An Automated MCQ Generator using NLP

Arpan Neupane[1]; Sonam Chaudhari[2]; Suruchi Shah[3]

[1,2,3]Dept. of Electronics and Computer Engineering Purwanchal Engineering Campus, IOE Dharan, Nepal

Publication Date: 2025/07/10

Abstract: The automation of multiple-choice question (MCQ) generation has emerged as a crucial advancement in educational assessment, aiming to reduce the time, effort, and domain expertise required for manual question creation. This research introduces a Natural Language Processing (NLP)-based system that generates high-quality, contextually relevant MCQs from textual content. The system accepts diverse input formats, including plain text and PDF documents, and utilizes advanced transformer models such as T5 (Text-to-Text Transfer Transformer), Flan-T5(Fine-tuned Language Net T5), and DistilBERT (Distilled Bidirectional Encoder Representations from Transformers) for keyword extraction, question formulation, and distractor generation. A web-based interface, developed using Django, enables users to customize parameters like question quantity and model selection, ensuring flexibility across educational domains. Evaluation using BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics confirms that fine-tuned models outperform their base counterparts in coherence and relevance. Additionally, the use of efficient fine-tuning techniques like LoRA (Low-Rank Adaptation) and QLoRA (Quantized Low-Rank Adaptation) significantly reduces computational overhead without degrading performance. The proposed system demonstrates strong potential to streamline formative assessment and enhance learning feedback loops, while future work will focus on mobile deployment and integration with digital learning platforms to expand accessibility.

Keywords: *Automated MCQ Generation; NLP; Transformer Models; Distractor Generation; Educational Technology*.

How to Cite: Arpan Neupane; Sonam Chaudhari; Suruchi Shah (2025). An Automated MCQ Generator using NLP. *International Journal of Innovative Science and Research Technology*, 10(6), 2941-2951.
https://doi.org/10.38124/ijisrt/25jun1878

I. INTRODUCTION

MCQ (Multiple-choice questions) serve as fundamental assessment tools in modern education due to their efficiency in standardized testing and digital learning platforms. The growing demand for these assessments has highlighted the need for automated generation methods, as manual creation requires educators to carefully balance question clarity, answer plausibility, and content coverage, often demanding significant time investment. Recent advances in natural language processing have enabled various automated approaches to question generation, from early rule-based systems that proved inflexible across domains to more recent neural methods using transformer architectures that still face limitations in distractor quality and domain adaptation. This research investigates how hybrid architectures combining multiple natural language processing models can overcome current limitations in automated question generation. By systematically integrating the strengths of different approaches while addressing their weaknesses, we propose a transformer-based NLP (Natural Language Processing) solution that maintains academic rigor while offering practical flexibility for educators. Our approach specifically focuses on preserving question quality across diverse subject domains and input formats to meet real-world educational requirements, combining the generative capabilities of language models with specialized quality control mechanisms.

➢ *Gap Identification:*

Despite advancements in automated MCQ generation, several gaps remain. Many systems produce questions that are either conceptually weak or lack linguistic accuracy and contextual relevance. Customization options are often limited, restricting users from selecting preferred models or controlling the number of questions. Additionally, most tools cannot process diverse input formats like PDFs, reducing their practical applicability.

Recent studies emphasize key limitations in current distractor generation approaches. Chung et al. [6] observed that BERT-based models frequently generate distractors lacking semantic relevance, particularly in domain-specific areas such as biomedicine. Similarly, Grover et al. [5] reported that T5 models trained on general-purpose datasets often fail to capture subject-specific terminology and contextual nuances, resulting in distractors that are either too generic or contextually inappropriate. These limitations highlight the need for more domain-aware and semantically consistent methods in automated distractor generation.

➢ *Objectives:*

The main objectives of this system are as follows:

- To develop an NLP-based MCQ generation system that processes diverse inputs, including raw text and PDFs.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun1878

- To Provide customizable features for PDF conversion, question count selection, and model choice.

II. LITERATURE REVIEW

➢ *Related Work:*

Several studies have explored automatic MCQ generation using NLP techniques. Folajimi and Omojola [1] utilized discourse connectives, words indicating logical relationships between sentences, to identify meaningful question candidates. Their system processed user-provided text to generate relevant questions using NLP methods. Nwafor and Onyenwe [2] proposed a three-step MCQ generation strategy: sentence selection, gap selection, and question formation, aiming to improve logical coherence in generated items. Susanti et al. [3] introduced a context-aware method for distractor generation in vocabulary-based MCQs by ranking candidates using semantic similarity and collocation measures. Das and Majumder [4] focused on factual cloze question generation using POS (Part-of-Speech) tagging rules to extract informative sentences and create fill in the blank questions, also providing hints to guide learner responses. Grover et al. [5] demonstrated the effectiveness of fine-tuned T5 (Text-to-Text Transfer Transformer) for question generation, showing improved relevance and quality over rule based methods. Chung et al. [6] developed a BERT (Bidirectional Encoder Representations from Transformers) based distractor generation model that applied multitask learning and negative answer training to produce plausible yet incorrect distractors, enhancing MCQ challenge and quality.

➢ *Related Theory*

In this project, the exploration of related theories serves as a crucial foundation for the development and implementation of the MCQ generation. The following topics encapsulate the diverse theoretical frameworks that guide various aspects of the project:

➢ *Natural Language Processing:*

Natural Language Processing (NLP) is a branch of AI that enables computers to understand and generate human language. It involves tasks like tokenization and is used in applications such as text generation, summarization, and question answering.

➢ *Multiple Choice Question Generation:*

MCQ generation involves automatically creating well-structured questions and answer choices from text. This project uses models like T5, BERT, and BART (Bidirectional and Auto-Regressive Transformers) for question generation, RAKE and DistilBERT (Distilled Bidirectional Encoder Representations from Transformers) for keyword extraction, and Sense2Vec (Sense-aware Word2Vec) and LLaMa (Large Language Model Meta AI) for distractor generation. The key challenge is ensuring clarity, accuracy, and relevance.

➢ *Language Models:*

The different language models used in this project are:

- *T5 Base Model:*

Google's T5-Base is a text-to-text transformer that reformulates NLP tasks as text generation, making it effective for MCQ creation. Using an encoder-decoder architecture, it processes input passages and generates grammatically correct, context-aware questions and answer choices. Fine-tuned on QA(Question Answering) datasets like SQuAD v2 (Stanford Question Answering Dataset version 2), it produces well-structured, high-quality MCQs automatically.

- *FLAN-T5 Model (Fine Tuned Language Net T5):*

Flan-T5-Small is a fine-tuned, instruction-optimized variant of T5 designed for better task understanding and efficiency. With the same encoder-decoder setup but smaller size, it generates coherent, contextually relevant MCQs from structured prompts, making it ideal for automated educational assessments.

- *DistilBERT:*

DistilBERT's architecture for MCQ generation tokenizes input text and uses a transformer encoder to capture contextual relationships, producing rich semantic embeddings. It generates questions by identifying key concepts and creates distractors via masked token prediction of plausible but incorrect options. Lightweight yet powerful, DistilBERT offers fast, accurate MCQ generation. Fine-tuned classifiers validate output quality. Integrated with Pandas and PyTorch, it enables an efficient, automated MCQ system.

- *LLaMa:*

LLaMa is a large-scale transformer model fine-tuned for instruction-following tasks, making it ideal for MCQ generation, keyword extraction, and distractor creation. It identifies key concepts to form clear questions, extracts relevant keywords, and generates plausible, context-aware distractors. Its advanced language understanding enables accurate, high-quality automated question creation.

- *BART:*

BART, a transformer model by Facebook, excels in text-to-text tasks like summarization and question generation. Its encoder-decoder design corrupts and reconstructs input to produce coherent, context-relevant MCQ stems, answer choices, and distractors. Pretrained on large datasets and fine-tuned on QA data, BART delivers high-quality questions. Integrated with Pandas and TensorFlow, it enables an efficient, automated MCQ generation pipeline for educational use.

➢ *Traditional NLP Techniques and Word Vector Models:*

- *RAKE(Rapid Automatic Keyword Extraction):*

RAKE (Rapid Automatic Keyword Extraction) is an unsupervised keyword extraction algorithm that identifies key terms by analyzing word co-occurrence patterns and frequency. It efficiently ranks candidate keywords without requiring training data, enabling automated generation of relevant MCQ stems and distractors through meaningful concept extraction. This lightweight approach supports context-aware question creation.

- *spaCy:*

spaCy is used in MCQ generation for efficient keyword extraction. It leverages NLP techniques like dependency parsing and named entity recognition to identify key terms from text. These keywords help create relevant question stems, options, and distractors. spaCy's speed and accuracy make it ideal for scalable, content-focused question generation

- *Sense2Vec:*

Sense2Vec is an advanced word embedding model that enhances distractor generation by considering word senses, part-of-speech tags, and entity types. It helps create context-aware distractors that are semantically similar but meaningfully distinct, making MCQs more challenging and less predictable.

- *WordNet:*

WordNet is a lexical database used in MCQ generation to create meaningful and challenging distractors. It provides synonyms, antonyms, hypernyms, and hyponyms, helping generate context-aware, semantically relevant options that are close but incorrect. By organizing words into synsets, WordNet ensures distractors are diverse, accurate, and not too obvious or unrelated.

III. METHODOLOGY

The system accepts raw text or PDF uploads, extracting text automatically from PDFs. Inputs over 500 tokens are chunked for efficiency. Keywords are extracted using methods like spaCy, RAKE, or DistilBERT, which guide transformer models to generate questions and distractors. Duplicate distractors are removed in post-processing. Final MCQs are saved in a database for authenticated users or in session storage for guests.

➢ *Data Collection and Model Fine-Tuning:*

In this study, relevant datasets were collected from diverse sources to fine tune models for tasks including question generation, keyword extraction, and distractor generation. For question generation, the T5 base model was fine tuned using the SQuAD v2 dataset, comprising approximately 130,000 question answer pairs, to enhance its ability to generate contextually appropriate and well formed questions. Additionally, the Flan T5 Small model was fine tuned on several domain specific datasets, namely ScienceQA, General Knowledge, and Biomedical QA datasets, to improve performance in generating accurate and domain relevant multiple choice questions (MCQs). Keyword extraction leveraged a hybrid approach combining RAKE, spaCy, and a DistilBERT model fine tuned on the SciQ dataset, which contains roughly 13,000 entries. For distractor generation, LLM T5 and Llama3.2 models were fine tuned on the RACE dataset, consisting of 27,000 instances, alongside supplementary scientific datasets from Hugging Face, focusing on producing plausible distractors.

➢ *System design:*

The Schematic Diagram of the developed system is given in the figure 1:

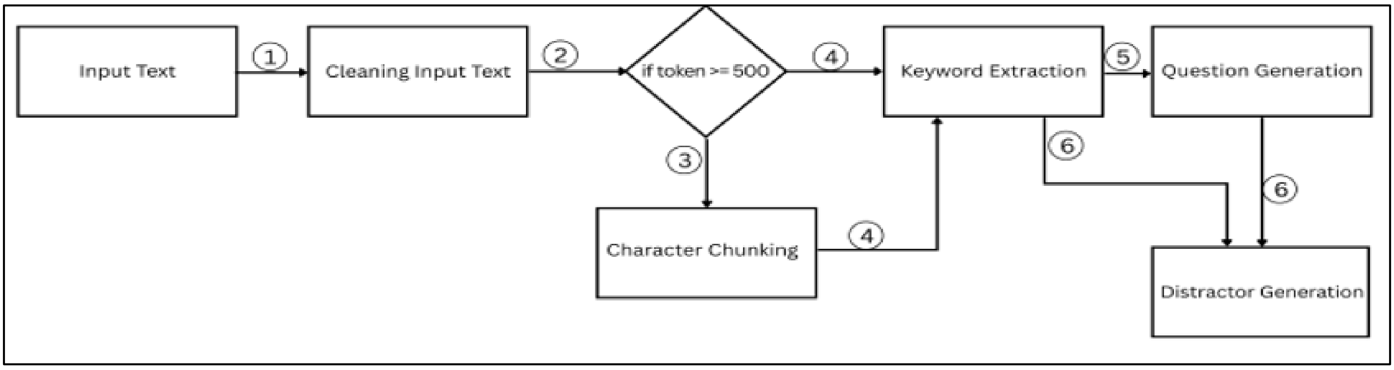

Fig 1 System Block Diagram

- *Input text Cleaning:*

The preprocessing of the input text is carried out using pandas, and all unnecessary elements such as symbols, HTML tags, numbering, bullets, and arrows are removed from the text. This leaves only textual content that is of importance to process further.

- *Tokenization and Chunking:*

If the input exceeds 500 tokens (approximately 2500 characters), it is segmented into smaller chunks to ensure effective processing by NLP models while preserving contextual coherence. For shorter inputs, keyword extraction is performed directly without chunking.

- *Keyword Extraction:*

After the text is divided through chunking, keyword extraction is carried out to identify the most significant terms required for effective question generation. This process is performed using RAKE and spaCy, which incorporates a BART model fine-tuned for keyword identification. Additionally, a DistilBERT model, fine-tuned via QLoRA on science-specific datasets, is utilized to further improve the contextual relevance and accuracy of the extracted keywords

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun1878

- *Question Generation:*

Question generation follows the keyword extraction phase. The system employs two models: T5-base, fine-tuned on the SQuAD v2 dataset, for general-purpose question generation, and Flan-T5 Small, adapted using domain-specific datasets such as ScienceQA, GeneralKnowledge, and Biomedical QA. These models utilize the extracted keywords and context to produce coherent and structurally sound questions.

- *Distractor Generation:*

After question formulation, distractors (incorrect options) are generated using a combination of Sense2Vec and WordNet. Sense2Vec first suggests candidates based on word similarity; if it fails, the system falls back on WordNet for synonyms. These methods are combined to ensure at least three distractors. Additionally, (*Large Language Model based on T5 architecture*) LLM-T5 (fine-tuned on the RACE dataset) and LLaMA (LoRA-tuned on science-domain data using the Unsloth platform) generate context-aware distractors based on the keyword and question, ensuring they are both plausible and challenging.

➢ *Selection of Transformer Models for MCQ Generation*

The choice of T5 and Flan T5 over large generative models such as GPT 3 or GPT 4 was due to their unified text-to-text framework, which is better suited for producing structured outputs like multiple-choice questions. While GPT models excel in open-ended generation, they often need postprocessing to fit MCQ formats, such as enforcing fixed answer choices. In contrast, the encoder-decoder architecture of T5 allows direct fine-tuning on task-specific templates, enhancing control and reducing variability. Flan T5 further improves this through instruction-based fine-tuning, enabling efficient domain adaptation without the high computational cost of larger language models.

➢ *Tools and Technique:*

- *Python:*

The core programming language used for algorithm implementation and backend development due to its simplicity and rich ecosystem.

- *Pandas:*

Used for data manipulation and preprocessing tasks, aiding in the handling of textual data before feeding it into NLP models.

- *PyTorch:*

Utilized for building and training NLP models, leveraging dynamic computation graphs and GPU acceleration

- *Hugging Face:*

Provided pre-trained NLP models and libraries critical for natural language understanding and question generation.

- *QLoRA(Quantized Low-Rank Adaptation) :*

A memory-efficient fine-tuning method utilizing 4-bit quantization and adapter layers to enable large model training on limited hardware while maintaining accuracy.

- *LoRA(Low-Rank Adaptation):*

A parameter-efficient fine-tuning technique that adds small trainable layers to large models, reducing resource requirements and accelerating training without compromising performance

- *Django:*

A high-level Python web framework used to develop the web backend, enabling rapid development and secure management of the application.

- *HTML, CSS, and JavaScript:*

Technologies used to create the responsive and interactive web frontend, facilitating user interaction such as MCQ generation and PDF uploads.

➢ *Evaluation Metrices:*

To assess the performance of the MCQ generation system, both automatic and training-based evaluation metrics were employed. These metrics help evaluate the contextual accuracy, structure, and relevance of the generated questions.

- *BLEU (Bilingual Evaluation Understudy):*

BLEU is an automated metric used to evaluate the quality of generated MCQ stems by comparing them to human-written references. It calculates the n-gram overlap (precision) between generated and reference texts. While a higher BLEU score indicates closer similarity, it does not fully capture semantic accuracy or creativity and is best complemented with human evaluation.

$$\boxed{\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \cdot \log p_n\right)}$$

Where:

- P_n = Modified *n*-gram precision, calculated for *n* ranging from 1 to *N* (commonly up to 4), representing the proportion of *n*-gram overlaps between the generated and reference texts.
- W_n = weights for each n-gram
- BP(Brevity Penalty) = A factor introduced to penalize excessively short candidate outputs, defined as:

$$\boxed{\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}}$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun1878

Where:

- C = Length of the candidate (generated) sentence.
- R = Length of the reference sentence.

- *ROUGE (Recall-Oriented Understudy for Gisting Evaluation):*

ROUGE evaluates the quality of generated text by measuring overlap with reference text, focusing on recall. In MCQ generation, it helps assess whether the question stem captures key information. However, it mainly considers word overlap and may not fully reflect grammatical accuracy or distractor quality.

$$\boxed{\text{ROUGE-N} = \frac{\sum_{\text{ref}\in\text{Refs}} \sum_{\text{gram}_n \in \text{ref}} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{\text{ref}\in\text{Refs}} \sum_{\text{gram}_n \in \text{ref}} \text{Count}(\text{gram}_n)}}$$

Where:

- Gram_n = Represents an *n*-gram, such as unigrams (single words), bigrams (two-word sequences), etc
- $\text{Count}_{\text{match}}(\text{gram}_n)$ = Denotes the number of matching *n*-grams between the generated output and the reference text.
- Refs = Refers to the set of reference texts used for comparison.

- *Training Loss:*

Training loss indicates how well the model learns from the training data. Calculated using loss functions like cross-entropy, lower values signify better alignment with expected outputs

- *Validation Loss:*

Validation Loss evaluates model performance on unseen data. A low validation loss implies strong generalization, while a significant gap between training and validation loss may suggest overfitting.

IV. RESULT AND DISCUSSION

➢ *System Overview:*

The MCQ Generator effectively processes input text to extract keywords, generate questions, and produce relevant distractors. T5-base performed well for general questions, while Flan-T5 Small improved accuracy on science content. RAKE, spaCy, and DistilBERT enabled robust keyword extraction, and Sense2Vec, LLM-T5, and LLaMA enhanced distractor quality.

For the system specification, the parameter required were configured as mentioned below in Table 1.

Table 1 System Specification Table

S. N	Particulars	System Specifications
1	GPU Type	RTX4090
2	GPU Size	32 GB
3	RAM size	32 GB

➢ *Training and Evaluation of T5 Base Model:*

- *Model Training:*

For the Training specification of T5 base, the parameter required were configured as mentioned below in Table 2

Table 2 Training Specification Table for T5 base model

S. N	Particulars	Applied Specifications
1	Number of Epochs	7
2	Batch Size	16
3	Optimizer	AdamW
4	Learning Rate	3e-10
5	Momentum	0.9
6	Decay	0.01

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun1878

- *Evaluation:*

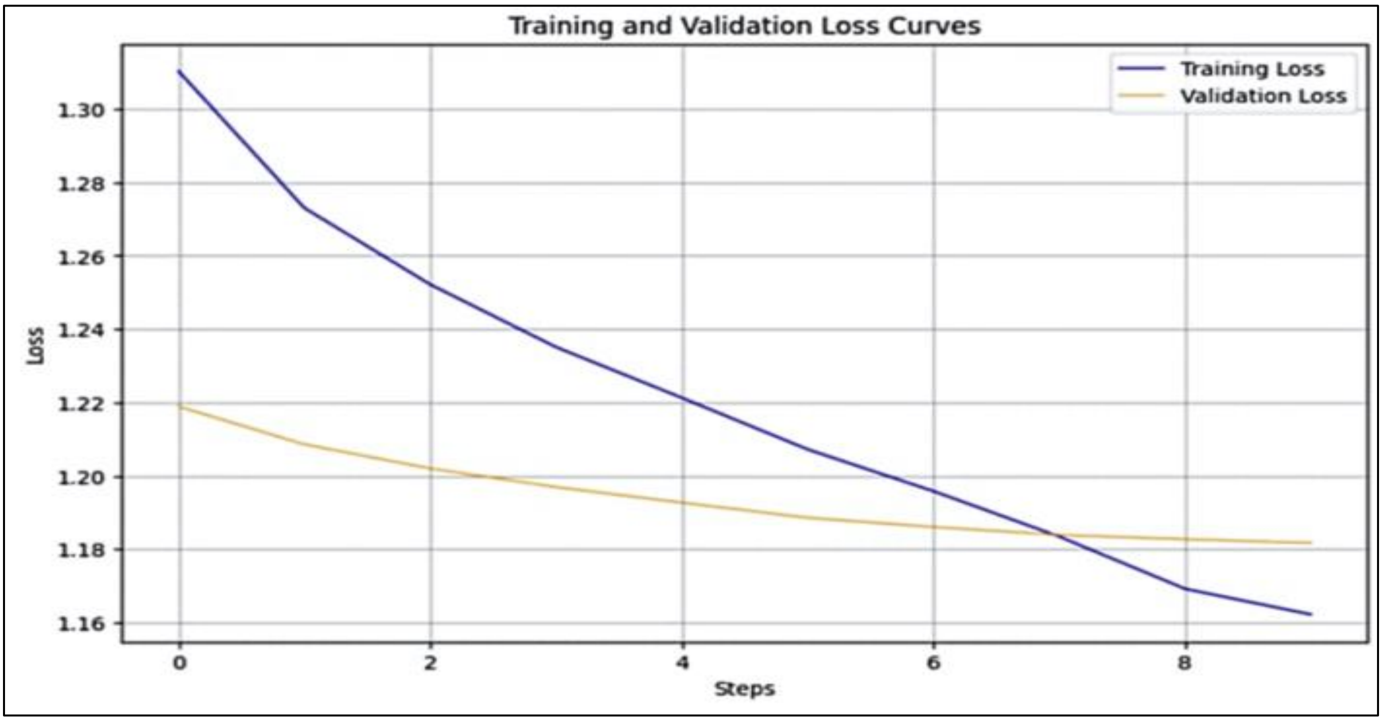

Fig 2 Training vs Validation Loss in T5 Base Mod

Fig 2 shows the training and validation loss for the T5-base model. While both losses decrease over time, the training loss drops more rapidly, indicating effective learning. The slower decline and consistent gap in validation loss suggest some overfitting and room for improved generalization.

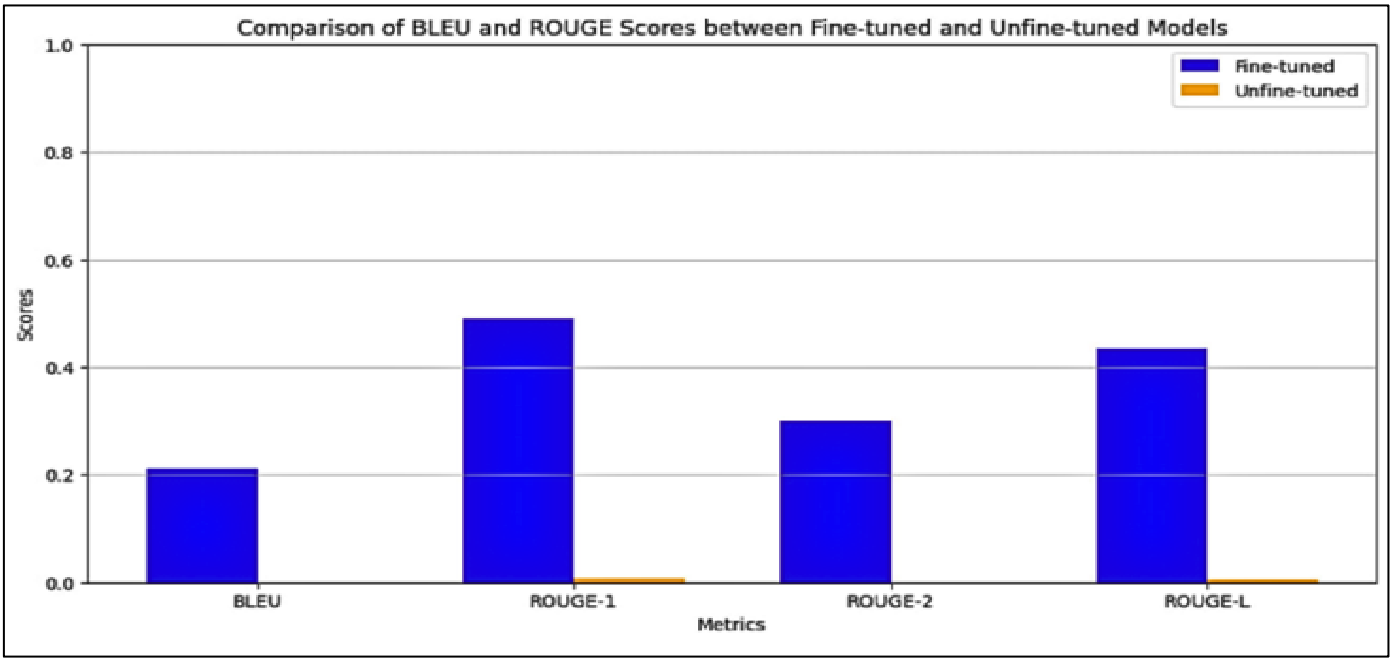

Fig 3 Comparison of BLEU and ROUGE Scores between Fine-Tuned and Unfine-Tuned Model in T5 Base Model

Fig 3 compares BLEU and ROUGE scores between fine-tuned and unfine-tuned T5-base models. The fine-tuned model significantly outperforms the baseline, achieving a BLEU score of 0.22 versus near-zero for the unfine-tuned version. ROUGE-1, ROUGE-2, and ROUGE-L scores also improve markedly to 0.48, 0.32, and 0.44, compared to 0.02, 0.005, and 0.015, respectively. These results highlight the impact of fine-tuning in enhancing text generation quality, coherence, and contextual relevance for MCQ generation.

➢ *Training and Evaluation of Flan-T5 Small Model:*

- *Model Training*

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun1878

For the Training specification of Flan-T5 small model, the parameter required were configured as mentioned below in Table 3

Table 3 Training Specification Table for Flan-T5 small model

S. N	Particulars	Applied Specifications
1	Number of Epochs	10
2	Batch Size	16
3	Optimizer	AdamW
4	Learning Rate	5e-10
5	Momentum	0.9
6	Decay	0.01

- *Evaluation:*

Fig 4 Training vs Validation loss in Flan-T5 small model

Fig 4 illustrates the behavior of training loss and validation Flan-T5 during the model training phase. The training loss begins higher than the validation loss (orange line) and decreases over time, though it exhibits fluctuations, indicating some instability in the training process. In contrast, the validation loss decreases smoothly, reflecting the model's ability to improve its generalization to unseen data. Throughout the process, the training loss remains consistently lower than the validation loss, signaling a degree of overfitting. The observed fluctuations in the training loss could be attributed to factors such as an inconsistent learning rate or noisy data. A smaller and more stable gap between the two losses would suggest better generalization performance.

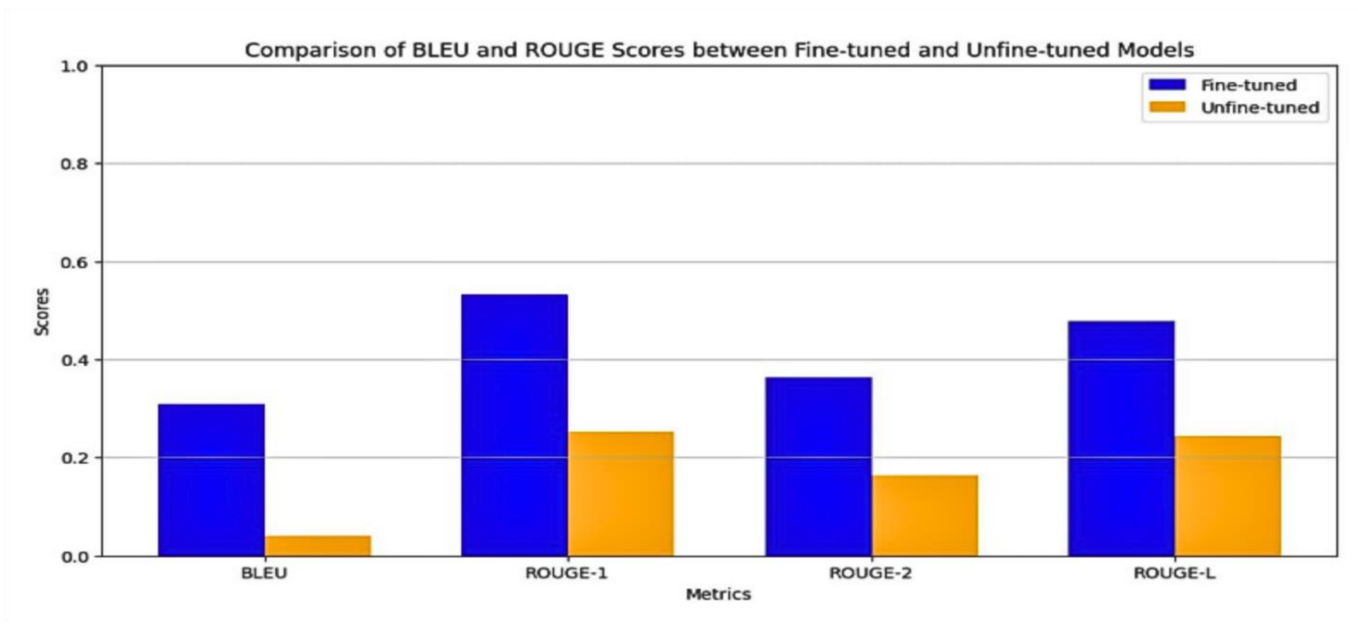

Fig 5 Comparison of BLEU and ROUGE Scores between Fine-Tuned and Unfine-Tuned Model in Flan-T5 Small Model

Fig 5 compares BLEU and ROUGE scores between fine-tuned and unfine-tuned Flan T5 Small models. The fine-tuned model achieves higher scores across all metrics, with a BLEU score of 0.28 versus 0.05 for the unfine-tuned version. ROUGE-1, ROUGE-2, and ROUGE-L scores also improve to 0.52, 0.35, and 0.47, compared to 0.26, 0.18, and 0.27, respectively. These results confirm that fine-tuning significantly improves text similarity, coherence, and contextual relevance in question generation.

➢ *Evaluation of LLM DistilBERT and LLM-T5 Distractor:*

- *LLM DistilBERT:*

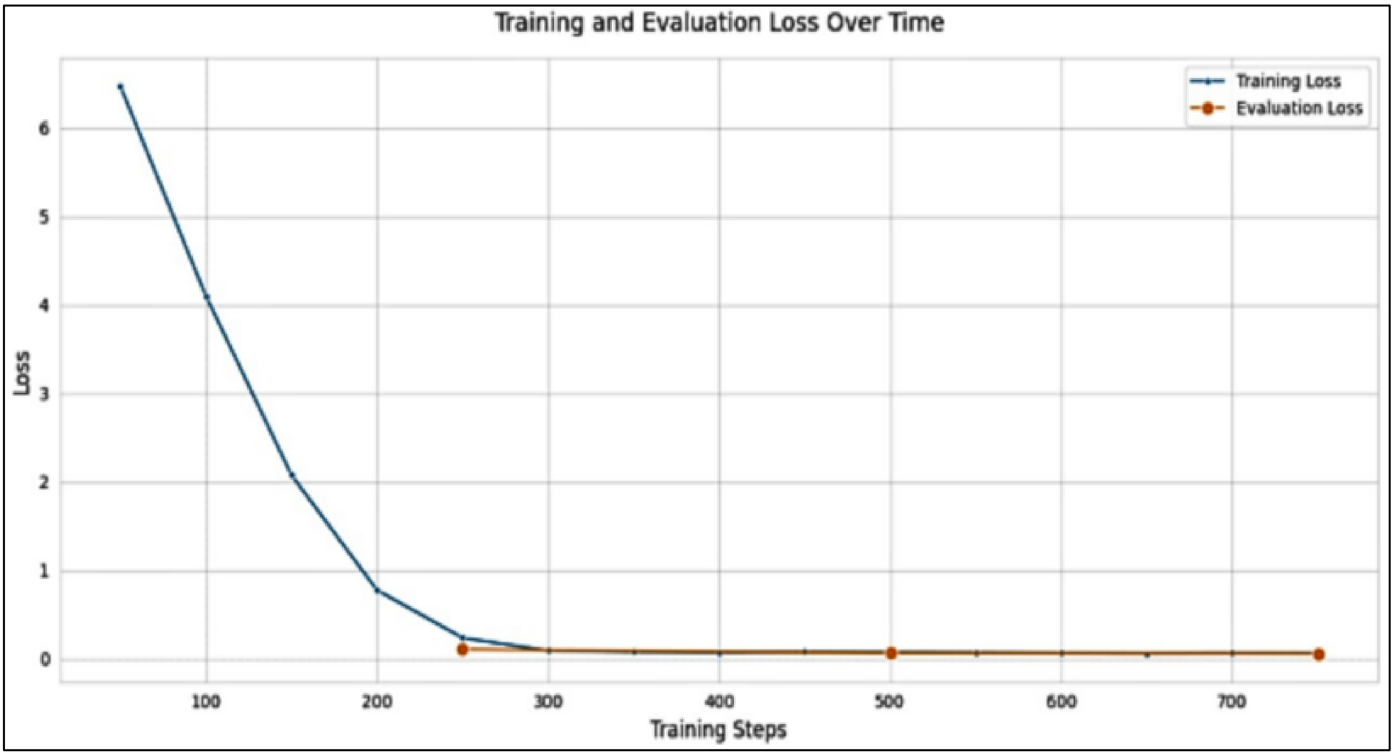

Fig 6 Training and Validation Loss in LLM DistilBERT

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun1878

Fig 6 shows a sharp decline in training loss, stabilizing near zero after step 300, while evaluation loss remains consistently low. The close alignment of both indicates effective learning with minimal overfitting and strong generalization.

- *LLM-T5 Distractor*

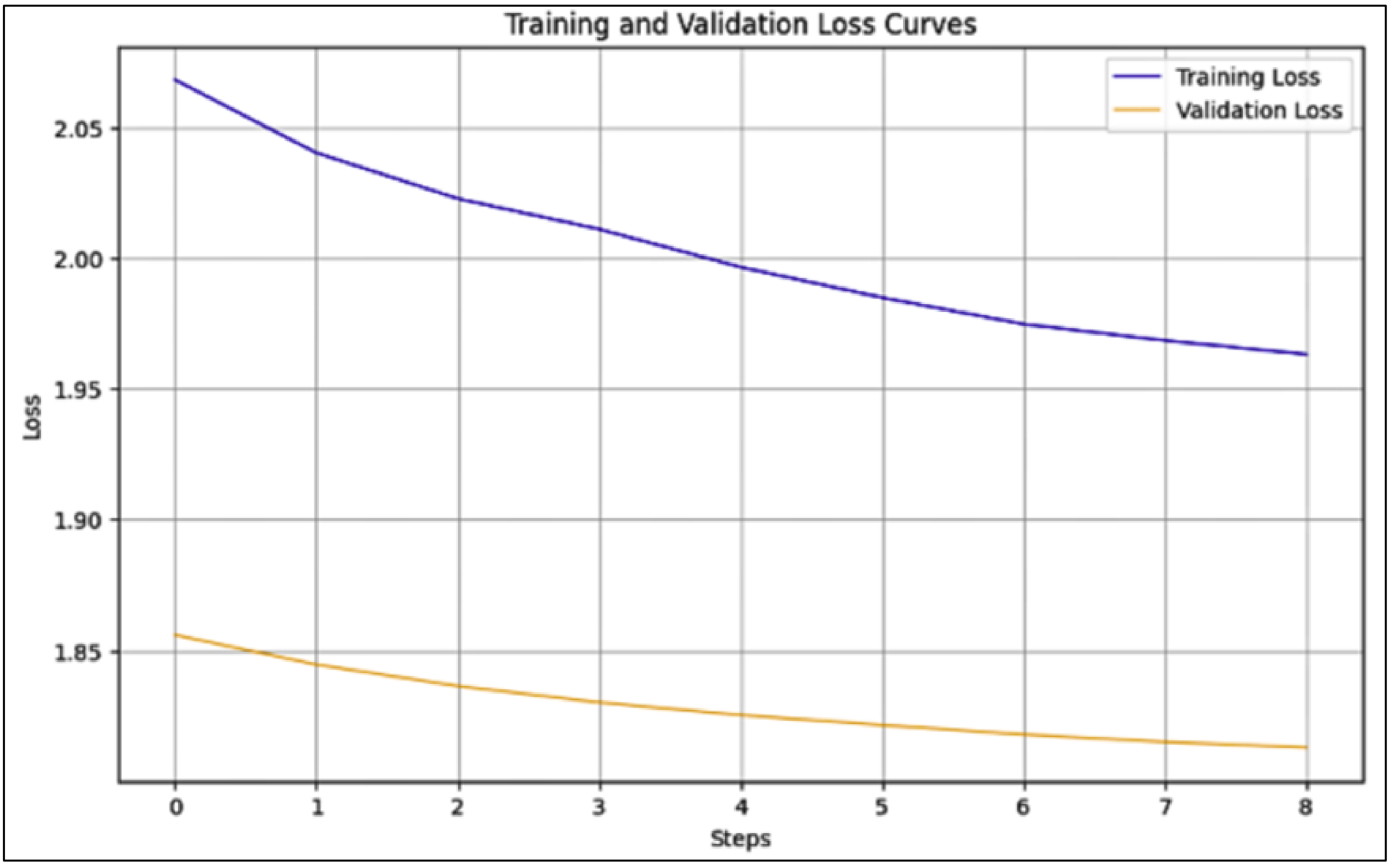

Fig 7 Training and Validation Loss in LLM-T5 Distractor Generator

Figure 7 shows the training and validation loss curves for the LLM-T5 distractor generator. The training loss drops sharply initially and stabilizes near zero by step 300, indicating efficient learning. The validation loss remains consistently low, closely tracking the training loss, which suggests strong generalization without overfitting. This demonstrates the model's robustness and effective training.

➢ *Final Output:*

The user interface of the webapp is shown in Figure 8 and Figure 9.

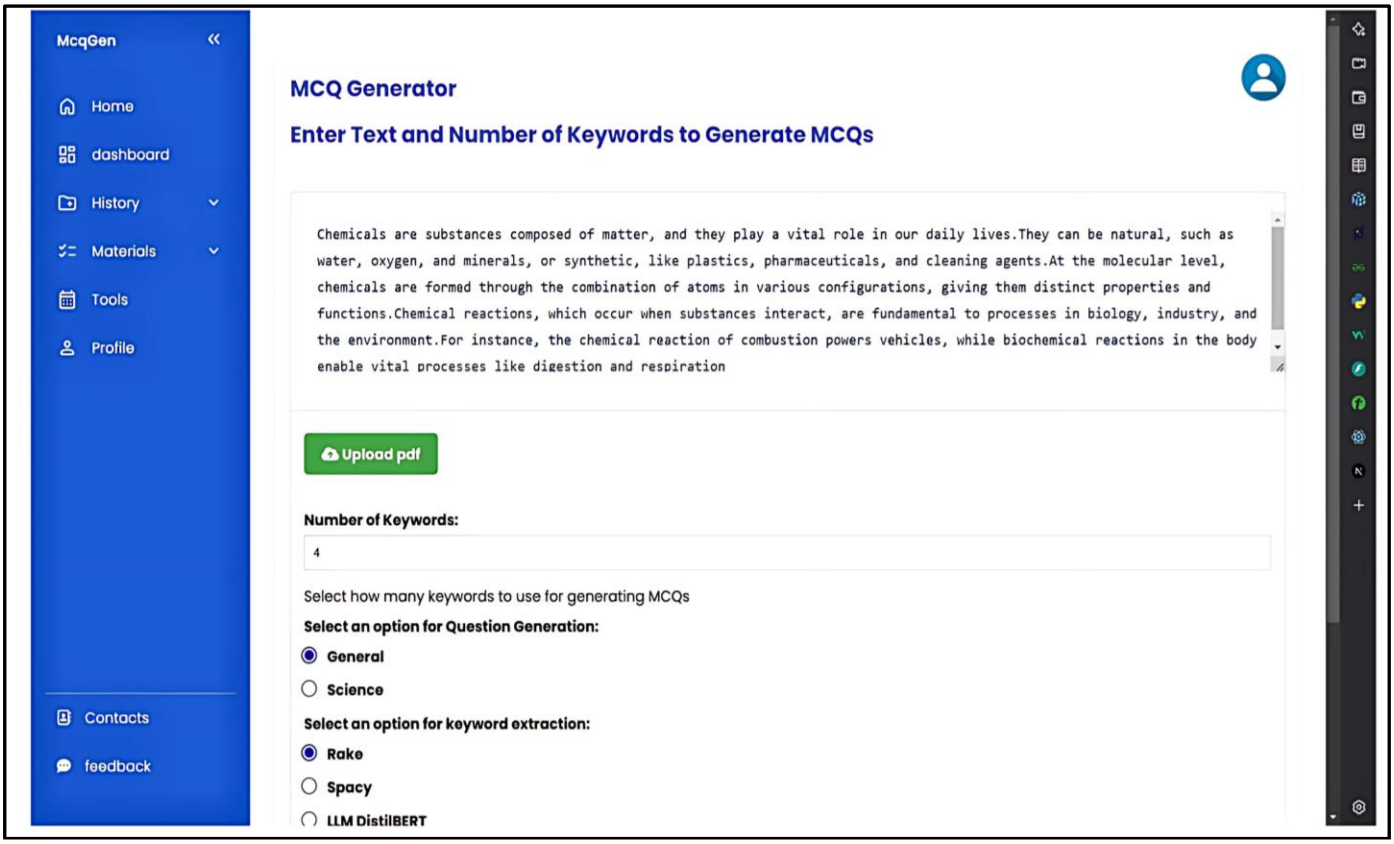

Fig 8 Text Input and Model Selection in System

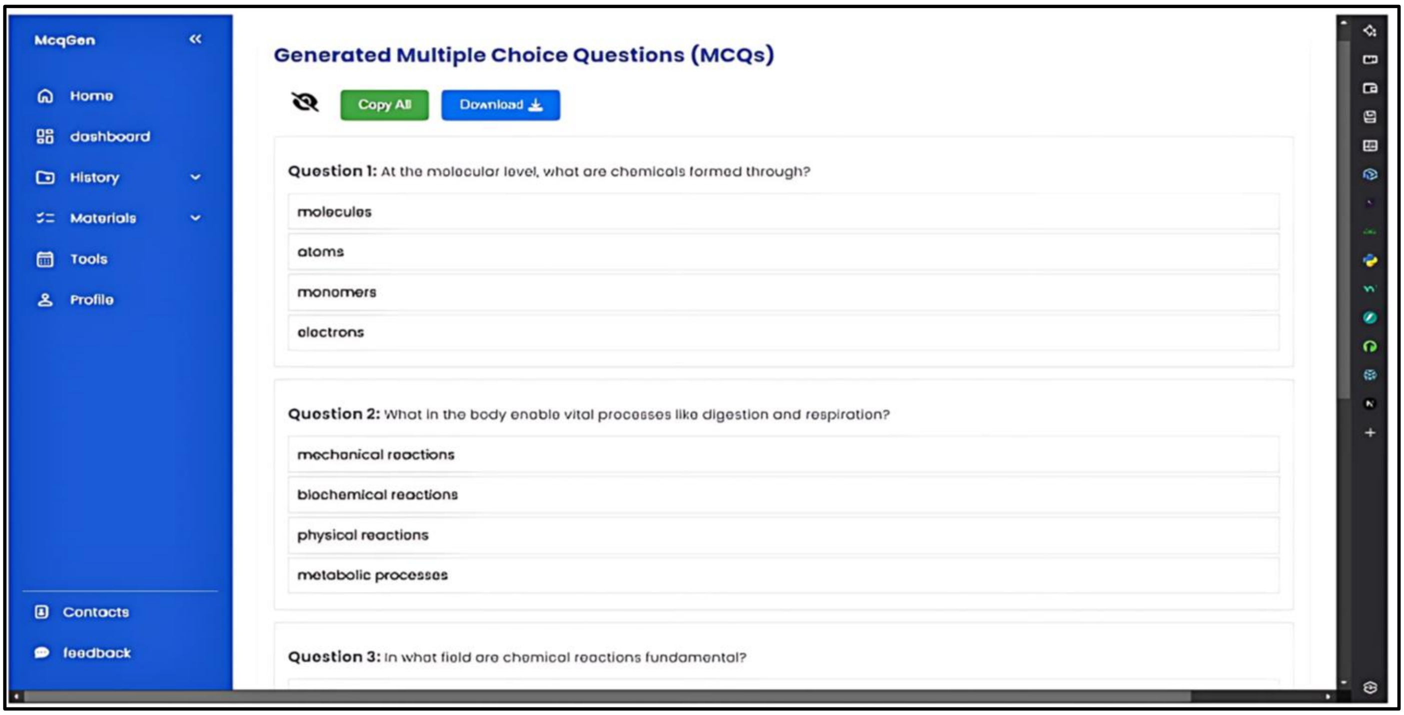

Fig 9 Generated MCQ

➢ *Limitations:*

- The system performs optimally in STEM subjects but shows limited effectiveness in humanities, where subjective interpretation and cultural context are critical.
- Performance is highly sensitive to input quality; noisy PDFs or unstructured text reduce the accuracy of question generation.
- The models are trained primarily on English-language datasets, limiting their applicability in non-English or multilingual educational contexts.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun1878

- Despite the use of LoRA and QLoRA for optimization, fine-tuning models for specialized domains such as law or medicine remains computationally intensive.

V. CONCLUSION

In conclusion this work addressed the challenge of automating MCQ generation, a traditionally time-consuming and expertise-driven task in education. By leveraging NLP techniques and fine-tuned transformer models, the system effectively processes input text or PDFs to generate contextually relevant questions and plausible distractors. Fine-tuning models like T5 and Flan-T5 on domain-specific datasets significantly improved question quality, while hybrid distractor generation using Sense2Vec, WordNet, and LLMs ensured relevance and variety. Evaluation metrics demonstrated clear performance gains, validating the effectiveness of the approach. The resulting Django-based application offers a flexible, scalable, and user-friendly solution for automated MCQ generation.

➢ *Future Enhancements:*

- Develop a lightweight mobile app that works offline, enabling students and teachers to generate and practice MCQs anytime, anywhere.
- Expand the project into an online exam platform with teacher-admin quizzes, student login, automatic grading, performance analytics, and adaptive question banks.
- Extend the system's capabilities by fine-tuning models on multilingual and humanities-focused datasets to improve performance in non-STEM domains that require nuanced understanding of language, culture, and subjectivity.

ACKNOWLEDGMENT

The authors would like to express sincere gratitude to Mr.Manoj Kumar Guragain, Head of the Electronics and Computer Engineering Department (IOE Purwanchal Campus) for his invaluable guidance and insightful feedback. The authors would also like to express gratitude towards the Electronics and Computer Engineering Department (IOE Purwanchal Campus), associated teachers, and all the well-wishers.

The authors take full responsibility for any errors or omissions that may have occurred in preparing this research paper and would like to welcome comments and queries.

REFERENCES

[1]. Y. Folajimi and O. Omojola, "Natural language processing techniques for automatic test questions generation using discourse connectives," *Journal of Computer Science and Its Application,* vol. 20, no. 2, pp. 60–76, 2013.

[2]. Nwafor and I. E. Onyenwe, "An automated multiple-choice question generation using natural language processing techniques," *arXiv preprint,* arXiv:2103.14757, 2021.

[3]. Y. Susanti, T. Tokunaga, H. Nishikawa, and H. Obari, "Automatic distractor generation for multiple-choice English vocabulary questions," *Research and Practice in Technology Enhanced Learning*, vol. 13, no. 1, p. 15, 2018

[4]. Das and M. Majumder, "Factual open cloze question generation for assessment of learner's knowledge," *International Journal of Educational Technology in Higher Education*, vol. 14, pp. 1–12, 2017.

[5]. K. Grover, K. Kaur, K. Tiwari, Rupali, and P. Kumar, "Deep learning based question generation using T5 transformer," in *Advanced Computing: 10th International Conference, IACC 2020, Panaji, Goa, India, December 5–6, 2020, Revised Selected Papers, Part I*, Springer, 2021, pp. 243–255.

[6]. H.-L. Chung, Y.-H. Chan, and Y.-C. Fan, "A BERT-based distractor generation scheme with multi-tasking and negative answer training strategies," *arXiv preprint*, arXiv:2010.05384, 2020.

[7]. H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Roziere, N. Goyal, E. Hambro, F. Azhar, and others, "LLaMA: Open and efficient foundation language models," *arXiv preprint*, arXiv:2302.13971, 2023.

[8]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[9]. V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," *arXiv preprint*, arXiv:1910.01108, 2019.

[10]. J. Oza and H. Yadav, "Enhancing question prediction with FLAN-T5: A context-aware language model approach," *Authorea Preprints*, 2023

[11]. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020

[12]. N. C. P. Prakash, A. P. Narasimhaiah, J. B. Nagaraj, P. K. Pareek, N. B. Maruthikumar, and R. I. Manjunath, "Implementation of NLP based automatic text summarization using spaCy," *International Journal of Health Sciences*, vol. 6, no. S5, pp. 7508–7521, 2022

[13]. S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," in *Text Mining: Applications and Theory*, pp. 1–20, 2010.