____ is used to filter groups.
9	Recognizing FIFO-based data structures..	L2	L3	compare	The _____ data structure uses FIFO order.
10	Python feature for creating anonymous functions..	L2	L4	use	In Python, _____ is used to create an anonymous function.
11	Network knowledge about HTTPS default port..	L2	L5	experiment	_____ is the default port for HTTPS.
12	Protocol responsible for DNS resolution..	L2	L6	design	The _____ protocol resolves domain names to IP addresses.
13	OOP concept of inheritance with hint..	L3	L1	define	(Hint: OOP): _____ allows a class to inherit properties from another.
14	TCP reliability mechanism with hint..	L3	L2	describe	(Hint: Network protocol): _____ ensures reliable data delivery in TCP.
15	LIFO principle in data structures with hint..	L3	L3	compare	(Hint: Data structure): _____ is used for LIFO operations.
16	Python decorator for static methods with hint..	L3	L4	use	(Hint: Python): _____ decorator marks a function as static.
17	Graph algorithm for shortest path with hint..	L3	L5	experiment	(Hint: Algorithm): _____ is used to find the shortest path in a weighted graph.
18	CPU scheduling algorithm with hint..	L3	L6	design	(Hint: OS): _____ scheduling algorithm allocates CPU time in circular order.
19	Multiple-choice question about interpreted languages..	L4	L1	define	Which language is interpreted? a) C b) Java c) Python d) C++
20	Python type system behaviour with strings and numbers..	L4	L2	describe	What is the output of '5 + "5"' in Python? a) 10 b) 55 c) Error d) 5

.

Table 4: Lexical evaluation of generated questions: normalized score [0-1].

S.No	LLM	Model	BLEU	ROUG-1	ROUG-2	ROUG-3	METEOR	F1
1	OpenAi	*LLM*	0.015	0.231	0.08	0.042	0.182	0.23
		RCL	0.023	0.327	0.17	0.133	0.257	0.35
		RAG	0.024	**0.348**	**0.318**	**0.286**	0.297	**0.43**
2	Ollama	*LLM*	0.003	0.222	0.14	0.098	0.187	0.27
		RCL	**0.028**	0.346	0.2	0.125	**0.327**	0.41
		RAG	0.012	0.311	0.279	0.244	0.261	0.38
3	Gemini	*LLM*	0.002	0.204	0	0	0.184	0.25
		RCL	0.006	0.291	0	0	0.254	0.31
		RAG	0.005	0.273	0.238	0.2	0.225	0.34

The findings show that RAG regularly outperforms baseline LLM and RCL configurations on every platform. ROUGE and F1 measures, which more accurately reflect the semantic relevance and quality necessary for question production, are where this superiority is most obvious. While Reference Context Learning (RCL) provides a valuable step up from zero-shot prompting, it remains an intermediate solution. Notably, the models' low BLEU scores indicate that BLEU is not a good fit for tasks requiring lexical diversity; on the other hand, ROUGE, METEOR, and F1 provide a more accurate assessment of content coverage.

Table 5: Human evaluation of generated questions: Likart score [0-5].

S.No	LLM	Model	Grm	Rlv	Clr	Usf	Anw	Spy	F-Sco
1	OpenAi	*LLM*	4	3	3	3	3	3	3.16
		RCL	4	2	3	3	4	3	3.16
		RAG	5	**5**	**4**	**4**	4	**4**	**4.33**
2	Ollama	*LLM*	4	4	3	3	4	3	3.50
		RCL	**4**	3	4	4	**4**	4	3.83
		RAG	5	4	4	4	5	5	4.50
3	Gemini	*LLM*	5	4	3	3	3	2	3.33
		RCL	4	4	4	3	3	3	3.50
		RAG	5	4	4	4	3	3	3.83

Qualitative results in Table 5 above further confirm these results. Although the baseline and RCL settings of OpenAI resulted in the same average score of 3.17, the RAG-based approach resulted in

a better score of 4.33. Most importantly, RAG received a perfect score of 5 on relevance, emphasizing its capability to make use of external context information for generating high-quality and specific questions. Analysis of Ollama and Gemini-based models shows that the Retrieval-Augmented Generation (RAG) approach is the best-performing configuration, as it results in the highest final scores and performs best on relevance, answerability, and specificity.

Among the models, Ollama showed the greatest improvement with RAG, as its score improved from 3.50 to 4.50. Although the Reference Context Learning (RCL) approach showed moderate improvement in clarity and usability, its effect was always overshadowed by that of RAG. The overall results clearly show a performance hierarchy: Baseline < RCL < RAG < RAG + RCL. This result clearly shows that the best possible approach is to combine the capability of external knowledge retrieval (RAG) with example-based prompting (RCL). Moreover, this study confirms that human-centric evaluation criteria are necessary for measuring these subtle qualitative improvements.

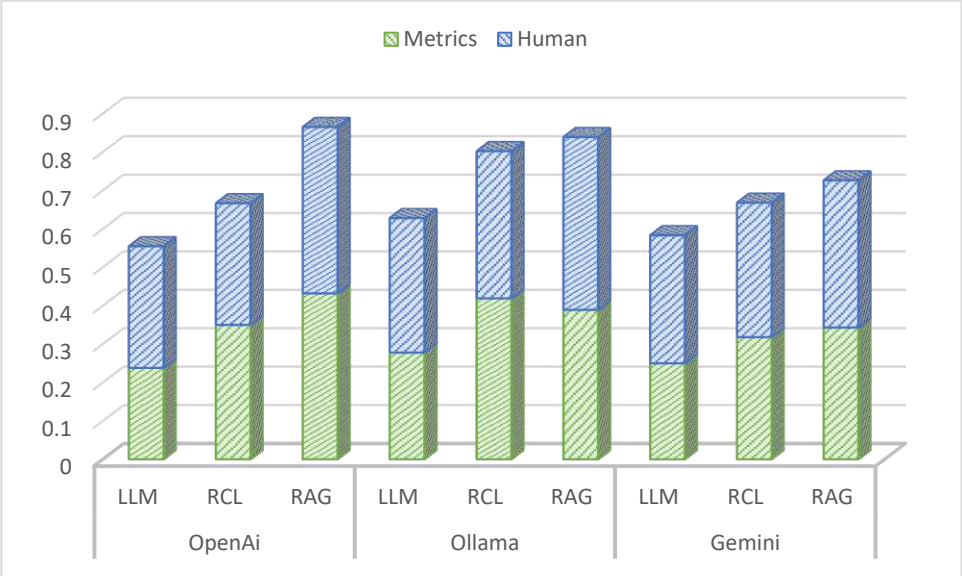


Fig. 3. Performance Comparison: Question generation with various approaches

5 CONCLUSIONS AND FUTURE WORK

The questions that are specific to exams and based on the syllabus cannot be generated by enhancing the overall capability of the LLM. Rather, by keeping the LLM within certain constraints, better results can be achieved, which will allow the LLM to generate questions in a more focused way. The fact that the technique performs well on all types of questions and levels of Bloom's taxonomy indicates that an LLM with a linear bound can perform very well if it is given the right input, such as proper prompts, few-shot examples, and reference questions using RAG. This technique can be improved in the future to generate mathematical and programming-related questions, along with text questions.

Declarations

Acknowledgments:
The authors would like to acknowledge Central University of Tamil Nadu, India for providing the necessary academic and computational support to conduct this research.

Funding:
No external funding was received for this research.

Author contributions:
[SENTHILKUMAR P]: Conceptualization, Methodology, Software, Writing – original draft, Validation.
[NANDHINI K]: Formal analysis, Writing – review & editing, Supervision.

Competing interests:
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics and Consent to Participate declarations: not applicable

Consent to publish declaration: All authors have read and approved the final version of the manuscript and consent to its publication. The authors confirm that the work is original and has not been published elsewhere.

Data availability:
The study utilized publicly available datasets and open-source tools.

Code availability:
The code used in this study is available from the corresponding author upon reasonable request.

References

[1] I. Zliobaite *et al.*, “Next challenges for adaptive learning systems,” *ACM SIGKDD Explor. Newsl.*, vol. 14, no. 1, pp. 48–55, Dec. 2012, doi: 10.1145/2408736.2408746.

[2] N.-T. Le, T. Kojiri, and N. Pinkwart, “Automatic Question Generation for Educational Applications – The State of Art,” in *Advanced Computational Methods for Knowledge Engineering*, vol. 282, T. Van Do, H. A. L. Thi, and N. T. Nguyen, Eds., in Advances in Intelligent Systems and Computing, vol. 282. , Cham: Springer International Publishing, 2014, pp. 325–338. doi: 10.1007/978-3-319-06569-4_24.

[3] B. Zou, P. Li, L. Pan, and A. T. Aw, “Automatic True/False Question Generation for Educational Purpose,” in *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, Seattle, Washington: Association for Computational Linguistics, 2022, pp. 61–70. doi: 10.18653/v1/2022.bea-1.10.

[4] M. Divate and A. Salgaonkar, “Automatic Question Generation Approaches and Evaluation Techniques,” *Curr. Sci.*, vol. 113, no. 09, p. 1683, Nov. 2017, doi: 10.18520/cs/v113/i09/1683-1691.

[5] B. Das, M. Majumder, A. A. Sekh, and S. Phadikar, “Automatic question generation and answer assessment for subjective examination,” *Cogn. Syst. Res.*, vol. 72, pp. 14–22, Mar. 2022, doi: 10.1016/j.cogsys.2021.11.002.

[6] S. Pal *et al.*, “Weakly Supervised Context-based Interview Question Generation”.

[7] G. Biancini, A. Ferrato, and C. Limongelli, “Multiple-Choice Question Generation Using Large Language Models: Methodology and Educator Insights,” in *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, Jun. 2024, pp. 584–590. doi: 10.1145/3631700.3665233.

[8] M. A. Ehsan, A. S. M. M. Hasan, K. B. Shahnoor, and S. S. Tasneem, “Automatic Question & Answer Generation Using Generative Large Language Model (LLM),” Sep. 28, 2025, *arXiv*: arXiv:2508.19475. doi: 10.48550/arXiv.2508.19475.

[9] N. E. Adams, “Bloom’s taxonomy of cognitive learning objectives,” *J. Med. Libr. Assoc. JMLA*, vol. 103, no. 3, pp. 152–153, Jul. 2015, doi: 10.3163/1536-5050.103.3.010.

[10] S. K. Sarkar, “Bloom’s Taxonomy and Examination Reform in Higher Education using ICT as a Tool,” *J. Adv. Educ. Philos.*, vol. 7, no. 05, pp. 173–177, May 2023, doi: 10.36348/jaep.2023.v07i05.005.

[11] B. Das, M. Majumder, S. Phadikar, and A. A. Sekh, “Automatic generation of fill-in-the-blank question with corpus-based distractors for e-assessment to enhance learning,” *Comput. Appl. Eng. Educ.*, vol. 27, no. 6, pp. 1485–1495, Nov. 2019, doi: 10.1002/cae.22163.

[12] D. R. Ch and S. K. Saha, “Automatic Multiple Choice Question Generation From Text: A Survey,” *IEEE Trans. Learn. Technol.*, vol. 13, no. 1, pp. 14–25, Jan. 2020, doi: 10.1109/TLT.2018.2889100.

[13] Z. Jiang *et al.*, “Active Retrieval Augmented Generation,” Oct. 22, 2023, *arXiv*: arXiv:2305.06983. doi: 10.48550/arXiv.2305.06983.

[14] A. Salemi and H. Zamani, “Evaluating Retrieval Quality in Retrieval-Augmented Generation,” Apr. 21, 2024, *arXiv*: arXiv:2404.13781. doi: 10.48550/arXiv.2404.13781.

[15] A. Fabbri, P. Ng, Z. Wang, R. Nallapati, and B. Xiang, “Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 4508–4513. doi: 10.18653/v1/2020.acl-main.413.

[16] D. Naber, “A Rule-Based Style and Grammar Checker”.

[17] H. Schmid, “Part-of-speech tagging with neural networks,” in *Proceedings of the 15th conference on Computational linguistics -*, Kyoto, Japan: Association for Computational Linguistics, 1994, p. 172. doi: 10.3115/991886.991915.

[18] C. Raffel *et al.*, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”.

[19] Y. Cao *et al.*, “A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT,” Mar. 07, 2023, *arXiv*: arXiv:2303.04226. doi: 10.48550/arXiv.2303.04226.

[20] K. Li and Y. Zhang, “Planning First, Question Second: An LLM-Guided Method for Controllable Question Generation,” in *Findings of the Association for Computational Linguistics ACL 2024*, Bangkok, Thailand

and virtual meeting: Association for Computational Linguistics, 2024, pp. 4715–4729. doi: 10.18653/v1/2024.findings-acl.280.

[21] N. Scaria, S. D. Chenna, and D. Subramani, "Automated Educational Question Generation at Different Bloom's Skill Levels using Large Language Models: Strategies and Evaluation," vol. 14830, 2024, pp. 165–179. doi: 10.1007/978-3-031-64299-9_12.

[22] Prof. Kanade P. G., Jayesh Gudaghe, Sanjasvi Sonawane, Mahek Shaikh, and Sarthak Ghumare, "Question Paper Generation using Bloom's Taxonomy," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 151–155, Nov. 2024, doi: 10.48175/IJARSCT-22131.

[23] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks".

[24] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," Jul. 22, 2020, *arXiv*: arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165.

[25] H. Chai, R. Chen, and H. Zhu, "RAG-Based Language Model with Enhanced Indexing Optimization for Financial Intelligent Customer Service Application".

[26] A. Vaswani *et al.*, "Attention Is All You Need," Aug. 02, 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[27] M. Heilman and N. A. Smith, "Good Question! Statistical Ranking for Question Generation".

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, Philadelphia, Pennsylvania: Association for Computational Linguistics, 2001, p. 311. doi: 10.3115/1073083.1073135.

[29] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries".

[30] A. Lavie and A. Agarwal, "Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments," in *Proceedings of the Second Workshop on Statistical Machine Translation - StatMT '07*, Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 228–231. doi: 10.3115/1626355.1626389.