MCQ Generator

V. Roshan Kumar[1], Chinta DivyaSurathi[2], Galla Bharghav Ram[3], Kola Govardhan[4], Kakumanu Amani [5]

1Assistant Professor, Department of CSE(AIML)Bapatla Engineering College, Bapatla 522101, AP, India 2,3,4,5 Student, Department of CSE(AIML)Bapatla Engineering College, Bapatla 522101, AP, India

roshan4linus4550@gmail.com, divyasurathi2704@gmail.com, bharghavramgalla@gmail.com,govardhankola95@gmail.com, kakumanumani@gmail.com

Abstract—The online learning resources made it possible to effectively enhance and showcase these resources; thus, making automated assessment generation even more fundamental in contemporary era education. In this study, we introduce a smart question generation system powered by LLMs to autogenerate different styles of assessment questions from educational material. The proposed system enables users to upload documents or provide website URLs from which MCQs, fill-in-the-blank questions, true-false questions, and short answer queries can be created. With an extensible architecture, the framework combines several LLM providers allowing users to use their own API keys for higher flexibility and performance. Using sophisticated prompt engineering techniques, the models are directed toward generating structured and localized questions. The system covers preprocessing and text extraction modules which use unstructured educational material as inputs to produce appropriate inputs for question generation. A web-based interface designed using Python and Flask allows teachers and students to intuitively interact with the system. The proposed system thus increases adaptability to different educational contexts by supporting diverse types of queries and smoothly integrating various models at run-time. Experimental observations show that the system generates answers that are logical and contextually appropriate at a tiny fraction of the time and effort required to generate questions manually.

Keywords—*Prompt engineering, educational technology, web content extraction, multiple choice questions, automated question generation, large language models, personalised learning.*

1. INTRODUCTION

Modern education has changed a lot in these days with the growth of digital learning tools. Online classes, learning websites, and virtual classrooms have made it very easier for students to learn from various places and at their own pace. These technologies also help teachers reach many students at the same time. Even with gradual changes in AI technology, multiple-choice questions are still widely used in exams, practice tests, assessments, and mock tests because they are easy to answer and simple to check. They also allow teachers to test many topics and modules in a short period of time. Nevertheless, it is not easy to prepare quality questions. Teachers need too much time to read materials, think of relevant questions, and prepare solutions manually.

Over the years, it had been easier for computers to read most plain-text formats. This led to making new tools that help ease the process of generating study questions automatically. These tools are able to read a passage and generate questions based on the major ideas in that passage. It can save teachers' time in designing assessments and also help students in solving more questions during their preparation phase. But a lot of the existing tools are still limited in certain ways. Some models only work with certain input forms; others have obstacles to being used on different categories of study materials.

In this work, we propose an intelligent Question Generator powered with AI to automatically generate a wide range of assessment questions. It allows teachers to rapidly generate multiple-choice questions, true or false questions, fill-in-the-blank and short-answer tests. The framework allows the system to autonomously obtain information from online sources, given a diverse mix of input formats (uploaded files [PDF or DOCX], text and website URLs). The system also uses a customizable architecture where users can configure their own API keys for different types of language models, ensuring better performance with flexibility.

2. LITERATURE SURVEY

Automated question generation systems have been developed in line with the fast-paced improvements of natural language processing and artificial intelligence. Multiple-choice questions (MCQs) are a significant tool for learner understanding assessment and improving adaptive learning systems. MCQs are used to evaluate student performance in many educational systems, which helps enhance personalized learning experiences [1].

Most of the early works on automated question generation were primarily based on template-driven and rule-based methodologies. Such systems produced questions based on pre-defined linguistic patterns and handcrafted templates. While these methods generated structured questions, they lacked adaptability and frequently failed to create context-relevant questions from intricate educational content [2]. Moreover, creating templates for rule-based systems was a tedious process requiring a large amount of effort and they could not maintain good performance when handling different types of text content [3].

As ML techniques advanced, more complex solutions for automatic question generation started to appear. Neural network-based models, in particular sequence-to-sequence architectures, showed better performance in generating questions from textual material. These models could learn the semantic relationship between words in a text, allowing them to generate more natural and contextually relevant questions than traditional rule-based systems [4]. In recent times, the development of LLMs has greatly enhanced automatic question generation. LLMs apply sophisticated NLP techniques to examine extensive amounts of text data and produce coherent educational material [5]. Smart question generation systems are also being integrated into digital learning platforms [6]. Most existing solutions only generate one type of question like MCQs [7] and support limited input formats.

In response to these challenges, the system presented here offers a flexible framework for generating assessment questions of various types. This system generates multiple choice, true/false, fill-in-the-blank and short-answer questions. Moreover, it works with multiple input sources (PDF, TXT, text input, website URL) by automatically extracting the educational material directly from those sources.

3. METHODOLOGY

Using large language models (LLMs), the suggested system presents an intelligent framework for automatically generating questions. The architecture is created to automatically generate various kinds of assessment questions from many input sources and to process educational content. Top-quality questions are created by the system via web content extraction, document processing, prompt engineering, and LLM-based natural language generation.

A. System Overview

Several connected modules make up the general design of the MCQ Generator system; they help to turn unprocessed instructional material into organised assessment questions. Users start with user input, offering content in many forms including text, PDF files, or website URLs. The system extracts significant textual material from the input, then generates questions depending on the extracted content using large language models. Emphasizing flexibility and scalability, the design lets teachers and pupils quickly produce various kinds of queries without laborious work.

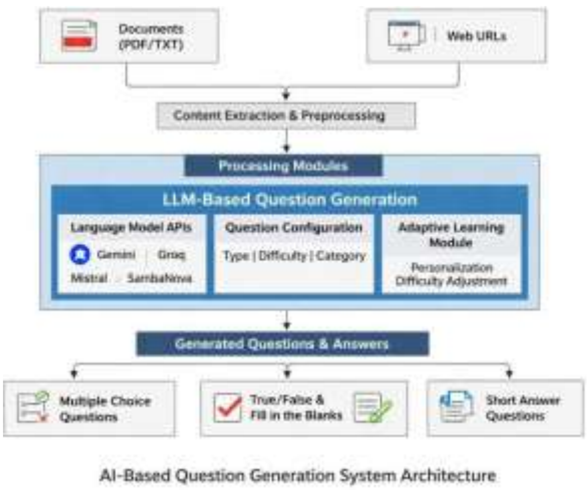

Fig. 1. AI-Based Question Generation System Architecture

B. Input Processing Module

The input processing module deals with different types of content inputted by the users. Document files including PDF and DOCX are allowed to be uploaded, and textual contents of these files are extracted by document parsing techniques applied by the system. The system also accepts textual content pasted directly to the application interface. Additionally, the system can extract information from website URLs. Once the user uploads a link, the system obtains the information on the webpage and analyses the appropriate textual data, allowing it to generate questions automatically from online learning material, blogs or educational articles.

C. Large Language Model Integration

The essence of the system is the combination of large language models for natural language processing and question generation. The system uses various LLM providers including Gemini, Groq, Mistral, and SambaNova to enhance reliability and performance. These models can interpret the text we feed them, give it a semantic meaning and come up with proper questions with regard to the context. The system employs an LLM pool that contains several model services which enable the application to select a model that is available at a given time.

D. Prompt Engineering Strategy

Prompt engineering is an important aspect in quality of questions created. Prompts steer the language model to generate structured educational questions that are accurate to the context. The prompts give the model a definite guideline on the number of questions to be asked, the type of question, the level of difficulty and the way the question should be answered. The prompt form also ensures that every question produced has the correct answer as well as various answer choices when required.

E. Question Generation Module

The question generation module converts the textual content that has been processed into different forms of assessment questions. The system supports multiple choice questions (MCQs), true or false questions, fill-in-the-blank questions and short answer questions. The purpose of each type of question is different, and the instructor can assess different areas of students' knowledge. The language model examines the context of the given text and produces questions which articulate the important concepts, definitions and associations found in the text.

F. Framework of System Implementation

A web-based architecture has been used to implement the system to make it accessible and easy to use. The system backend is built on a lightweight web framework (Flask) which receives and processes user requests, as well as communicates the input data with the language models. The application interface allows the user to upload files, type text or place web addresses. The suggested methodology combines the latest natural language processing tools with a versatile system design that generates an effective and useful approach to automated generation of questions in education.

4. ALGORITHM

The suggested system utilizes a designed algorithm which combines text extraction, preprocessing, prompt engineering, and large language model (LLM) based question generation. The algorithm converts unstructured textual material to structured educational assessment questions and supports different sources of inputs while generating numerous types of questions on a contextual basis.

A. Mathematical Foundation

Take the input educational data to be a textual dataset obtained in the form of documents, URLs or user-supplied text. Let $T = \{t_1, t_2, t_3 \dots t_n\}$ represent the collection of textual sentences deduced from the input source. The preprocessing module transforms the raw text into formatted parts: $P(T) \rightarrow S = \{s_1, s_2, s_3 \dots s_k\}$ where S is meaningful semantic pieces of original text. The mathematical formulation of the question generation process is: $Q = G(S, P_e, M)$ where Q is the set of questions generated, G is the generation function, S is the semantic text segments after processing, P_e is the prompt engineering approach, and M denotes the selected large language model. The question set at the output can be formulated as: $Q = \{Q^{McT}, Q_{te}, Q^{fb}, Q_{sa}\}$ where $Q^{McT} \rightarrow$ Multiple Choice Questions, $Q_{te} \rightarrow$ True/False Questions, $Q^{fb} \rightarrow$ Fill in the Blank Questions, $Q_{sa} \rightarrow$ Short Answer Questions.

B. Question Generation Algorithm

The LLM-based automated Question Generation algorithm is as follows. Input: Document, text or URL educational content. Output: Generated questions (MCQ, True/False, Fill-in-the-Blank, Short Answer). Step 1: Receive the input (document upload, text input or web address). Step 2: Pull text data from the input source. Step 3: Undergo preprocessing including text cleaning, sentence breaking and elimination of irrelevant information. Step 4: Accommodate meaningful ideas and semantic units of the extracted text. Step 5: Code prompts using prompt engineering. Step 6: Send the prompt to the selected large language model from the available LLM pool. Step 7: Generate questions with the LLM. Step 8: Code the resulting output into structured question categories. Step 9: Present generated questions to the user. Step 10: End process.

C. Hyperparameters

Key parameters include: Temperature (randomness in the generated responses; a moderate temperature provides moderation in creativity and factual accuracy), Max Tokens (specifies the length of the output generated), Top-p Sampling (restricts the choice of tokens according to probability distribution), and Frequency Penalty (reduces the amount of repetition of words or phrases in generated questions). These parameters are applied to providing high-quality and uniform question generation.

D. Performance Metrics

To measure the quality and effectiveness of the generated questions, a number of performance measures are considered: Relevance (calculates the relevance of the formulated questions with the input material), Accuracy (concludes on whether the responses generated are correct), Diversity (ensures that the system produces various types of questions that touch on various concepts), and Readability (assesses the query intelligibility and grammatical correctness of the generated queries).

E. Comparison with Traditional Methods

Conventional question generation methods usually use rule-based systems or templates that are hand designed. Such approaches often involve a lot of work to establish linguistic rules and are often inflexible for processing a wide range of educational material. On the other hand, the proposed system gives the contextual knowledge of large language models. The linguistic rules defined manually are not needed, and complex textual information can be analyzed and contextually relevant questions are generated by the LLM. Also, conventional systems can only respond to predetermined question formats, but the framework suggested is able to respond to various questions like MCQs, True/False, Fill-in-the-Blanks and Short Answer questions.

5. IMPLEMENTATION AND EXPERIMENTAL RESULTS

This section provides a report on how the proposed automated question generation system is practically adopted and how the system performs in generating educational assessment questions. The system is an integration of large language models, prompt engineering techniques, and a web-based application to provide a powerful platform to produce an array of question types concerning educational material.

A. System Implementation

The proposed system was created using the Python programming language and modern libraries and frameworks. The system architecture consists of four notable components: processing of inputs, text retrieval, posing of questions founded on the large language model, and web-based user interface. The implementation allows the user to enter data in three ways: uploading of documents, typing a piece of text or a web address. It was developed in Flask in order to provide the users with an interactive interface.

B. Development Environment

The system was created and implemented using the Python programming language, which has vast data processing and machine learning libraries. The development tools were Visual Studio Code for coding and debugging, and Python virtual environments for maintaining dependencies. The web-based interface has been developed with the help of Flask, which allows seamless interplay between the front end and the question generating system of the back end.

C. Input Processing and Text Extraction

The system is made to be more flexible and usable by supporting different forms of input. Users are able to post educative content in the form of a document or provide the URL of educational websites. These sources are mined in order to extract the textual information and question generation is done using the system. When a document has been uploaded, the system parses the file and extracts the textual information using the right parsing methods. In the case of webpage input, the system extracts the content of the webpage and removes unneeded information like navigation menus or advertisements.

D. Experimental Results

The outcomes of the experiment demonstrate that the proposed system can generate quality educational questions on the basis of various sources of input. The questions that were created were evaluated on a small number of criteria including contextual relevancy, grammatical validity and educational value. The system was able to generate meaningful MCQs whose options and distractors were correct, and True/False and Fill-in-the-Blank questions were correct. Short Answer questions were generated as well, which stimulated further inquiry about the material. The fact that multiple language models were applied led to improvement in the system as the quality of the output was always the same regardless of the complexity of the input material.

6. RESULTS AND DISCUSSION

A. System Configuration Interface

The system consists of an administration configurable interface that integrates several different LLM providers. This is a module through which users can set model parameters and API keys for different providers including Gemini, Groq, Mistral, and SambaNova. Such a flexible configuration enhances the system reliability as it allows dynamically choosing between available models for question generation.

Fig. 2. System Configuration Interface showing LLM provider settings.

B. Student Question Generator Interface

The input can be study material or links to the website provided by users through the student generator module. It extracts the text from the input and preprocesses it for automatic question generation. Generators of tests, quizzes, and answer sheets are customized to the users' input. It is easy to use and students can derive the practice questions on their own for self-learning and preparation for exams.

Fig. 3. Student Question Generator Interface.

C. Teacher Portal Interface

Educational materials can be uploaded through the teacher portal, and educators then have the option to generate structured assessments. Teachers can enter a range of topics, types of questions and even difficulty levels to generate targeted question sets for quizzes or tests. The module significantly cuts the time taken to prepare academic assessments while assuring that questions are still applicable to the study material given.

Fig. 4. Teacher Portal Interface.

D. Question Configuration Module

The question configuration module allows educators to configure question parameters including topics, types of questions and difficulty levels to generate targeted question sets for quizzes or tests. The module significantly cuts the time taken to prepare academic assessments while assuring that questions are still applicable to the study material given.

Fig. 5. Question Configuration Module.

E. Generated Question Output

The generative part of the answer was produced based on a prompt-based generation method. Processing the structured content and context allows the system to generate well-structured questions. Explicitly generated questions such as statements, options in MCQs and answers where possible. The results show that the system can generate relevant, meaningful questions based on both documents and web content.

Fig. 6. Generated Question Output showing MCQs with answer options.

F. Discussion

The experiments demonstrate that the approach automates educational question generation effectively. The system also shows flexibility in the types of input formats and questions that it supports. The proposed system serves as a solution to these issues, generating various question formats automatically and saving time for educational institutions or independent teachers, as well as students who need to pass a specific evaluation. The findings indicate that the combination of large language models and structured prompt engineering is a good solution to automated question generation. The system has exhibited high capability of providing a variety of questions from document-based and web-based learning materials.

7. FUTURE SCOPE

Some improvements will feature extra enhancements to the automatic question generation system, such as converting images such as scanned documents or diagrams into text using optical character recognition (OCR) techniques. This would enable the system to build questions from a broader class of learning materials. Another enhancement is a mobile app that lets students and teachers create practice questions on the go. By integrating such components, the system can be augmented with adaptive learning algorithms that adjust question difficulty based on student performance to enable personalized learning. Subsequent versions may also offer support for many regional languages to use it in various educational strands. This will enable the system to be scaled and effectively utilized with current digital education platforms.

8. CONCLUSION

In this project, an AI-based Automatic Question Generation System is proposed by integrating document processing, web content extraction and large language models as a complete intelligent system for generating questions to assess students educationally. It lets users feed in materials (PDF/TXT) and text or web URLs, which it analyzes to formulate a range of questions. Our platform has been designed to handle different types of questions like MCQ, Fill in the Blanks, True/False and Short Answer with customizable features like difficulty levels (Easy, Medium, Hard) and categories — Contextual, Coding and Mathematics. With the combination of technology and creativity, both students and educators are able to design a wide range of assessments to meet all learning styles. The system saves users time and effort by automating the manual creation process of these questions for an optimized deployment of assessment sheets. Plugging multiple large language model providers also increases system fault tolerance and better question diversity. This project showcases the application of artificial intelligence, particularly in the form of modern language models, to enhance the capabilities of digital learning environments by providing an efficient way to generate educational assessments like quizzes and exams.

REFERENCES

[1] Y. Kurdi, J. Leo, and C. Parsia, "A systematic review of automatic question generation for educational purposes," Int. J. Artif. Intell. Educ., vol. 30, no. 1, pp. 121–204, 2020.

[2] M. Heilman and N. A. Smith, "Good question! Statistical ranking for question generation," in Proc. NAACL HLT Workshop Question Generation, Los Angeles, USA, 2010, pp. 609–617.

[3] S. Mitkov and L. A. Ha, "Computer-aided generation of multiple-choice tests," Natural Language Eng., vol. 9, no. 1, pp. 37–54, 2003.

[4] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL), Vancouver, Canada, 2017, pp. 1342–1352.

[5] T. Brown et al., "Language models are few-shot learners," in Advances Neural Inf. Process. Syst. (NeurIPS), vol. 33, 2020.

[6] A. Alsubait, B. Parsia, and U. Sattler, "Automatic generation of multiple choice questions from ontologies," in Proc. 7th Int. Conf. Knowledge Capture (K-CAP), Banff, Canada, 2013, pp. 135–142.

[7] S. P. Singh, A. Kumar, and R. Singh, "Automatic question generation using NLP techniques," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 6, pp. 52–58, 2019.

[8] M. Rus, S. Cai, and A. Graesser, "Question Generation: Example of a Multi-Year Evaluation Campaign," Proc. Workshop Question Generation Shared Task Evaluation Challenge, 2010.

[9] N. Mostow and W. Chen, "Generating Instructional Questions Automatically for Reading Practice," Proc. Int. Conf. Intelligent Tutoring Systems, 2009.

[10] A. Yao, J. Wan, and X. Xiao, "Automatic Question Generation from Text Using Neural Networks," Proc. Int. Conf. Natural Language Processing, 2018.

[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL-HLT, 2019.

[12] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020.

[13] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692, 2019.

[14] A. Radford et al., "Improving Language Understanding by Generative Pre-Training," OpenAI Technical Report, 2018.