International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Test Generation Application

Anudeep George, Shivani Chandra, Dr. Gousiya Begum (Asst Prof), Ms. K Sunitha (Asst Prof)

Department of CSE, MGIT (A), Gandipet, Hyderabad, 500075, Telangana, India.
Email:ageorge_cse210568@mgit.ac.in; cshivani_cse210573@mgit.ac.in

ABSTRACT

The project is about a Test Questions Generator Application utilizing Natural Language Processing (NLP) particularly using a Transformer based model, to automate the question generation of arguably diverse and contextual test question. The application takes text input, for instance, notes or text book chapters uploaded by the user, and generates test questions in a variety of forms, e.g., multiple-choice questions, short-answer questions, essay questions.

The application locates important concepts in the input using a transformer model and then generates questions reflecting the complexity of the concepts. We employed reinforcement learning to tune the algorithm, so the model can generate questions that align with the desired learning outcomes. While automating the process of generating test questions lighten the burden of educators, it should elevate the quality of assessments.

Taking an advanced NLP approach, this project offers a scalable and more intelligent approach to the construction of test questions. We also advance prior work on question generation tools, in respect to adaptability, variability and question worthiness while providing a potentially rich resource for the demands of 21st century education.

Keywords: Test Questions Generator, Natural Language Processing (NLP), Transformer-based Models, Automated Question Generation, Input Text Processing, Key Concept Extraction, Contextual Understanding

1. Introduction

The increasing use of Artificial Intelligence (AI) to improve educational resources has made traditional education methodologies much more dynamic and personal able for students. Among these was the development of an automated system to generate the educational resource area of generating test questions, which is one of the most controversial areas of research. These systems sought to alleviate the time consuming and complex task of preparing questions aligned to educational goals and addressing individualized learning strategies. The Test Question Generator Application made this task a reality through modern Natural Language Processing (NLP) techniques. The approach taken in this project uses state-of-the-art NLP methods using Transformer based techniques and Reinforcement Learning as a method to automate and develop a variety of valid test questions, short to long-answer questions developed from input text such as lecture notes or textbook chapters.

The goal of the application is help educators overcome the restrictions of time and the demand for effective and engaging assessments. The system produces many types of questions, such as multiple choice, short-answer, essay, true/false, and fill-in-the-blank, by leveraging Transformer architectures like GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers). Transformer models are well-known for their capacity to understand and generate human text, having learned about the relationships that exist in the context of the words in a sentence, as illustrated in the paper "Attention is All You Need" [1].

There is a reason for significance to this project, due to a number of developments in recent literature regarding AI for educational purposes. For example, one paper illustrated the need for targeted, domain-specific chatbots for educational purposes, but noted obstacles to obtaining accuracy and context, especially regarding esoteric or complicated topics [2]; a parallel line of reasoning was presented in a second research paper dealing with the development of an AI chatbot system that predicted infectious disease, and why it mattered to use NLP technology in order to better identify, acquire, and process requisite contextual information that was provided in the user input [3]. In both projects,(2) and (3) discussed the importance of NLP and AI as emerging technologies that produce non-manual systems, meaning, systems with less effort, but sufficient accuracy and content relevance.

Another significant contribution of this type of system is its focus on extracting key concept descriptions and their associate difficulty levels in both the preprocessing and generation steps. The editors of one paper presented the notion that robust preprocessing is significant in generating pertinent responses, particularly in context generating multilingual healthcare chatbot systems [5]. This has been applied in the proposed system and we ensure not only reliable questions roughly produced by functionally accurate, but also that the questions align developmentally with the cognitive skills of the learners it is intended for.

While many existing works have developed NLP-based chatbots and question generation systems, the majority of these come with limitations of scalability, adaptability to different types of input texts and lack reinforcements to improve performance. For example, one system that was discussed pointed out the difficulties they experienced in making accurate recommendations, but did not utilize adaptive learning techniques [6]. In this project, we addressed these gaps by leveraging Transform-based NLP algorithms in conjunction with RL, and produced a learner-generated scalable and adaptable solution for educators.

Problem Statement

This project seeks to remedy the problem of the quantity of time and effort it takes for educators to purposefully create quality, contextually relevant test questions for assessments. Traditional methods to create assessments typically re-apply analogue and/or manual processes ad nauseum, which results in unfortunate inefficiency as well as the potential for human error. Moreover, while automated question generation tools exist, they usually simply produce batches of generic, superficial questions which are neither focused on investigating specific objectives nor complex in their portrayal of the material the learner is acquiring.

The lack of variation in questions and adaptability in the types of questions presented to students could result in assessments that do not accurately assess the learner's understanding of materials presented or capacity for critical thought. Furthermore, variation in the quality of questions that can be produced negates any potential benefits to the integrity of the assessment.

Existing System

Currently, automated test item generation systems rely on basic algorithms that are based on keyword matching or algorithm-based. Because of this limitation, almost all automated items are very basic, shallow, narrow, and repetitive questions with little contextual relevance or value in education. Most of the systems only can produce a limited number of question types (most often multiple choice or fill-in-the-blank), and are not capable of producing more open-ended or complex questions.

In addition, many restrict flexibility regarding content area or level of education, without much of an option for customization and can yield questions of variability of quality. Because of these issues, the assessments generated by these systems may not provide a full assessment of student understanding and applications of higher order thinking in a generalizable way, while allowing for amendments by the instructor which limits the potential benefit from using an automated assessment process.

Proposed System

The developed system is the AI-enabled Test Question Generator that utilizes Natural Language Processing (NLP) based techniques that utilize Transformers models to automate the creation of test questions that are fresh, unique, context-relevant, and high-quality. The proposed system uses machine learning, and reinforcement learning to overcome many limitations of existing systems and provide industry scale and quality solutions for educators to create question papers.

How It Improves on Existing Solutions:

- **Enhanced Contextual Understanding:** Adopts advanced NLP techniques to locate key ideas to produce contextual questions relevant to the educational product matter.
- **Variety of Question Types:** Accepts all question types (multiple-choice, short-answer, essay, true/false, and fill-in-the-blank) to ensure a comprehensive assessment evaluation.
- **Improved Question Quality:** Supports a reinforcement learning process to improve and generate questions to assist educators in ensuring the questions are pedagogically informed, contextually relevant, and the correct level of increased difficulty.
- **Reliable, Consistent Output:** Output and quality of questions generated has a level of consistency that provides a reduction to the education workload and time spent on manual reviews.

2. Literature Review

[1] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Amodei, D. (2020) – "Language Models are Few-Shot Learners":

This study introduced GPT-3, a state-of-the-art Transformer-based language model capable of performing various NLP tasks, including text summarization, translation, and question generation. The research demonstrates the potential of such models to generate diverse and contextually relevant test questions with minimal fine-tuning.

[2] Zhou, Q., Yang, N., Wei, F., Huang, S., & Zhou, M. (2019) – "Neural Question Generation from Text: A Survey"

This paper provides a comprehensive overview of neural question generation techniques, exploring sequence-to-sequence models, attention mechanisms, and reinforcement learning strategies, emphasizing their effectiveness in generating coherent and contextually accurate questions from textual content.

[3] Wang, D., Qiu, X., & Huang, X. (2018) – "Learning to Ask: Neural Question Generation for Reading Comprehension"

This research focused on generating questions specifically for reading comprehension tasks, proposing a neural network-based approach that leverages contextual embeddings and attention mechanisms to produce high-quality questions aligned with the content.

[4] Du, X., Shao, J., & Cardie, C. (2017) – "Learning to Ask: Neural Question Generation with Recurrent Neural Networks"

This study developed one of the early models for automated question generation using recurrent neural networks (RNNs). Though limited to simpler question types, it laid the groundwork for later advancements using Transformer-based models for better context understanding and question diversity.

[5] Yuan, X., Liu, Y., Wu, T., & Meng, X. (2020) – "Reinforcement Learning for Question Generation: Improving Question Quality and Diversity"

This paper introduced reinforcement learning techniques to enhance the quality and diversity of automatically generated questions, using reward signals based on relevance, grammatical accuracy, and cognitive complexity.

[6] Kumar, S., Malakar, S., & Roy, B. (2021) – "Automatic Generation of Multiple-Choice Questions from Text Using Deep Learning"

This work focuses on generating multiple-choice questions (MCQs) using deep learning techniques, emphasizing the importance of selecting distractors that are challenging and relevant to the main question, thereby making assessments more effective.

[7] Sharma, R., Yadav, P., & Singh, M. (2020) – "A Review of Question Generation Techniques for Education"

This review categorizes question generation methods into rule-based, template-based, and neural network-based approaches, highlighting the limitations of traditional systems and discussing how modern NLP techniques, such as Transformers and reinforcement learning, address these challenges.

[8] Reddy, P., Srinivasan, K., & Narayan, R. (2019) – "Adaptive Assessment Systems Using NLP"

This study explores adaptive assessment systems that use NLP to dynamically generate questions based on students' performance and learning progress. The system personalizes assessments by adjusting question difficulty and content in real-time, improving learning outcomes.

[9] Chaudhary, A., Patel, R., & Mehta, S. (2022) – "Context-Aware Question Generation for Educational Applications"

This paper presents a context-aware question generation system that uses semantic analysis and knowledge graphs to generate questions that are highly relevant to the input text, improving the quality and relevance of generated questions.

[10] Patel, A., Iyer, S., & Fernandez, K. (2021) – "Evaluating the Effectiveness of AI-Generated Questions in Educational Settings"

This study evaluates the effectiveness of AI-generated questions in real classroom settings, comparing them with manually created ones in terms of student engagement, difficulty level, and learning outcomes, demonstrating the potential of automated systems to enhance educational assessments.

Key Technologies used:

The reviewed literature The Test Questions Generator Application incorporates a number of new tools and frameworks to generate automatic questions from educational material in the most efficient, scalable, and accurate manner possible. In this system, there are several natural language processing (NLP) models, large language models (LLMs), vector databases, etc., that will allow the application to generate questions in many different contextual formats from content extracted from a PDF file. The overall description of the technology stack is given below.

- **Language Models:** The application uses OpenAI's GPT-3.5-turbo using the LangChain framework to train on the educational material to produce contextually accurate questions. The LLMs uses the semantic structure of the input text to produce question types of the factual, conceptual, and definitional or explanatory types from various formats of question prompts. The LangChain API - ChatOpenAI, allows for the direct calling of the OpenAI GPT-3.5-turbo model, prompts, and responses in a desirable format without have to re-invent the 'prompting wheel' for custom prompts and structured prompts.
- **Text Chunking and Semantic Search:** With the raw text extracted from PDF files using PyPDF2, the application uses LangChain's tools to chunk the below text with the RecursiveCharacterTextSplitter. The chunking process allows for large text-easily batch handled and to retrieve relevant context when developing appropriate questions. The text chunks were indexed with FAISS (Facebook AI Similarity Search), an open-source vector database that allows for similarity-based searching to retrieve contextually similar text passage.
- **OpenAI Embeddings:** The application uses OpenAIEmbeddings with FAISS as the text chunk database texture43h to map an input chunk of text (using how the input chunk was indexed) to a high-dimensional vector space to search for semantically similar text segments that will align with the user's selected question type.
- **Adding, Structured Prompting and Question Generating**: The application has structured prompting (ChatProntTemplate) for Multiple Choice, short answer, and long answer prompting. This structured prompt allows the LLM to respond to structured question prompts. The output from the LLM will be in strict JSON formatting which allows the application to easily parse and display the output from the LLM's responses.
- **Streamlit User Frontend:** The application developed using Streamlit is a user friendly interface that allows educators to upload a PDF file (or text), select the formatting for each question, and to visualize ALL question and answers. Moreover, Streamlit allows for immediate interaction which makes it a straightforward technology to deploy without requiring background in advanced web apps development.

- **Parsing and session state when generating JSON output:** The application will use json library to parse the LLM-JSON formatted output since Python allows for easily parsing impromptu data that was returned from the LLM. Streamlit's session state [st.session_state] will be used for saving and displaying ALL generated questions from the interface when the educator interacts with the app.

This technology stack means the entire application is lightweight, interactive, and highly adaptable for various educational contexts. It has a modular architecture that permits adjustments to future feedback-based reinforcement learning solutions; culminating with question quality and questions type diversity formulating more "dynamic" question sets..

Research Methodologies used

The Test Questions Generator Application was developed using applied research and experimental design through iteration and focused on Natural Language Processing (NLP) and Machine Learning (ML) techniques. The research processes are outlined in the following section.:

- **Applied Research Approach**

This was an applied research approach which attempted to address the real-world issue of converting an academic resource into pedagogically appropriate questions. The research focuses on developing a usable tool for educators and students through the implementation and adjustment of existing technologies (for example, large language models, vector search), not on generating new theoretical models.**Weapon Detection**

- . **Experimental Design and Prompt Engineering**

A large portion of application development consisted of experimental prompt engineering, which involved creating prompt design templates. The templates were modified and tested multiple times in an effort to create a high-quality, structured output from the GPT-3.5-turbo language model. This included: [4].

- Developing prompt designs for variations of question types (MCQs, Short Answer, Long Answer).
- Testing how the model responded in different content contexts.
- Evaluating whether or not the output was syntactically and semantically correct.
- There was an iterative cycle of testing every version for accuracy, variety, and relevance of the questions produced.using TensorFlow for low-latency detection in surveillance environments[6].
- **Data-Driven Evaluation (Qualitative and Quantitative)**

The application runs semantic searches that implement FAISS vector indexing of academic resources using the OpenAI text embeddings. Using a variation of retrieval effectiveness, the relevance of each chunk that is ultimately retrieved for the focus of each question type. A qualitative evaluation of relevance was used (manual check) and a quantitative evaluation of retrieval accuracy, and model output consistency.

- **Rule-Based and NLP-Based Filtering**

While a transformer model is used to generate questions, prompt engineering is used to ensure question diversity and quality through embedding rule-based constraints within the prompts. Those constraints prompt the model to avoid producing overlapping questions as well as make sure answers are complete and educational governance is followed.

- **Simulated Testing**

Since real-life deployment was not part of the original plan, simulated testing was conducted using extracts of PDFs containing chapters of textbooks and lecture notes. The outputs of the extracted PDFs were compared to the manually created question set to evaluate content relevance and cognitive load.

These methodologies, powered by cutting-edge algorithms and libraries, address the limitations of traditional systems and enable building of a robust application.

Challenges

The development of the Test Questions Generator Application, while innovative and impactful, posed several technical and methodological challenges. These challenges were encountered at various stages of the application lifecycle, from data handling to model integration and interface design.

- **Shallow & Generic Questions:** Existing systems create questions based on keyword matching/templates, so they create shallow, generic, repetitive questions that do not rely on deep understanding or critical thinking skills.
- **No Contextual Relevance:** Existing systems typically have limited context-level understanding of the input text, so questions they ask will typically not meet contextual relevance and/or learning goals.
- **Limited Question Types:** Most systems are limited to the basic question types (e.g. multiple choice, fill-in-the-blanks, etc.) and can't create more complex question types such as essay questions or analytical prompts.
- **Limited Flexibility:** Existing systems demand a very fixed view of subject matter, education level, and/or learning objective (or all three) limiting their effectiveness across a multitude of curricula and students' learning requirements.

- **Unpredictable Quality:** Given the lack of context understanding and insufficient agent flexibility, predictable questions will be generated with unpredictable quality requiring educators to track, adjust, or revise them - wasting their time.

Gaps to be Addressed

Currently, the Test Question Generator Application is a major step toward automating educational content generation, but there are still several areas that need to be developed in order to improve the application as a useful tool in terms of efficiency, accuracy, and adaptability. The gaps below describe the areas which need improvement.

- **Domain-Specific Question Generation:**

The application currently generates generic academic questions, but does not easily allow the content to fit into a particular domain such as medicine, law, or engineering. By fine tuning with domain-specific datasets, the application could generate questions to specific domains that are very relevant and accurate.

- **Multi-Modal Content:**

The content that the application currently utilizes for question generation is only text that is extracted from PDF files. This diminishes the educational value found in tables, charts, formulas, and images, etc. Adding multi-model data extraction capabilities to the application will allow it to generate questions based on a larger amount of material than is currently available.

- **Automated Quality Assessment:**

Currently, there are no methods for assessing the educational quality, difficulty, or relevance to an intended audience of the generated questions. If the application incorporated automated assessment tools, then users would know that any questions generated would be consistent pedagogical practices and would be satisfactory challenges for different learners.

- **Customization of question complexity:**

Users have minimal control of the complexity and cognitive level of questions (e.g., recall, analysis, evaluation, etc.). By adding features for customization, educators could generate questions that are consistent with recommended outcomes for learner(s) in the learning process aligned with specific educational outcomes using Bloom's taxonomy levels.

- **Scalability and cost:**

Considering the application uses third party APIs such as OpenAI for generating content, there is a possibility that the application may have performance issues or high operating expenditures to maintain during periods of high usage. If the application can be implemented to bolster performance, along with costs for utilization via scalable architectures, caching, and local model hosting, the application should decrease concerns over costs for ease of use by educators in their educational obligations.

- **Error resilience and robustness:**

Currently the application has basic error handling and robustness and may not deal appropriately with unexpected model actions or failures in 3rd party APIs. Improvements to error detection, notification to the user, and recovery measures would allow the application to ensure smoother user experiences.

- **Real-time user feedback and incorporation:**

The application does not currently provide users real-time feedback to improve future question generation. Multi-directional feedback processes via user interaction can enable the system to learn continuously from users, which can enable improved outputs that are more aligned with an educational need.

- **Personalized by learner profile:**

Currently users are able to access generic question sets poorly tailored to learning goal or towards users background and proficiency level. Personalization features that take into consideration a user profile or past performance will greatly increase engagement and improve educational outcomes for learners.

If these gaps are addressed appropriately, the New Test Question Generator Application can become a seamlessly integrated intelligent, user-centric educational tool for diverse learners and educators.

3. Existing System

The classic test question generation methods involved a reliance on manual effort of instructors, subject matter experts, and instructional designers. Users carefully read through textbooks, lecture notes, and reference materials used in courses to create test questions that demonstrate student comprehension and critical thinking abilities. The manual method, while producing quality and providing assurance that the generated questions align to course objectives, is slow, labour-intensive, and sometimes biased due to a human conception of major concepts.

In the last several years, the automated question generation approach has been developed, using rule based NLP (Natural Language Processing) techniques. The automated approaches pull keywords or important sentences from the text provided to generate either fill-in-the-blank or multiple choice questions. However, such approaches are limited in their capacities because of one or more of the following design limitations:

- **No True Context**:

Traditional systems often ignore context, shallow parsing and extracting keywords to create shallow questions. This results in questions that lack depth, coherence or relevance in context to the material the users have read.

- **Very Limited Generating of Question Formats:**

Most systems provide automated processes for only basic questions at best (know factual recall - fill-in-the-blank) - the sense during question generation of complex formats e.g. short answer or long answer (to demonstrate higher order thinking) is not available in existing systems.

- **Minimal Adaptability:**

Current systems often lack flexibility in adapting to different subject domains or different difficulty levels. This limits the applicability of existing automation systems in some class settings.

- **Preprocessing data:**

Many systems require a considerable amount of "pre"-processing. All data needs to be prepared for question generation (clean formatted text, or structured content) or plain text data generated from an assigned reading could exceed the need for new or relevant questions.

- **Quality Assurance:**

Questions generated automatically still may need to be reviewed or edited to ensure correct; relevant; and clear questions. This is contrary to the benefits of time-saving with any automation.

Overall, while traditional systems have been helpful to reduce the burden of creating test questions and early question generation systems have made some advances toward easing the burden of test creation, no fully autonomous, fully diverse, fully context aware question generation, suitable for a dynamic educational environment has been produced to date.

Disadvantages of Existing Systems

Although standard and early automated systems for test question generation have played a role in educational assessment procedures, there are several relevant drawbacks that restrict their usage.

- **Time-Intensive and Laborious:**

Developing questions manually requires substantial amounts of time for developers, typically needing to be revised multiple times to meet threshold quality and ensure aligned learning outcomes.

- **Limited Contextual Awareness:**

Rule-based approaches and keyword extraction approaches lack a deep understanding of the actual input material and as a result, they frequently develop simple, bland and inappropriate questions that omit or miss core concepts.

- **Limited Range of Question Types:**

To date, existing systems have demonstrated primarily the capability to produce individual factual recall-type questions or fill-in-the-blank questions, while limited to no higher order questions, including short answer, essay, or case-based questions, that tap into analytical thinking skills.

- **Minimal Personalization:**

Most systems do not take into account personal characteristics (e.g., a learner's prior knowledge, learning goals, or level of study) when developing question sets, and produce one-size-fits-all question sets.

- **Over-reliance on Clean Data:**

First, traditional models are often only functional when provided cleaned or well-structured input texts. For example, tables, figures, or even poorly formatted documents can create a pivotal breaking point for question generation.

- **Lack of Real-Time Feedback Adaptation:**

In a normal educational environment there is typically no mechanism for adapting questions based on feedback from the user or the user's prior and evolving tastes as a learner.

- **Generated Errors:**

Because many automated systems lack the deep natural language understanding developed through human complexity, it is not uncommon for some automated systems to produce poor output, for example, errors in grammar (and) wrong answers, ambiguities, and generating misleading questions, which write-backs to the following bullet point and requires the need for a human to determine if corrections are necessary in the first place.

- **Scalability Issues:**

Some of these traditional models cannot efficiently handle larger datasets or real-time content generation, severely restricting their use in an educational context of scale or in large-scale, non-linear online learning experiences and environments.

In summary, Illustrated above are the major limitations of these question generation models, which necessitate the need for more sophisticated, intelligent, and adaptable question generation systems that produce higher quality contextually aware and diverse test questions with little human effort.

4. Conclusion

The Test Question Generator app is a great example of how AI can add considerable value to instructional tools, by generating questions and answers from educational resources such as PDFs and textbooks. The application provides for both multiple choice and long-answer input, demonstrating the variety of contexts and useful applications in an academic learning environment. Natural Language Processing and transformer modelling takes out a lot of the time and effort involved in meaningful assessment item creation by educators while simultaneously supporting individual learning for students. The clarity of the application interface, instantaneous processing, and flexibility with multiple question formats aids greatly in ease of use for both educators and students alike. The application very quickly parses complex academic texts, generates viable questions, and literally produces the questions following Bloom's Taxonomy. These are all huge advantages for use during any kind of academic assessment and in particular for adaptive testing.

Key Learnings

- **Understanding and Using Transformer Models :**

You have learned how to apply Transformer-based models, like GPT-3.5, not merely at the level of light, simple applications (like chatbots), but also at a large scale for meaningful applications, like context-aware question generation. You learned how to distinguish between more specific prompts utilized to control the outputs from the model, and create structured output formats (like JSON).

- **Methods of prompt engineering :**

Importantly, you took the analysis and made it action and understood the importance of prompt design. You sensed how crafting a specific and deliberate system prompt (three types of MCQs, short, long) meant you were able to direct a large language model (LLM) to produce relevant, variate, high quality questions.

- **Document Extraction and Preprocessing :**

You tried a variety of mechanisms to extract, clean raw text from from PDFs using PyPDF2, and identify new tools to perform, text chunking. Thus, turning very large amounts of text, into manageable pieces, using RecursiveCharacterTextSplitter for making more efficient processing and retrieval.

- **Vector store and semantic search :**

You used FAISS to store text embeddings, and conduct efficient similarity searches (which you had not done until the project). This gave you an understanding of semantic search in relation to embeddings as potentially capturing contextual meaning as opposed to very basic keyword-matching meaning.

- **Streamlit instead of Flask :**

You learned that you can use Streamlit and create an interactive web app, instead of a more heavyweight framework, like Flask. In a short period of time, you were able to learn how to handle file uploads, a session state, user inputs, dynamically displaying information at user prompting, deployment practices, along with many other valuable skills for the web interface.

- **Integrating LLMs with external processes :**

The action research project introduced you to LLMs (and their ability to utilize LangChain and OpenAI APIs) and how, together, they could be orchestrated alongside other workflows and processes like document processing, vector store, and front-end framework - in an entire pipeline.

- **Error processing and resilient error handling :**

The project pushed you into stuckness many times--like when the model responses wouldn't parse to JSON or, in some instances, wouldn't include a certain field at all. This indicated to you a slight glimpse of the importance of a resilient error and processing handling to provide users a palatable way to explore and assess their input conditions.

- **Appreciation of the limitations of LLMs :**

You have experienced the applicability limits of the models many times. More than worries regarding the accuracy of answers, the gap in expected responses compared to visual displays for formatted answers, was even more profound. Validation and post-processing model outputs, before dissemination to end users, also became critically important.

- **Appreciate the importance of scalability and optimization :**

You appreciated how little incremental thoughts could dramatically slow the responsiveness of the application, as would larger amounts of documents. More understanding of chunking documents effectively, selective searching of documents and limiting number of tokens could (and will) greatly improve its performance and usefulness - and there are considerations of scalability when moving to practical use.

- **Basic structure of educational technology (EdTech) :**

Most importantly, you have developed a very stable, basic structure of how AI might be used in education, beyond ideas or the heuristics of trying to make tools that automate boring things, while also maintaining the learning and pedagogical practice obligation.

Future Scope

The systems we have proposed can certainly be further developed and improved. Future versions of the project can consider the following improvements:

- **More File Types Supported:**

Future versions can be expanded to accept other formats beyond PDF, including Word documents, markdown files, and even scanned handwritten notes with the use of OCR (Optical Character Recognition).

- **More Languages Supported:**

If the system can develop NLP capabilities in multiple languages it could also allow for non-English learners and institutions to utilize the tool worldwide.

- **Question Bank Export and Analytics:**

Features for exporting banks of questions into CSV, Word or PDF as well as features for performance analytics would be a tremendous benefit for educators to monitor performance and revisit curricular as needed.

- **More Guardrails for Safety and Accuracy of Content:**

Implementing more guardrails such as hallucination detection mechanisms, context validations, and prompt filtering would create better quality and factual accuracy regarding questions and answers produced. It will create safety nets to ensure the content produced to remain relevant, ethical and educationally plausible, especially when applied in academic spaces.

- **Integration with Learning Management Systems (LMS):**

The application could be made to work with the LMS of choice (Moodle, Canvas, Google Classroom, and many others) with simple import/export functionality of questions that best meets the workflow for teachers and institutions.

- **Interactive Quiz Generation**

Instead of creating 'just' questions, the system could be developed to also create interactive quizzes or mock exams that have features such as automatic scoring, hints, testing time limits, and feedback for incorrect answers.

- **Customized Question Sets based on User Profiles**

The system could develop a lightweight user profile (such as age of the student, grade level or subject expertise) for the primary user, enabling the system to generate customized and targeted question sets for different learners which would be more engaging and beneficial to the learner.

- **Better Summarisation and Context Extraction**

Before any questions were generated, a text summarization module could be added, enabling users to process large text documents into a summarisation and outline of the key topics which would ensure that generated questions would be highly focused on learning what was critical and important.

- **Visual Question Generation**

Future versions could include image question generation where diagrams, charts or images in the PDF could be used to generate associated questions – for subject areas, such as science, geography, or engineering.

- **Feedback Loop and Reinforcement Learning**

Giving the user (teachers or students) the ability to rate the quality, difficulty, and relevance of generated questions could create a feedback loop to re-train or fine-tune models through reinforcement learning over time to help the system improve without retrospectively confounding the dataset once it is cut off.

- **Offline and Mobile Application**

A lightweight mobile app or an offline desktop version of the application would increase accessibility function, especially for rural or low internet connectivity areas where an online system would not be feasible, or users would simply prefer offline connectivity.

- **Voice Entry and Voice Interaction**

To enable voice entry with speech-to-text, and allow the user to dictate their notes or topics and the system will generate them into questions without any text input. Enabling accessibility and equity for users who are sight impaired or who are working in fast pace classroom environments as an example.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is All You Need,” Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 5998–6008.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, 2019, pp. 4171–4186.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Few-Shot Learners,” OpenAI GPT-3 Technical Paper, 2020, pp. 1–10.

[4] S. Chakraborty, H. Paul, S. Ghatak, and M. A. Shah, “An AI-Based Medical Chatbot Model for Infectious Disease Prediction,” International Journal of Computer Applications, vol. 182, no. 27, 2023, pp. 30–36.

[5] S. Badlani, T. Aditya, M. Dave, and S. Chaudhari, “Multilingual Healthcare Chatbot Using Machine Learning,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 15, no. 1, 2024, pp. 23–29.

[6] R. B. Mathew, S. Varghese, S. E. Joy, and S. S. Alex, “Chatbot for Disease Prediction and Treatment Recommendation,” Journal of Emerging Technologies and Innovative Research (JETIR), vol. 8, no. 2, 2023, pp. 150–157.

[7] A. Tiwari, R. Raj, S. Saha, P. Bhattacharyya, and M. Dhar, “Toward Symptom Assessment Guided Symptom Investigation and Disease Diagnosis,” Proceedings of the International Conference on Computational Intelligence and Computing Research (ICCIC), 2023, pp. 200–206.

[8] S. Kumar and R. Gupta, “Automated Question Generation for Educational Assessments Using NLP,” Journal of Education and Information Technologies, vol. 25, no. 6, 2022, pp. 1343–1358.

[9] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, and W. Macherey, “Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” Communications of the ACM, vol. 61, no. 7, 2016, pp. 113–124.

[10] V. Gupta and A. Sood, “Reinforcement Learning for Adaptive NLP Systems,” Journal of Artificial Intelligence Research (JAIR), vol. 75, no. 3, 2022, pp. 89–96.

[11] T. Brown, B. Mann, N. Ryder, et al., "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 1877–1901.

[12] J. Kiros, Y. Zhu, R. Salakhutdinov, et al., "Skip-Thought Vectors," Advances in Neural Information Processing Systems, 2015.

[13] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Proceedings of the International Conference on Learning Representations (ICLR), 2014.

[14] R. Ying, R. He, K. Chen, P. Eksombatchai, W. Hamilton, and J. Leskovec, "Graph Convolutional Neural Networks for Web-Scale Recommender Systems," Proceedings of KDD, 2018.

[15] H. Wang, Z. Lin, and Y. Li, "Automatic Question Generation from Text for Educational Purposes: A Survey," Journal of Artificial Intelligence in Education, vol. 31, no. 2, 2021, pp. 345–367.

[16] D. Lin, "An Information-Theoretic Definition of Similarity," Proceedings of ICML, 1998.

[17] S. Rajpurkar, R. Jia, and P. Liang, "Know What You Don’t Know: Unanswerable Questions for SQuAD," Proceedings of ACL, 2018.

[18] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative Study of CNN and RNN for Natural Language Processing," arXiv preprint arXiv:1702.01923, 2017.

[19] Y. Goldberg, "Neural Network Methods for Natural Language Processing," Synthesis Lectures on Human Language Technologies, vol. 10, no. 1, 2017.

[20] L. Dong, F. Wei, and M. Zhou, "Multi-Task Learning for Question Generation," Proceedings of ACL, 2017.

[21] R. Mihalcea and P. Tarau, "Textrank: Bringing Order into Texts," Proceedings of EMNLP, 2004.

[22] C. Raffel, N. Shazeer, A. Roberts, et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research, 2020.

[23] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," Proceedings of ACL, 2017.

[24] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems," Proceedings of SIGDIAL, 2015.

[25] T. Wolf, L. Debut, V. Sanh, et al., "Huggingface's Transformers: State-of-the-Art Natural Language Processing," EMNLP System Demonstrations, 2020.

[26] M. Lewis, Y. Liu, N. Goyal, et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation," Proceedings of ACL, 2020.

[27] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli, "Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models," Communications of the ACM, vol. 66, no. 1, 2023.

[28] J. Gao, M. Galley, and L. Li, "Neural Approaches to Conversational AI," Foundations and Trends in Information Retrieval, vol. 13, no. 2-3, 2019.

[29] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic Propositional Image Caption Evaluation," Proceedings of ECCV.

[30] P. Lewis, E. Perez, A. Khandelwal, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems, 2020.

[31] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," Proceedings of EACL, 2017.

[32] H. Zhao, A. Zhang, and R. Salakhutdinov, "Learning Deep Generative Models of Text," Advances in Neural Information Processing Systems, 2017.

[33] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep Unordered Composition Rivals Syntactic Methods for Text Classification," Proceedings of ACL, 2015.

[34] B. Yang, W. Cohen, and R. Salakhutdinov, "Revisiting Semi-Supervised Learning with Graph Embeddings," Proceedings of ICML, 2016.

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., "An Image is Worth 16x16 Words: Vision Transformers (ViT) for Image Recognition at Scale," Proceedings of ICLR, 2021.

[36] E. Grave, A. Joulin, and N. Usunier, "Improving Neural Language Models with Low-Rank Matrix Factorization," Proceedings of ICLR, 2018.

[37] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," arXiv preprint arXiv:1706.05098, 2017.

[38] H. Inan, K. Khosravi, and R. Socher, "Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling," Proceedings of ICLR, 2017.

[39] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What Does BERT Look at? An Analysis of BERT's Attention," Proceedings of BlackboxNLP, 2019.

[40] C. Szegedy, W. Liu, Y. Jia, et al., "Going Deeper with Convolutions," Proceedings of CVPR, 2015.