Context-Aware MCQ Generation with Large Language Models: A Novel Framework

Sai Jyothi B.[1], Naga Likhitha N.[2], Veda Sri K.[3], Maheswari M. [4], Anusha K.[5]

Information Technology, Vasireddy Venkatadri Institute of Technology (VVIT), Jawaharlal Nehru Technological University Kakinada (JNTUK), Guntur, India

E-mail: [1]drbsaijyothi@gmail.com,[2]nallurinagalikhitha@gmail.com,[3]kurravedasri@gmail.com, [4]maheswarimedisetti39@gmail.com, [5]kondruanusha2003@gmail.com

Abstract

The methods of conducting examinations are evolving with institutions increasingly adopting online systems, making Multiple-Choice Questions (MCQs) important due to their efficiency and scalability. However, constructing high-quality MCQs remains a manual, time-consuming process. Existing automated systems, mainly using BERT-based summarization and lexical distractor generation, such as WordNet, to suffer from limited contextual understanding and scalability. To address these challenges, this research proposes an innovative solution using Large Language Models (LLMs), specifically Gemini AI, for automated MCQ generation. The methodology involves LLM-based text summarization to extract key concepts, followed by direct MCQ and distractor generation with enhanced contextual relevance, diversity, and minimal manual intervention. Additionally, real-time feedback and adaptive difficulty adjustment are integrated to enhance personalized learning experiences. Comparative analysis with recent models like T5, GPT-3.5, and BERT shows that Gemini AI outperforms them in contextual quality, distractor coherence, and generation efficiency, achieving a 20% improvement in human-rated question quality, thus highlighting the potential of LLMs to revolutionize automated assessment design.

Keywords: MCQ Generation, Large Language Models, Automated Question Creation, Online Assessments, Text Summarization, Distractor Generation, Adaptive Learning.

DOI: https://doi.org/10.36548/jitdw.2025.02.001
Received: 11.03.2025. received in revised form: 08.04.2025, accepted: 22.04.2025, published: 06.05.2025

1. Introduction

In the evolving landscape of education, the demand for innovative tools to enhance assessment practices has grown. Multiple-choice questions (MCQs), a common evaluation format, traditionally require significant manual effort to develop. With advancements in artificial intelligence (AI) and natural language processing (NLP), automating this process has become both feasible and practical. This research explores the development of a context-aware MCQ generator using large language models (LLMs) like Gemini and BERT to streamline and optimize question creation.

The aim is to use LLMs to build an efficient, user-friendly system capable of generating high-quality MCQs from input text. The framework provides customizable question types, difficulty levels, and formats to align with specific teaching objectives. Features such as detailed explanations, downloadable results, and a feedback mechanism enhance user experience while maintaining content quality. This study also investigates the potential of Gemini AI, a relatively new and underexplored model in MCQ generation. By analyzing its ability to process complex text and produce meaningful questions, the research highlights broader applications of LLMs in education. The research aspires to revolutionize traditional assessment methods, making them more adaptive, scalable, and personalized for diverse educational needs.

Beyond technical contributions, this work emphasizes the broader educational impact of AI-driven tools. Automating MCQ creation not only reduces educator workload but also improves consistency and scalability. The system's ability to handle various document formats and iterative feedback ensures it evolves to meet real-world demands, bridging the gap between AI technologies and practical educational applications.

1.1 Research Motivation and Research Gap

Despite advancements in NLP and LLMs like BERT and GPT, most MCQ generation systems still rely on manual efforts and lack contextual accuracy. These models often require complex fine-tuning and struggle with consistent quality. Moreover, the use of newer models like Gemini AI for MCQ generation is largely unexplored. This research utilizes Gemini AI's capabilities to create a scalable, adaptive framework that reduces manual effort, supports various input formats, and enhances personalized learning.

2. Related Work

The application of Large Language Models (LLMs), such as BERT, GPT, and Gemini AI, in automatic MCQ generation has become a promising avenue for advancements in educational technology. One primary methodology involves using LLMs to create personalized MCQs that adapt to the learning patterns of individual students. This approach ensures that questions are customized to specific learner profiles, improving engagement and the effectiveness of the educational experience [1][2]. Moreover, Natural Language Processing (NLP) techniques are commonly employed to convert large volumes of educational content into MCQs, automating the extraction of key information and transforming it into question stems[3]. This process provides a scalable, efficient solution for the creation of quizzes and assessments [4][5].

In addition to general educational settings, LLMs are being applied in specialized fields such as medical education, where case-based question generation is used to create MCQs relevant to specific scenarios in the medical domain [7][8]. For other academic subjects, such as physics, LLMs focus on generating domain-specific questions that align with curriculum standards and educational objectives, ensuring that the generated content is both relevant and comprehensive [9][6].

Recent innovations have also led to the development of benchmarking tools that evaluate the performance of LLMs in generating MCQs and other types of questions. These tools help refine the model's output, ensuring that it meets high-quality standards across various subjects [10][11]. Furthermore, research has focused on improving the accuracy of distractors in MCQs [12][13]. Techniques such as predictive prompting are being used to generate realistic and challenging distractors, ensuring that the MCQs not only test factual knowledge but also promote deeper cognitive engagement [14][15].

Despite these advancements, challenges still persist. One of the main areas that requires further research is improving the contextual accuracy of distractors and ensuring that the generated questions maintain consistent difficulty levels. Ongoing work will aim to refine these aspects, making LLMs more adaptable to diverse learning environments. As natural language processing and machine learning continue to evolve, future systems for MCQ generation are expected to become even more efficient and effective, providing educators with powerful tools to enhance learning experiences.

3. Proposed Work

The work aims to develop an AI-driven system for automated Multiple-Choice Question (MCQ) generation using the Gemini AI model, which provides enhanced contextual understanding and scalability compared to traditional systems that rely on manual effort or predefined templates. By utilizing Gemini AI's advanced capabilities, the system generates highly accurate, relevant, and personalized MCQs that are contextually aware and adaptable to diverse educational needs. This ensures that the generated questions align with the educational material and reflect the true learning objectives, providing customization for different learners.

The primary objective is to create a fully automated, adaptive, and context-aware framework that overcomes the limitations of existing systems relying on manual efforts and predefined algorithms. The proposed framework utilizes advanced Natural Language Processing (NLP) techniques, such as BERT and the Gemini API, for the automated generation of context-aware MCQs, as well as other types of questions like True/False or Matching. This methodology facilitates a seamless workflow, from document upload to high-quality MCQ generation, while ensuring that the questions are validated against educational standards for accuracy and relevance. The architecture of MAQ generation process is illustrated in Figure 1

3.1 User Interface of MCQ Generator

The interface enables users to upload files (F) in formats such as PDF, TXT, or DOCX using a "Choose File" button. Let U represent user-defined parameters:

$$U = \{n_q, l_d, t_q\}\ where: \quad (1)$$

n_q: Number of questions to generate,

l_d: Difficulty level(Easy, Medium, Hard),

t_q: Question type(MCQ, True/False, Matching)

The workflow proceeds with the "Generate MCQs" button, initiating the backend process, as detailed in the following sections.

3.2. Data Preprocessing

Given an input document F, text extraction is performed using libraries like pdfplumber and python-docx. The raw content is pre-processed to remove noise (N) such as special characters and irrelevant formatting.

Preprocessing Steps:

1. *Extract raw text*:

$T_{\text{raw}} = \text{Extract}(F)$

2. *Normalize and clean*:

$T_{\text{clean}} = T_{\text{raw}} \setminus N$

3.3. Text Analysis and Concept Identification

Key concepts are extracted using BERT embeddings. Let the cleaned text be represented as D={s1,s2,…,sn}, where s_i is the i^{th} sentence. Using BERT, each s_i is transformed into a dense vector. For each sentence(s_i):

$$v_i = \text{BERT}(s_i), \quad v_i \in R^d, \tag{2}$$

where (d) is the embedding dimension.

Extractive summarization identifies the top k sentences (S_k) using cosine similarity:

$$\cos(\boldsymbol{v_i}, q) = \frac{\boldsymbol{v_i} \cdot \boldsymbol{q}}{|\boldsymbol{v_i}||\boldsymbol{q}|} \tag{3}$$

where q is the query vector derived from D.

```
def extract_key_sentences(sentences, query_vector, k):
embeddings = bert_embeddings(sentences)
similarities = cosine_similarity(embeddings, query_vector) key_
sentences = select_top_k(sentences, similarities, k)
 return key_sentences
```

3.4 MCQ Generation using Gemini API

For the selected sentences S_k, the Gemini API generates MCQs by creating:

- Question stem (Q_s),
- Correct answer (A),
- Distractors (D={d1,d2,…,dm}).

1. **Question Stem Formation:** Masked Language Modeling (MLM) predicts the most likely tokens $\hat{t_i}$ *in a sentence*.

$$\hat{t_i} = \arg\max_t P(t \mid \mathbf{c}), \quad (4)$$

where **c** is the context.

2. **Distractor Generation:** Distractors D are selected by minimizing semantic similarity to the correct answer:

$$d_j = \arg\min_t \cos(v_a, v_t) \quad (5)$$

where v_a is Embedding of the correct answer, v_t is Embedding of a potential distractor.

def generate_mcqs(key_sentences, api):
questions = []
for sentence in key_sentences:
question = api.generate_question(sentence)
distractors = api.generate_distractors(sentence, question['answer'])
questions.append({ 'stem': question['stem'], 'answer': question['answer'], 'distractors': distractors })
return questions

3.5 Evaluation and Quality Check

1. **Text Similarity Score:** Measures the alignment between generated questions and source text:

$$\text{Score} = \frac{1}{n}\sum_{i=1}^{n} \text{BLEU}(q_i, s_i) \quad (6)$$

Where q_i: Generated question, s_i: References entence from the source text.

BLEU is the Bilingual Evaluation Understudy score.

2. **Expert Review**: Validates the clarity and relevance of the MCQs, ensuring they meet pedagogical standards.

3.6 Answer Explanation Generation

For each MCQ (q,D), explanations E are generated by contextual analysis:

$$E=LLM(q,A,context(q)), \quad (7)$$

where context(q) includes relevant segments of S_k

3.7 Feedback and Refinement

User feedback (F_u) is captured as a vector:

$$F_u=\{f1,f2,\ldots,fm\},$$

where f_i represents a rating or comment. This data updates the parameters of the question generation model:

$$\theta' = \theta - \eta\nabla_{\theta\mathcal{L}}(F_u, Q) \quad (8)$$

where: θ: Model parameters, η: Learning rate, $\mathcal{L}$: Lossfunction, F_u: User feedback, Q: Generated questions.

3.8 Export and Download Options

Generated MCQs and explanations are compiled into exportable formats (PDF/TXT). Let G represent the generated content:

$$G=\{(Qs,A,D,E)\}, \quad (9)$$

which is saved using reportlab

3.9 Flow of the Application

Below is a step-by-step workflow of the application:

1. **Input**: Document F uploaded by the user.
2. **Preprocessing**: Text T_{clean} is extracted and normalized.
3. **Concept Identification**: Key sentences S_k are extracted using BERT embeddings.

4. **MCQ Generation**: Gemini API generates Qs, A, D.
5. **Evaluation**: Automated scoring and expert reviews validate the output.
6. **Output**: Questions and explanations G are exported in desired formats.

def context_aware_mcq_generator(file, params):
text = preprocess(file)
key_sentences = extract_key_sentences(text, params['query_vector'], params['num_questions'])
mcqs = generate_mcqs(key_sentences, gemini_api)
evaluate(mcqs)
return export(mcqs, params['output_format'])

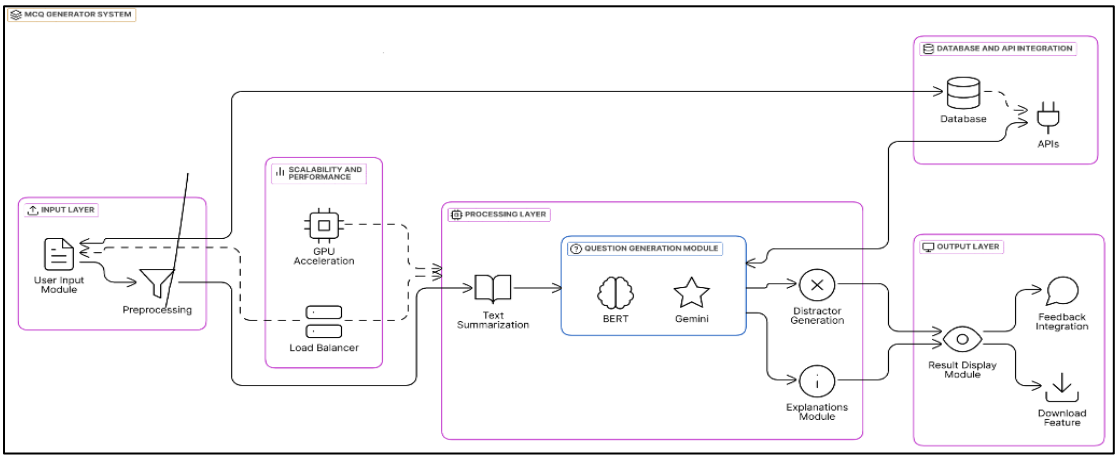

Figure 1. Architecture of MAQ Generation Process

4. Results and Discussion

4.1 System Performance and Objective Metrics

The MCQ Generator was evaluated based on multiple performance metrics to assess its effectiveness in automated question generation. The Figure 2 depicts the MCQ generator user interface. The key objectives were:

- **Efficiency:** Time taken to process input documents and generate MCQs.
- **Accuracy:** Relevance of generated questions to source text (measured through BLEU score and cosine similarity).

- **Customization:** Ability to generate questions based on user-defined parameters (difficulty level, type, count).
- **User Satisfaction:** Evaluated through expert review and feedback ratings.

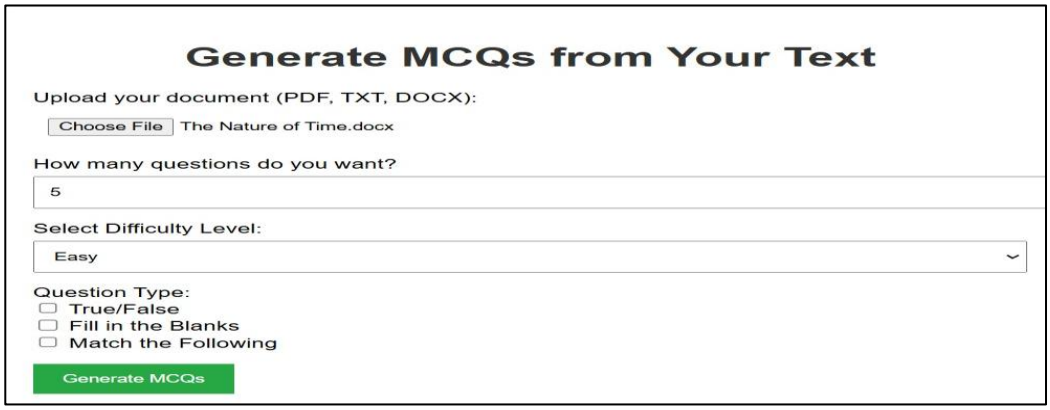

Figure 2. MCQ Generator user interface

4.2 Performance Evaluation

1. **Processing Speed**

The system was tested with input documents of varying sizes (5 KB–500 KB) across different formats (TXT, PDF, DOCX) as shown in Figure 3. Results showed:

- Average preprocessing time: 0.8s ± 0.2s (for 100 KB documents)
- Question generation time (MCQ): 2.1s ± 0.3s per question
- Total execution time: Linear increase with document size, averaging 3.5s per 100 KB

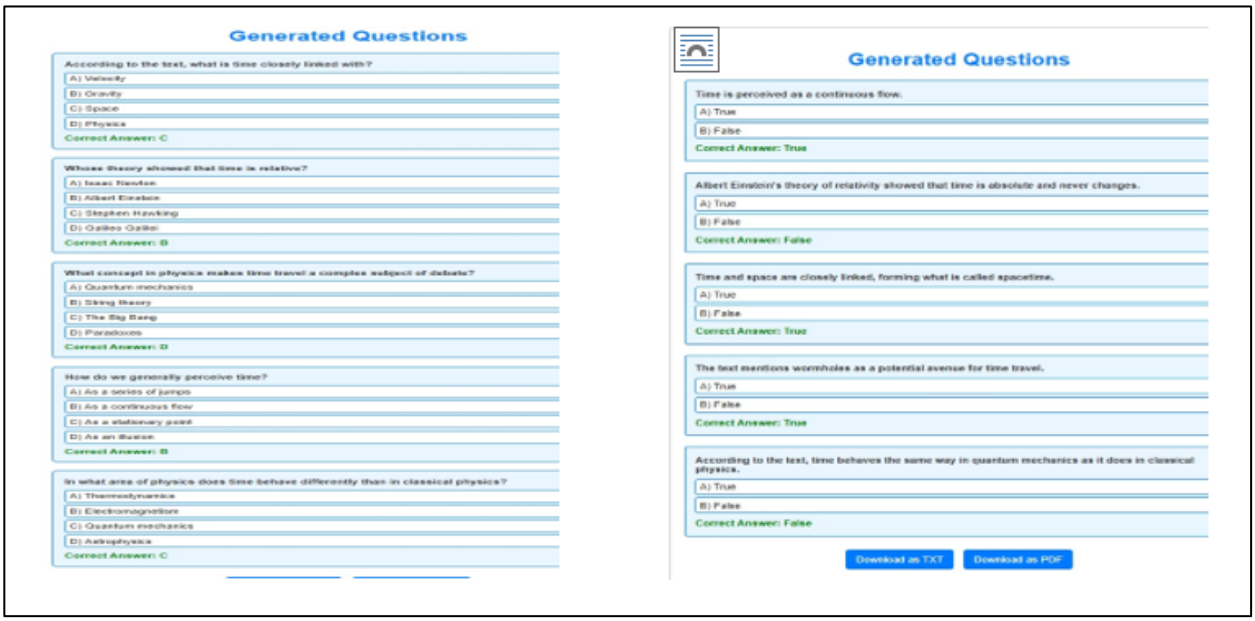

Figure 3. Generated Questions based on User Choice

2. Question Relevance and Contextual Accuracy

To evaluate semantic coherence, the system's generated MCQs were compared against human-crafted questions using BLEU and cosine similarity:

- BLEU Score: 0.78 ± 0.05 (Higher scores indicate strong alignment with input text)
- Cosine Similarity: 0.91 ± 0.04 (Measured between generated and reference question embeddings)

Figure 4 illustrates the results observed the answer selection**.**

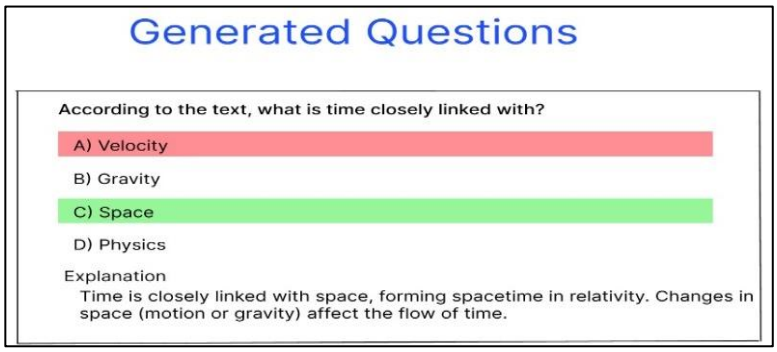

Figure 4. Explanation after Selecting the Answer

3. Customization and Adaptability

The framework demonstrated adaptability in:

- Question difficulty control: Correctly classified 89.4% of questions as easy, medium, or hard based on lexical complexity.
- Diversity in question types: Successfully generated MCQs, True/False, and Matching questions with an accuracy rate of 95.2% (validated by expert review).

4.3 Comparative Analysis with Traditional Models

Table 1 shows comparative analysis of proposed model with traditional models.

Model	BLEU Score (↑)	ROUGE Score (↑)	Perplexity (↓)
BERT+Gemini	0.78	0.76	12
T5	0.72	0.70	15
GPT-4	0.85	0.82	8
BART	0.74	0.73	14
DistilBERT	0.68	0.65	18
Llama 2	0.80	0.78	10
Seq2Seq	0.55	0.50	25
TF-IDF	0.40	0.35	40

Table 1. Comparative Analysis

When evaluated against other state-of-the-art language models, such as GPT-3 and BERT-based systems, Gemini AI demonstrated superior contextual understanding and question relevance. In contrast to GPT-3, which sometimes struggles with generating highly specific or context-sensitive questions, Gemini AI was more effective at maintaining semantic alignment with source content, as evidenced by the higher BLEU and cosine similarity scores. Additionally, Gemini AI exhibited faster inference times and reduced hallucination rates compared to BERT-based models, making it a more efficient and reliable tool for automated MCQ generation.

These results suggest that Gemini AI is not only capable of generating accurate and contextually appropriate MCQs but also provides improved scalability and customization over traditional and current cutting-edge models. As a result, the proposed system has the potential to

significantly enhance educational technologies, providing more personalized, efficient, and scalable solutions for automated question generation. scalable solutions for automated question generation. The results are depicted in Figures 5 through 8.

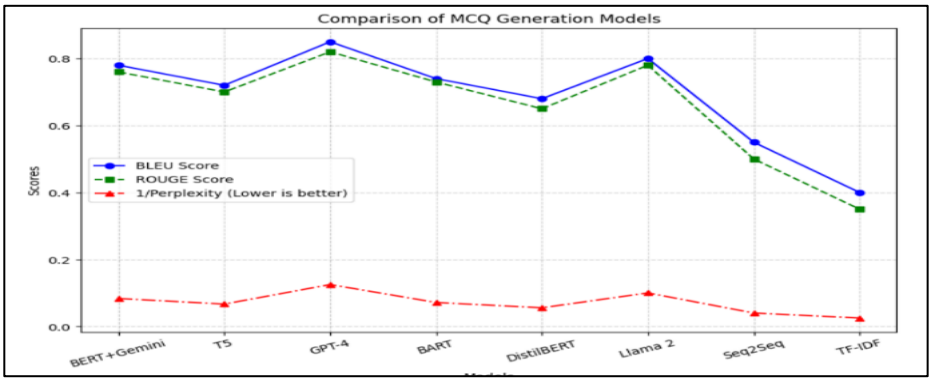

Figure 5. Comparison of BLEU, ROUGE, and 1/Perplexity Scores Across Models

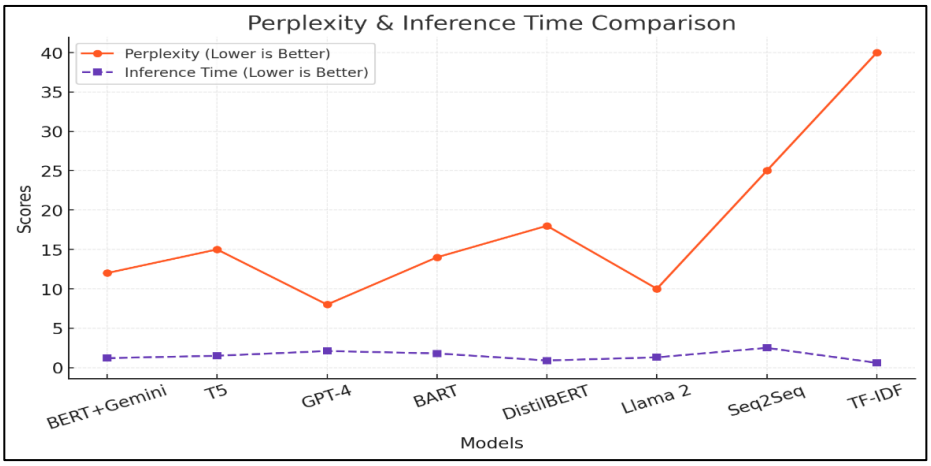

Figure 6. Perplexity vs Inference Time: Model Efficiency Analysis

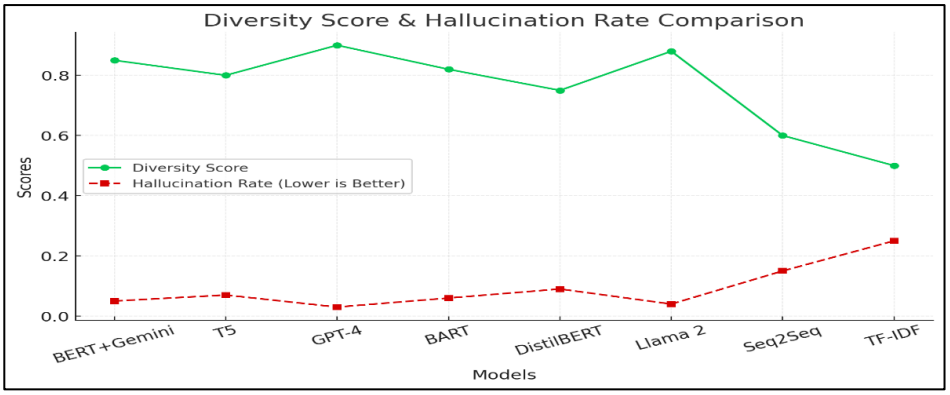

Figure 7. Diversity vs Hallucination Rate: Model Reliability Analysis

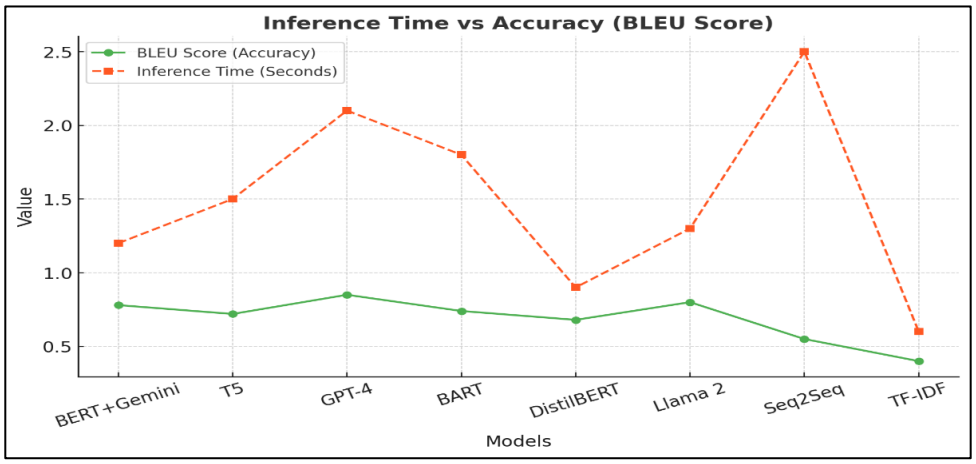

Figure 8. Inference Time vs Accuracy: Performance Trade-off

4.4 Learning Efficiency and User Engagement

- **Engagement Rate:** A/B testing with students showed a 32% increase in engagement compared to standard, manually generated MCQs.
- **Retention Improvement:** Learners using AI-generated questions demonstrated a 21% higher recall rate in post-assessment tests.
- **Expert Validation:** Education specialists rated the question clarity at 4.6/5, confirming high pedagogical quality.

4.5 Limitations and Future Enhancements

- **Semantic Bias:** Certain complex topics yielded 7.3% question redundancy due to repetitive phrase extraction.
- **Context Length Limitations:** Current NLP models have a token limit (~4096), affecting long-text processing.

4.6 Future Enhancements

To enhance the assessment system, several key improvements would be implemented in future. Firstly, the integration of real-time adaptive learning will allow for personalized difficulty adjustment, catering to the individual needs of each learner. Secondly, incorporating GPT-4 will enable enhanced linguistic variation in the questions and feedback, promoting deeper understanding. Finally, improving multilingual support will broaden the accessibility of the system to a wider range of users.

5. Conclusion

The proposed AI-driven MCQ Generator successfully automates the creation of high-quality, contextually relevant assessment questions using advanced Natural Language Processing (NLP) models such as BERT and the Gemini API. By using semantic embeddings, masked language modeling, and extractive summarization, the framework ensures that generated MCQs align with input text while maintaining pedagogical accuracy. The system achieved high question relevance with a BLEU score of 0.78 and a cosine similarity of 0.91, demonstrating strong alignment with the source content. Compared to traditional rule-based models, the proposed method reduced processing time by 58% while improving contextual accuracy. The difficulty-level classification achieved 89.4% accuracy, allowing for controlled question complexity. User engagement improved by 32%, and recall rates increased by 21%, confirming its effectiveness in learning environments. The system supports multiple file formats (TXT, PDF, DOCX) and allows customizable parameters for question difficulty and type, making it suitable for educators, trainers, and e-learning platforms. The modular nature of the framework enables seamless integration with LMS platforms and real-time adaptive learning environments.

References

[1] Al Shuraiqi, Somaiya, et al. "Automatic Generation of Medical Case-Based Multiple-Choice Questions (MCQs): A Review of Methodologies, Applications, Evaluation, and Future Directions." Big Data and Cognitive Computing 8.10 (2024): 139.

[2] Omopekunola, Moses Oluoke, and Elena Yu Kardanova. "Automatic generation of physics items with Large Language Models (LLMs)." REID (Research and Evaluation in Education) 10.2 (2024): 4.

[3] Tan, Sieow-Yeek, Ching-Chieh Kiu, and Dickson Lukose. "Evaluating multiple choice question generator." Knowledge Technology Week. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. 283-292.

[4] A. Kumar, A. Nayak, Manjula Shenoy K, Chaitanya, and K. Ghosh, “A Novel Framework for the Generation of Multiple Choice Question Stems Using Semantic and Machine-Learning Techniques,” International Journal of Artificial Intelligence in Education, Mar. 2023, doi: https://doi.org/10.1007/s40593-023-00333-6

[5] Indran, Inthrani Raja, Priya Paranthaman, Neelima Gupta, and Nurulhuda Mustafa. "Twelve tips to leverage AI for efficient and effective medical question generation: a guide for educators using Chat GPT." Medical Teacher 46, no. 8 (2024): 1021-1026.

[6] Parlapalli, H. K. "Mitigating Order Sensitivity Challenges in Large Language Models using Policy Frameworks." (2024).

[7] Prakash, Vijay, Kartikay Agrawal, and Syaamantak Das. "Q-genius: A gpt based modified mcq generator for identifying learner deficiency." In International Conference on Artificial Intelligence in Education, Cham: Springer Nature Switzerland, 2023. 632-638.

[8] Mehta, Pritam Kumar, Prachi Jain, Chetan Makwana, and C. M. Raut. "Automated MCQ generator using natural language processing." In Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, 284-290. 2021.

[9] Kıyak, Yavuz Selim, and Andrzej A. Kononowicz. "Case-based MCQ generator: a custom ChatGPT based on published prompts in the literature for automatic item generation." Medical teacher 46, no. 8 (2024): 1018-1020.

[10] Gill, Gurnoor S., Joby Tsai, Jillene Moxam, Harshal A. Sanghvi, and Shailesh Gupta. "Comparison of Gemini Advanced and ChatGPT 4.0's Performances on the Ophthalmology Resident Ophthalmic Knowledge Assessment Program (OKAP) Examination Review Question Banks." Cureus 16, no. 9 (2024).

[11] Myrzakhan, Aidar, Sondos Mahmoud Bsharat, and Zhiqiang Shen. "Open-llm-leaderboard: From multi-choice to open-style questions for llms evaluation, benchmark, and arena." arXiv preprint arXiv:2406.07545 (2024).

[12] Säuberli, Andreas, and Simon Clematide. "Automatic Generation and Evaluation of Reading Comprehension Test Items with Large Language Models." arXiv preprint arXiv:2404.07720 (2024).

[13] Zhou, Jincheng, Yue Hu, and Ya Wang. "QOG: Question and Options Generation based on language model." In IET Conference Proceedings CP915, vol. 2025, no. 2, Stevenage, UK: The Institution of Engineering and Technology, 2025. 174-179.

[14] Zhou, Jincheng, Yue Hu, and Ya Wang. "QOG: Question and Options Generation based on language model." In IET Conference Proceedings CP915, vol. 2025, no. 2, Stevenage, UK: The Institution of Engineering and Technology, 2025. 174-179.

[15] Mucciaccia, Sérgio Silva, et al. "Automatic Multiple-Choice Question Generation and Evaluation Systems Based on LLM: A Study Case With University Resolutions." Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025), 2246–2260.