ExamGenerator

Zac Castaneda
Department of Computer Science
Texas A&M University - Corpus Christi
Corpus Christi, Texas, United States of America
Ecastaneda7@islander.tamucc.edu

Ryan Grieb
Department of Computer Science
Texas A&M University - Corpus Christi
Corpus Christi, Texas, United States of America
rgrieb@islander.tamucc.edu

Christopher Corkill
Department of Computer Science
Texas A&M University - Corpus Christi
Corpus Christi, Texas, United States of America
ccorkill@islander.tamucc.edu

Nico Elwell
Department of Computer Science
Texas A&M University - Corpus Christi
Corpus Christi, Texas, United States of America
nelwell@islander.tamucc.edu

Gabriel Cavazos
Department of Computer Science
Texas A&M University - Corpus Christi
Corpus Christi, Texas, United States of America
Gcavazos3@islander.tamucc.edu

***Abstract*— ExamGenerator is an online tool driven by artificial intelligence that can create exam questions and answers automatically from document inputs including Word, PDF, and text files. This article presents ExamGenerator. The application turns course materials into extensive tests and quizzes with multiple-choice, true/false, and multiple-selection questions by leveraging Open AI's ChatGPT API and additional libraries. The purpose of ExamGenerator, associated research, suggested methods, and system architecture are all described in this study. The system attempts to provide high-quality evaluations in a substantial amount less time than educators currently spend creating them. ExamGenerator has the potential to have a significant impact on exam development efficiency, assessment quality uniformity, and question type generation.**

Keywords—ChatGPT, OpenAI

I. Introduction

Today's ever changing digital education environment has made good evaluation tools imperative. Exam question generation using traditional methods is labor- and time-intensive, which forces educators to look for creative solutions that streamline the process. ExamGenerator emerges as a solution to these challenges, utilizing cutting-edge technologies to enhance both the efficiency and accuracy of exam creation. By automating key steps and offering customizable features, it empowers educators to focus on what truly matters—ensuring high-quality education. Furthermore, its intuitive interface makes it accessible to educators of all technical levels, providing a seamless transition from outdated methods. By tapping into vast educational content and applying sophisticated natural language processing techniques with OpenAI, ExamGenerator will be able to produce professionally made exam and quiz content.

A. Problem Statement

Creating exam questions from educational resources usually requires a lot of manual work, which can be time-consuming and prone to mistakes. Teachers often struggle with limited time, maintaining consistent question quality, and covering all the important material. As digital content grows, manually going through documents to find relevant information for assessments becomes increasingly impractical. As a result, there is a pressing need for an automated solution that can efficiently transform diverse educational documents into structured exam formats while maintaining accuracy and relevance.

B. Motivation

ExamGenerator was developed with the goal of simplifying the process of producing excellent tests and quizzes for instructors and educational establishments. ExamGenerator saves teachers time and allows them to concentrate on improving the learning environment for their students by automating the question production process. Our team has a special interest in this solution because some of us are from educational families. Our team is motivated to create a technology that relieves instructors of the load of creating exams since we have seen firsthand the difficulties, they encounter in juggling administrative responsibilities and classroom management. Our dedication to providing a solution that tackles practical issues and frees up educators to focus more on what matters most in education—teaching—is motivated by our personal connection to the teaching profession.

II. Related Work

Related tools that utilize AI text generation to assist educators and students:

- Quizlet: This popular study tool allows users to create flashcards and quizzes, but relies on user input for question generation and lacks automation features.

- Socrative: An interactive platform for quizzes and polls, Socrative offers real-time feedback but requires manual question creation, limiting scalability for large educational materials.
- ProProfs Quiz Maker: This online tool enables quiz creation with various question types and templates. However, it focuses on manual input and does not automate question extraction from educational documents.
- QTI-compliant Tools: These tools adhere to the Question and Test Interoperability standard for sharing assessment content but do not automate question generation, which remains a manual task.
- AI-Powered Educational Platforms: Platforms like Gradescope and Respondus streamline grading and online testing but do not focus on generating questions from documents.

ExamGenerator, in contrast to similar work, automates the creation of test questions from a range of document formats using natural language processing.

III. Proposed Approaches

ExamGenerator is a deliberate invention that meticulously integrates cutting-edge technology, contemporary frameworks, and user-centered design concepts to produce a scalable, highly efficient exam generation automation solution. With the use of cutting-edge natural language processing, adaptable file format management, and smooth connection with educational platforms, the tool aims to provide educators with a flexible and intuitive interface. The key approaches include:

A. Natural Language Processing with APIs:

ExamGenerator intends to leverage the ChatGPT API for natural language processing tasks. This integration will enable the application to analyze and interpret educational documents, extracting relevant information to generate coherent and contextually appropriate exam questions. By employing advanced NLP techniques, the system will be capable of understanding complex educational language, ensuring that questions are not only accurate but also diverse in format and complexity. This use of ChatGPT is expected to enhance the quality and variety of the generated content, allowing for tailored questions that meet different learning objectives.

B. Import Formats:

ExamGenerator will be designed to support various input file formats, including PDF, DOCX, and Excel. Because of this flexibility, instructors will be able to upload instructional resources in the format of their choice, increasing the tool's accessibility. Multiple file formats are included to make it easy for users to incorporate the tool into their current processes. In addition, the system will make use of libraries like pandas for Excel spreadsheets and python-docx for DOCX files in order to guarantee precise content extraction and facilitate the processing of various educational resources.

C. Export Formats:

The tool will offer multiple options for exporting generated exams, including PDF, DOCX, and plain text formats. This flexibility would allow educators to use the generated assessments across various platforms while maintaining compatibility with existing educational standards. Additionally, ExamGenerator could integrate with popular learning management systems such as Blackboard, Canvas, and Moodle, enabling users to export exams directly into these platforms. This would streamline instructor operations and lessen the need for manual imports by making the deployment of assessments and management of student contributions simpler.

ExamGenerator hopes to serve instructors and students alike by combining these strategies to offer an effective, reliable, and user-focused solution for automating the creation of exam questions.

IV. System Design

The system design for ExamGenerator is structured to leverage modern technologies and methodologies to ensure the proposed approaches are able to be implemented in a timely manner. The architecture consists of several key components:

A. Web Framework:

The application will be developed using the Quart framework, selected for its support of asynchronous actions. Quart is an asyncio-based framework that allows for non-blocking requests, enabling ExamGenerator to handle multiple user requests concurrently. This capability is crucial for providing a responsive user experience, especially when processing large documents or handling simultaneous requests from multiple educators. Quart's compatibility with the Flask ecosystem also allows for easy integration of existing Flask extensions, which can enhance functionality.

B. Containerization with Docker:

To enhance deployment and scalability, ExamGenerator will utilize Docker for containerization. Docker will package the application and its dependencies into isolated containers, ensuring consistency across different environments—development, testing, and production. This approach simplifies the deployment process, allowing the development team to replicate environments easily and reducing issues related to environment discrepancies. Docker also supports orchestration tools like Docker Compose and Kubernetes, enabling easy management of multiple containers and microservices, which can be beneficial as the application scales.

C. Database Management:

A MariaDB database will be employed to manage and store user data, exam questions, and historical records. MariaDB is chosen for its reliability, performance, and compatibility with MySQL, making it a robust option for relational data storage. The database schema will be designed to efficiently handle various data types, including user profiles, uploaded documents, generated exams, and user feedback. Proper indexing and normalization will ensure optimal query performance, enabling fast access to data during the exam generation process. Additionally, the database will support data backup and recovery strategies to prevent data loss.

D. Document to Text Conversion:

To facilitate the extraction of text from various document formats, ExamGenerator will integrate the unstructured-io API for converting PDF files to structured text. Additionally, the application will support other document types by utilizing libraries such as python-docx for handling DOCX files, allowing extraction of text and formatting information; pandas for reading Excel files, enabling the conversion of spreadsheet data into a structured format suitable for exam generation.

E. Natural Language Processing:

The application will leverage the ChatGPT API for natural language processing tasks. This integration will enhance the application's ability to analyze educational content, ensuring high-quality and diverse exam questions are generated from the provided materials. The system can comprehend context, recognize important concepts, and create questions that meet academic requirements by utilizing sophisticated natural language processing (NLP) techniques. In order to accommodate users with varying educational backgrounds, the system can also modify the difficulty of the questions according to user-specified parameters.

F. User Interface:

Teachers will be able to engage with the application with ease because to the intuitive and user-friendly design of the user interface. A preview and edit feature will allow users to review generated questions and make any necessary formatting or wording changes before finalizing the exam. Other key features will be the ability for users to upload educational materials in supported formats (PDF, DOCX, Excel); parameter selection options for specifying exam generation parameters, such as the number of questions, desired wording styles, and question formats (e.g., multiple-choice, true/false, or open-ended questions); and a variety of export options that will allow users to select the format (PDF, DOCX, or plain text) for the generated exams. The design will prioritize accessibility and responsiveness, ensuring a smooth experience across devices and screen sizes.

V. References

TBA

VI. Appendix

TBA