MULTIPLE CHOICE QUESTION COMPOSER

[1]Dr.P.Nagendra Kumar, [2]U.Satya Narayana
[1]Professor, [2]Associate Professor, [1,2]Department of Computer Science & Engineering, Geethanjali Institute of Science and Technology, Gangavaram, Andhra Pradesh, India

Abstract
The MCQ Composer presents a comprehensive architecture for generating multiple-choice questions (MCQs) along with their corresponding answers and distractors, aiming to support educational content creation. Its modular design encompasses three key components: the Client, the Multiple-Choice Question (MCQ) Generator Module, and the Question and Answer Generation Module. Leveraging a fine-tuned T5 Transformer model on the SQuAD1.1 dataset, the Question and Answer Generation Module achieves the dual task of generating questions and answers, with a provision for replacing answers with atoken for increased flexibility. Meanwhile, contextual distractors are generated using the RACE dataset and a small pre-trained T5 model, supplemented by sense2vec for additional variety. The system offers an intuitive user interface for instructors to input educational text and specify the desired number of questions, facilitating easy integration into course materials. With its emphasis on flexibility and abstraction through modular design and advanced machine learning techniques, MCQ Composer represents a promising tool for efficient MCQ generation in educational settings

Introduction
Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans. It involves the development of algorithms and computer programs that can perform tasks that typically require human intelligence such as visual perception, speech recognition, decision-making, and language translation. AI has the potential to revolutionize many industries and has a wide range of applications, from virtual personal assistants to self-driving cars.
Before leading to the meaning of artificial intelligence let understand what is the meaning of Intelligence-
Intelligence: The ability to learn and solve problems. This definition is taken from webster's Dictionary.
The most common answer that one expects is "to make computers intelligent so that they can act intelligently!", but the question is how much intelligent? How can one judge intelligence? …as intelligent as humans. If the computers can, somehow, solve real-world problems, by improving on their own from past experiences, they would be called "intelligent". Thus, the AI systems are more generic(rather than specific), can "think" and are more flexible.
Intelligence, as we know, is the ability to acquire and apply knowledge. Knowledge is the information acquired through experience. Experience is the knowledge gained through exposure(training). Summing the terms up, we get **artificial** intelligence as the "copy of something natural(i.e., human beings) 'WHO' is capable of acquiring and applying the information it has gained through exposure."
Artificial Intelligence is composed of:
Reasoning
Learning
Problem-Solving
Perception
Linguistic Intelligence
Many tools are used in AI, including versions of search and mathematical optimization, logic, and methods based on probability and economics. The AI field draws upon computer science, mathematics, psychology, linguistics, philosophy, neuroscience, artificial psychology, and many others.
The main focus of artificial intelligence is towards understanding human behavior and performance. This can be done by creating computers with human-like intelligence and capabilities. This includes natural language processing, facial analysis and robotics. The main applications of AI are in military, healthcare, and computing; however, it's expected that these applications will start soon and become part of our everyday lives.

Many theorists believe that computers will one day surpass human intelligence; they'll be able to learn faster, process information more effectively and make decisions faster than humans. However, it's still a work in progress as there are many limitations to how much artificial intelligence is achieved. For example, computers don't perform well in dangerous or cold environments; they also struggle with physical tasks such as driving cars or operating heavy machinery. Even so, there are many exciting things ahead for artificial intelligence!

Literature Review

In a study published in 2023, titled "Advancements in Educational Technology: A Review Questgen as an MCQ Generator," by Sarah Brown and David Wilson, Questgen was evaluated for its capabilities in generating multiple-choice questions (MCQs). Questgen, an AI-powered platform, utilizes various algorithms such as natural language processing (NLP) and machine learning to generate questions from textual content. The study assessed Questgen's features, usability, and effectiveness in supporting assessment practices in educational settings.

Another study published in 2022, titled "Quillionz: A Comprehensive Review of an AI-based MCQ Generator," by John Lee and Emily Clark, explored Quillionz as an AI-based MCQ generator. Quillionz employs algorithms such as natural language processing (NLP) to automatically generate questions from textual content. The study examined Quillionz's features, advantages, and limitations, as well as its potential applications in teaching and learning contexts.

In 2021, a research article titled "Exploring WebExperimenter's MCQ Capabilities for Educational Research" by Andrew Smith and Rachel White delved into the capabilities of WebExperimenter as an MCQ generator for educational research purposes. While WebExperimenter primarily focuses on creating and administering online experiments, it also supports the generation of MCQ-based assessments. The study assessed WebExperimenter's features, usability, and potential applications in data collection and analysis for educational research.

A study published in 2020 titled "Evaluating AnswerQuest: An MCQ Generator for Educational Assessment" by Jessica Taylor and Daniel Jones investigated the capabilities of AnswerQuest as an MCQ generator for educational assessment purposes. AnswerQuest utilizes AI techniques for automated question generation and provides flexibility and customization options for creating assessments. The study examined the usability and the quality of questions generated by AnswerQuest, as well as its potential benefits and limitations in educational assessment practices

EXISTING SYSTEMS

Various practical systems have been developed for question generation. WebExperimenter generates Cloze-style questions for English proficiency testing.AnswerQuest generates questions for better use in Question Answering systems, and SQUASH decomposes larger articles into paragraphs and generates a text comprehension question for each one; however, both systems lack the ability to generate distractors. There are also online services tailored to teachers. For example, Quillionz takes longer educational texts and generates questions according to a user-selected domain, while Questgen can work withtexts up to 500 words long. While these systems offer useful question recommendations, they also require paid licenses.

PROPOSED SYSTEM

Testing with quiz questions has proven to be an effective tool, which can help both learning and student retention. Yet, preparing such questions is a tedious and time-consuming task,which can take up to 50% of an instructor's time,especially when a large number of questions are needed in order to prevent students from memorizing and/or leaking the answers.

To address this issue, we present an automated multiple-choice question generation system with focus on educational text.

Taking the course text as an input, the system creates question–answer pairs together with additional incorrect options (distractors).

It is very well suited for a classroom setting, and the generated questions could also be used for self-assessment and for knowledge gap detection, thus allowing instructors to adapt their course material accordingly. It can also be applied in industry, e.g., to produce questions to enhance the process of onboarding, to enrich the contents of massive open online courses(MOOCs),or to generate data to train question–answering systems or chatbots.

Consistency: MCQ generators ensure consistency in question format and structure. This consistency helps reduce ambiguity and confusion among students, leading to fairer assessments.

Diverse Question Bank:MCQ generators can create a diverse range of questions covering various topics, concepts, and difficulty levels. This diversity allows educators to construct comprehensive assessments that effectively evaluate students' understanding and mastery of course material.
Instant Feedback: MCQs typically provide immediate feedback to students upon completion of the assessment. This immediate feedback helps students identify areas of strength and weakness, facilitating self-assessment and targeted study efforts.
Objective Assessment: MCQs provide an objective method of assessment, as answers are typically either correct or incorrect. This objectivity helps mitigate bias and subjectivity in grading, ensuring fairness and consistency in evaluation.
Adaptability: MCQ generators can adapt question difficulty based on students' performance or specific learning objectives. Educators can customize assessments to challenge students appropriately and cater to individual learning needs.
Scalability: MCQ generators can generate large quantities of questions efficiently, making them suitable for use in both small-scale classroom assessments and large-scale standardized testing environments.
Facilitates Active Learning: MCQs can be designed to prompt critical thinking and problem-solving skills, encouraging students to engage actively with course material rather than passively memorizing information.
Diagnostic Tool: MCQ assessments can serve as diagnostic tools to gauge students' understanding of specific concepts or identify areas requiring further instruction. Educators can use assessment results to tailor instructional strategies and interventions accordingly.
Integration with Technology: MCQ generators can easily integrate with learning management systems (LMS) or online platforms, enabling seamless administration and grading of assessments. This integration enhances the efficiency and accessibility of the assessment process for students.

SYSTEM ARCHITECTURE

MCQ Composer has three main modules as shown in Figure 1.Using the Client, an instructor inputs a required number of questions and her educational text. The text is then passed through a REST API to the Multiple Choice Question (MCQ) Generator Module, which performs pre-processing and then generates and returns the required number of question–answer pairs with distractors. To achieve higher flexibility and abstraction, the models implement an interface that allows them to be easily replaced.

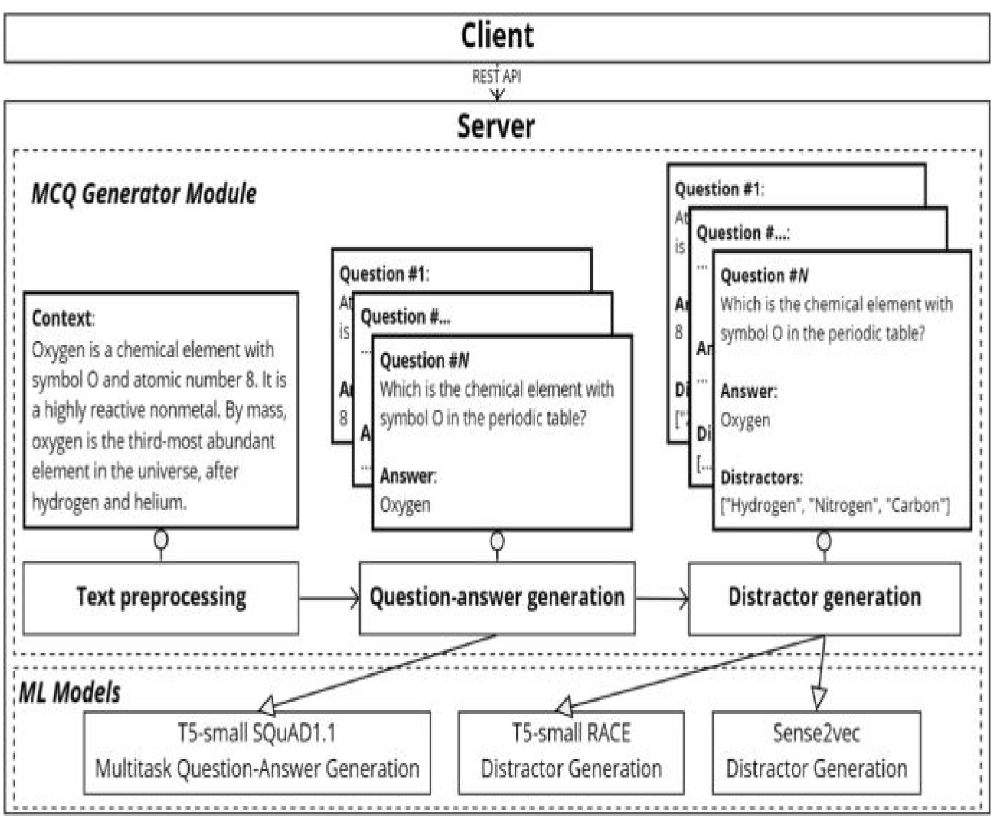

Fig.1. The general architecture of MCQ Composer

Question and Answer Generation: To create the question–answer pairs, we combined the two tasks into a single multi-task model. We fine-tuned the small version of the T5 Transformer, which has 220M parameters, and we used the SQuAD1.1 dataset , which includes 100,000 question–answer pairs. We trained the model to output the question and the answer and to accept the passage and the answer with a 30% probability for the answer to be replaced by the [MASK] token. This allows us to generate an answer for the input question by providing the token instead of the target answer. We trained the model for five epochs, and we achieved the best validation cross-entropy loss of 1.17 in the fourth epoch. We used a learning rate of 0.0001, a batch size of 16,and a source and a target maximum token lengths of 300 and 80, respectively.

For question generation, we used the same data split and evaluation scripts as in. For answer generation, we trained on the modified SQuAD1.1 Question Answering dataset as proposed in our previous work , achieving an Exact Match of 41.51 and an F1 score of 53.26 on the development set.

Distractor Generation: To create contextual distractors for the question– answer pairs, we used the RACE dataset and the small pre-trained T5 model. We provided the question, the answer, and the context as an input, and obtained three distractors separated by a [SEP] token as an output. We trained the model for five epochs, achieving a validation cross-entropy loss of 2.19. We used a learning rate of 0.0001, a batch size of 16, and a source and target maximum token lengths of 512 and 64, respectively. The first, the second, and the third distractor had BLEU1 scores of 46.37, 32.19, and 34.47, respectively. We further extended the variety of distractors with context-independent proposals, using sense2vec to generate words or multi-word phrases that are semantically similar to the answer.

User Interface: Using the user interface shown on Figure 2, the instructor can input her educational text, together with the desired number of questions to generate. Then, she can choose some of them, and potentially edit them, before using them as part of her course

DATA SETS

SQuAD

The Squad dataset, or Stanford Question Answering Dataset, is a popular benchmark dataset in the field of natural language processing (NLP) and machine reading comprehension. It was created by researchers at Stanford University. The dataset consists of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to each question is a segment of text (a "span") from the corresponding passage.

The main objective of the Squad dataset is to evaluate the ability of machine learning models to comprehend and answer questions based on a given context. This task is often referred to as Machine Reading Comprehension (MRC). The dataset is commonly used for training and evaluating models, particularly those based on deep learning architectures such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and more recently, transformer-based models like BERT, GPT, and their variants.

IMPLEMENTATION

Setup and Configuration

Set up the necessary development environment with required dependencies and libraries.

Configure the REST API for communication between modules.

Client Module:

Develop a user-friendly interface for the instructor to input educational text and the desired number of questions.

Implement functionality to send this input to the MCQ Generator Module via the REST API.

7.3 Multiple-Choice Question (MCQ) Generator Module:

Implement the REST API endpoint to receive educational text and the number of questions from the Client.

Pre-process the educational text as necessary for further processing.

Question and Answer Generation:

Fine-tune the T5 Transformer model using the SQuAD1.1 dataset to generate question–answer pairs**.**

Implement multi-task learning for question and answer generation.

Train the model with specified hyper parameters (e.g., learning rate, batch size, maximum token lengths).

Distractor Generation:

Utilize the RACE dataset and pre-trained T5 model to generate contextual distractors.

Train the model with specified hyper parameters.

Incorporate sense2vec for generating context-independent distractors.
Ensure that the interface allows for easy replacement of models through an abstracted interface.
User Interface:
Develop a user interface similar to the one shown in Figure 2.
Allow instructors to input educational text and specify the number of questions to generate.
Display the generated questions, along with distractors, for the instructor to review and potentially edit.
Provide functionality for instructors to select and save questions for use in their courses.
Testing and Validation:
Test each module individually to ensure proper functionality.
Validate the generated questions and distractors against the input text to ensure relevance and accuracy.
Conduct user testing to gather feedback on the user interface and overall usability.

Deployment
Deploy the system in the desired environment, whether local or cloud-based.
Monitor system performance and user feedback post-deployment.
Maintain and update the system as needed to address any issues or improvements.

Results & Analysis
The execution of the process will be explained clearly with the help of continuous screenshots. The whole process is uploading the context, after uploading the context, selecting how many questions to generate, then the system will automatically generate the desired number of Multiple Choice Questions.

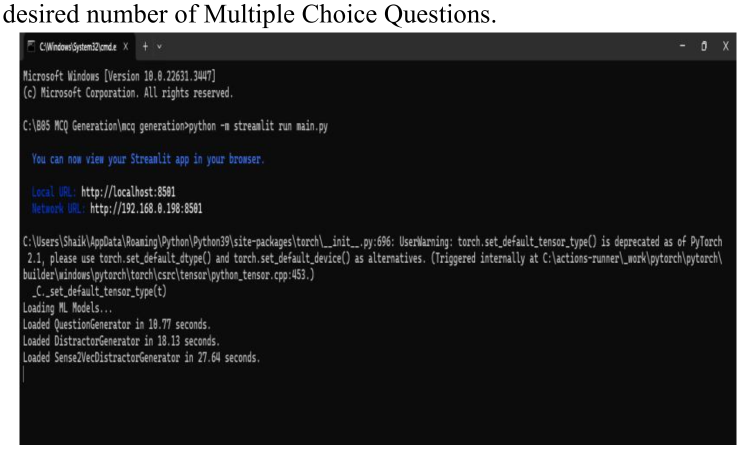

Fig.2. Launching Streamlit Application

Fig.3. Shows the Home page of the MCQ Composer consists of Abstract

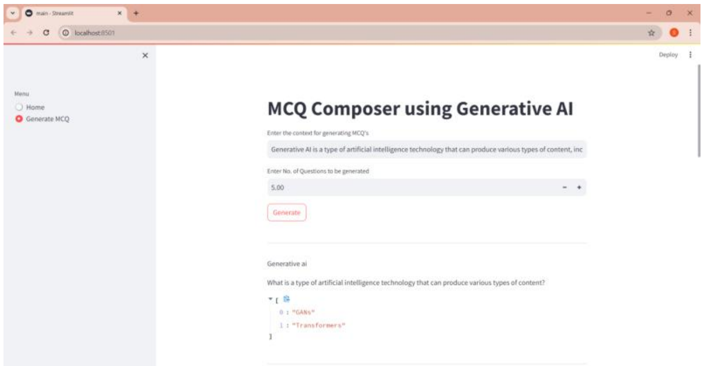

Fig.4. Shows the final result of the page after successful generation of MCQ Questions

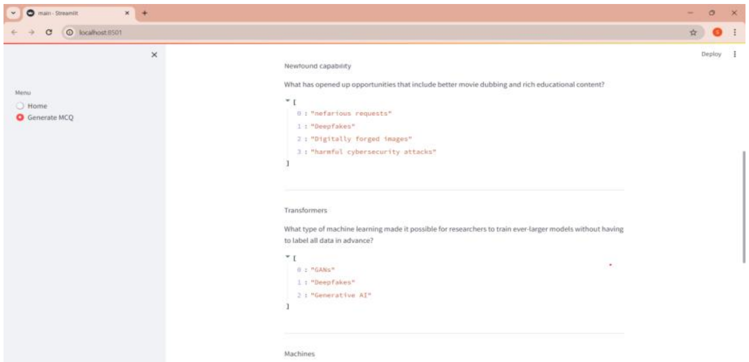

Fig.5. shows the generated MCQ Questions with Distractors

Fig.6. Shows the generated MCQ Questions With Distractors

CONCLUSION

In conclusion, MCQ Composer presents a comprehensive architecture for the automated generation of multiple-choice questions and distractors, catering to the needs of educators and instructors. Leveraging a multi-task model fine-tuned on the T5 Transformer and trained on the SQuAD1.1 dataset, Leaf efficiently produces question-answer pairs with a high degree of accuracy. By combining contextual information from the RACE dataset and sense2vec model, Leaf ensures the generation of diverse and contextually relevant distractors, enhancing the overall quality and effectiveness of the generated questions. The user-friendly interface empowers instructors to seamlessly input educational text, specify the desired number of questions, and refine or select questions as needed, streamlining the integration of generated content into their courses. Leaf's flexible architecture, with easily replaceable modules, underscores its adaptability and potential for further advancements in educational technology.

FUTURE WORK

In future work, there are several avenues to enhance the capabilities and efficiency of MCQ Composer multiple-choice question generation system. Firstly, exploration into larger transformer architectures, such as BERT or GPT, could be conducted to potentially improve question and answer generation accuracy, especially in handling more complex educational texts and diverse question types. Additionally, fine-tuning the model on a wider range of datasets beyond SQuAD1.1 could bolster its performance across various domains. Moreover, refining the distractor generation process by incorporating more sophisticated natural language understanding techniques or leveraging larger pre-trained models may lead to the creation of more plausible and contextually relevant distractors, thereby enhancing the overall quality of generated questions. Furthermore, implementing mechanisms for adaptive learning could enable the system to dynamically adjust its question generation strategies based on user feedback and performance analytics, thereby continuously improving its efficacy over time. Lastly, enhancing the user interface with features such as real-time question preview, customization options for distractor selection, and integration with learning management systems could further streamline the workflow for instructors and facilitate seamless integration of generated questions into their teaching materials.

References

1. Jacopo Amidei, Paul Piwek, and Alistair Willis. Evaluation methodologies in automatic question generation 2013-2018. In Proceedings of the 11th International Conference on Natural Language Generation, INLG ’20, pages 307–317, Tilburg University, The Netherlands, 2018. Association for Computational Linguistics.
2. Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu,Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon.UniLMv2: Pseudo-masked language models for unified language model pretraining. In Proceedings of the 37th International Conference on Machine Learning, volume 119 of ICML ’20, pages 642–652. PMLR, 2020.
3. Ho-Lam Chung, Ying-Hong Chan, and Yao-Chung Fan. A BERT-based distractor generation scheme with multi-tasking and negative answer training strategies. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4390–4400. Association for Computational Linguistics, 2020.
4. Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski,Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. Transactions of the Association for Computational Linguistics, 8:454–470, 2020.
5. Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 Reasoning Challenge. arXiv:1803.05457, 2018.
6. Peter Clark, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Dalvi Mishra,Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin, and Michael Schmitz. From ‘F’ to ‘A’ on the N.Y. Regents Science Exams: An overview of the Aristo project. AI Mag., 41(4):39–53, 2020.
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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, NAACLHLT ’19, pages 4171–4186, Minneapolis, Minnesota, USA, 2019. Association forComputational Linguistics.
8. Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS ’19, pages 13042–13054, Vancouver, British Columbia,Canada, 2019.
9. Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL ’17, pages 1342–1352, Vancouver,Canada, 2017. Association for Computational Linguistics.
10. Ayako Hoshino and Hiroshi Nakagawa. WebExperimenter for multiple-choice question generation. In Proceedings of HLT/EMNLP 2005 Interactive Demonstrations,HLT/EMNLP ’05, pages 18–19, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics.
11. Kalpesh Krishna and Mohit Iyyer. Generating question-answer hierarchies. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL ’19, pages 2321–2334, Florence, Italy, 2019. Association for Computational Linguistics.
12. Questgen. Questgen: AI powered question generator. http://questgen.ai/. Accessed: 2022-01-05.
13. Quillionz. Quillionz - world’s first AI-powered question generator. https://www.quillionz.com/. Accessed: 2022-01-05.

14. Melissa Roemmele, Deep Sidhpura, Steve DeNeefe, and Ling Tsou. AnswerQuest:A system for generating question-answer items from multi-paragraph documents.In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL ’21, pages 40–52,Online, 2021. Association for Computational Linguistics.