Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 259 (2025) 504–511

Procedia Computer Science

www.elsevier.com/locate/procedia

Sixth International Conference on Futuristic Trends in Networks and Computing Technologies (FTNCT06) held in Uttarakhand, India

Retrieval-Augmented Generation for Multiple-Choice Questions and Answers Generation

Pradeesh N[a], Remya T[b], MG Thushara[c,*], K Arun Krishna[c], Pranav V[c]

[a]*Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India*
[b]*Omnex Software Solutions Pvt.Ltd, Thoraipakkam, Chennai, India*
[c]*Department of Computer Science and Applications, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India*

Abstract

Generating a large volume of diverse questions for educational purposes can be a challenging and time-consuming task for educators. Traditional methods like manually writing questions or using templates often fall short when it comes to variety and relevance. In order to cope with these difficulties, we have examined a different approach adopting Retrieval-Augmented Generation (RAG). RAG is an enhancement technique that allows retrieving independent documents and combines it with artificial intelligence that is able to qualitatively generate contextually adequate questions. In this research, RAG was employed through the Ample LMS platform in which PDF documents were the source of generating MCQ questions. The results of the research indicate that with this method, the process of formulating the questions is relatively faster and the questions generated are of better and diverse in nature. With RAG, the burden of constructing questions is lessened for teachers and the students are provided with better interactive learning opportunities. This approach presents a better alternative that is flexible and suitable for various educational requirements, demonstrating the effectiveness of AI in the improvement of the learning and teaching processes.

Keywords: RAG; Education; Question Generation;LLM;

1. Introduction

Questions are very much essential in an educational setting. They motivate students' interest, check comprehension, and even more, guide students in the direction of new ideas. Nowadays, where there's information overload, it is increasingly important to know how to formulate the right questions. It is necessary to generate many questions which are useful for understanding content at a deep level, and there are growing means for automating the creation of

* MG Thushara
E-mail address: thusharamg@am.amrita.edu

10.1016/j.procs.2025.03.352

considerable number of questions for this purpose. Bulk question generation not only streamlines the process for educators but also enriches the learning journey by providing students with diverse and engaging materials.

However, generating high-quality questions in bulk comes with its own set of challenges.It's crucial that questions are clear, relevant, and match the learners' level of understanding. They should be varied to keep students engaged and avoid becoming repetitive. Also, the questions needs to be grammatically correct and should be in the context of concepts to effectively convey the intended meaning. Current bulk question generation approaches include rule-based methods [19, 5, 3, 24], template-based approaches [6, 14], data-driven models [23, 21], and machine learning techniques [5, 13]. Although these methods have their advantages, they usually lack flexibility and adaptability. They also find it hard to create a wide range of high-quality questions that cover different subjects and levels of difficulty.

The traditional method involves framing questions manually which leads to drawbacks. That is takes more time and is subjected to mistakes. Most of the times rule-based methods [18] work for certain types of questions, they usually can't provide the variety and depth required in today's educational settings. Similarly, template-based methods are restricted to their fixed formats, making it difficult to handle different subjects or situations. Due to this, there is an increasing need for more advanced, automated systems that can create a wider variety of questions.

Questions created manually were compared to questions created by three AI models: ChatGPT, Gemini, and Perplexity. We used the following comparison models in order to calculate the similarity: BERT, CodeBERT, and XLNet. Overall, it was observed that ChatGpt produced questionsthat were very much similar to manually produced questions. Based on our findings, we used openAI with Retrieval-Augmented Generation (RAG). This integration of retrieval of relevant information together with the generative capabilities of OpenAI models captures more accurate and meaningful responses. In applications across domains, the use of RAG through OpenAI can improve the quality of responses based on the good performance of ChatGPT with the conducted similarity analysis.

This research aims at addressing issues related to the question generation process and improving its efficiency. Using AMPLE LMS[16], a learning management system from Omnex Software Solutions, we conducted an experiment where questions and answers were generated from PDF documents. We found this setting ideal for testing more complex solutions for bulk question generation. In this investigation, we look at the possibility of Retrieval-Augmented Generation (RAG)[7] as a potential solution. RAG integrates information retrieval and generative models to make possible the generation of questions about diverse, yet relevant, factual content. Based on a large corpus of educational material, RAG avoids the limitations inherent in conventional methods, hence allowing the process to be highly flexible, efficient, and scalable.

Generating bulk questions with the help of RAG saves teachers' efforts to work out more effective ways of teaching in favor of students. Adaptation of different subjects and learning levels will enable personalized practice materials on individual needs.With education increasingly becoming digital platforms like Ample LMS, there's a need for automation of question generation. RAG offers a viable approach that facilitates the proficient generation of high-caliber questions spanning multiple subjects, thereby transforming the manner in which educators and students engage with educational resources.

The key contribution of the paper are as follows

- **Integration of RAG with AMPLE LMS**: Our research demonstrates how RAG can be seamlessly integrated into the AMPLE LMS to automate the generation of MCQs from educational content, specifically PDFs. This integration facilitates a more efficient workflow for educators, reducing the time and effort required to create assessments.
- **Enhanced Question Quality**: The use of RAG allows for the generation of high-quality questions that are not only relevant but also diverse in nature. By leveraging a broad corpus of educational materials, our approach mitigates the limitations often seen in traditional question generation methods, which can lead to repetitive or irrelevant questions.
- **Implications for Educators and Learners**: This integration provides significant benefits for both educators and learners. Educators can focus more on teaching and less on question preparation, while learners benefit from a richer set of assessment tools that cater to various learning styles and levels.

This paper is structured as follows: The section 2 gives an overview of current methods for generating questions, pointing out their limitations and the need for more advanced approaches like Retrieval-Augmented Generation

(RAG). In the section 3, the proposed operational flow and working on Ample LMS, by clearing giving insights on how RAG is applied to create multiple-choice questions from PDF documents. The section 4 discusses the findings of teh proposed approach, comparing RAG's performance to traditional methods in terms of accuracy, relevance, and diversity of questions. Finally, the section 5 Conclusion summarizes our key findings and discusses how RAG can impact the automation of question generation, along with suggestions for future research and improvements.

2. Related Works

Historically, creating a large number of questions has involved various approaches, each with its own pros and cons. Rule-based methods [19, 5, 3, 24], which use specific guidelines to generate questions from text, provide a great deal of control over the format of the questions. However, they can lead to repetitive and predictable outcomes because they stick closely to the rules and lack flexibility. Likewise, template-based methods [6, 14] use fixed formats, where details from the text are plugged into pre-existing question templates. While these techniques will ensure consistency, they do not always have creativity or flexibility for many different kinds of topics or question types. Most of the times [11] questions and answers are not in the desired format.

Another perspective is methods based on data [23, 21, 8], which supplement the process of machine learning [22, 9] using trained models on large amounts of data, thus giving more options. Such methods are expected to generate questions on a greater range of texts and topics. However, still they may produce irrelevant or broad questions, particularly in case of insufficient or poor quality training data. This inconsistency can be a challenge to the teachers who require proper and relevant questions to evaluate the students appropriately [12].

Recently, there has been a new approach that involves the use of large language models [17] such as GPT-3 and GPT-4 in generating questions. Such models are capable of generating more conversational and realistic-sounding questions, but there are drawbacks as well. This means one of the greater concerns is about the phenomenon of "hallucinations" that are very believable, yet not factual, due to lack or out of date information.

This is where Retrieval-Augmented Generation (RAG) [7] enters the scene. RAG combines the best features of generative models which incorporate information retrieval wherein data from outside sources is used for generation of questions that are accurate as well as contextual. Unlike the models that work by only using existing knowledge, RAG improves the question generation process in ensuring the data are up-to-date and relevant. It reduces the chances of generating incorrect or irrelevant questions, making it a more dependable tool for educators.

Previous researches such as works like REALM[25] and Atlas [2] have proven that a combination of retrieval with generation, improves the accuracy in AI-generated contents. However, these systems still come across many challenges, especially with complicated questions that demand several rounds of reasoning or when there are ambiguous and conflicting information. Here, we discuss the possibilities of further improving RAG on these counts so that it benefits the quality of the artificially generated questions for learning purposes.

This is the main advantage of RAG when it replaces the previously utilized strategies. It offers a promising way to address the problems of bulk questioning. With advancement in the AI technology, this could potentially change the way educators approach the creation and use of learning materials, for efficiency and adaptability [15, 10, 20], and accessible to a very diverse group of learners.

3. Methodology

The process for generating multiple-choice questions from a PDF document (process flow shown in figure 1) begins with extracting the document's content using PyPDFLoader. We observed that a text/transcript of at least 100 words was sufficient to generate 5–7 questions [1] of reasonable quality. This utility reads the entire PDF file and loads its text into the program. Since handling large sections of text as a whole can be inefficient and may result in a loss of context during processing, the text is then split into smaller, manageable chunks using the CharacterTextSplitter function. This splitter breaks the document into overlapping segments of 1000 characters each, with an overlap of 200 characters between adjacent segments to maintain context across sections. These chunks of text serve as input for later steps in the pipeline, helping ensure that content remains coherent when used for question generation.

Once the document has been split into chunks, the process moves into embedding these text chunks into a vector store using FAISS (Facebook AI Similarity Search) [4]. Embeddings are dense vector characterizations of the text that capture the semantic meaning of the given text content. In this case, FAISS was used to build a vector store, which allows efficient storage and similarity search on the text embeddings. In case a vector store of the topic has already been built and exists on the local machine (saved from a previous run), it's used in order to save on time. If it does not exist, a vector store is built using the text chunks and made available on the local machine for future access. Vector store is beneficial because it helps the system to pull out the most relevant part of the PDF content for analysis during the question generation phase, based on the semantic similarity of the input prompt.

After the text is embedded, the system then applies OpenAI's GPT-4 model to generate questions. The GPT-4 model is given a fixed structure prompt to generate a set of multiple questions on the content from the PDF. The structure also specifies the need for four answer options per question of which one has to be right. The language model initiates the query to the FAISS vector store through a conversational retrieval chain to obtain content from the vectorized records of the PDF. The process allows for generation of questions that are contextually relevant to the document throughout the retrieval process.The system has a global memory of generated questions which makes the interactions more advanced and enables to remember previous generated question sessions.

In order to ensure the non-repetition and creativity of the questions produced by the system, a duplicate-checking system based on semantic similarity is used. Once after a batch of questions is produced, the system analyzes each new question against questions which were already generated and stored in memory. This assessment is achieved through checking embeddings of the current question and prior questions, supported by cosine similarity calculations. If this similarity value is greater than a certain level (0.85), the new question will be regarded as duplicate and discarded. Only unique questions are kept and they are then stored in memory. As a result, the system is able to generate questions which are not the same, thus avoiding repetitive occurrence of questions. After that, the final unique questions are either populated on web interface for analysis purpose or sent as a JSON response, with proper measures taken on questions generation and invalid JSON formatting related issues.

This methodology has allowed us to create questions that are not only correct but also diverse and relevant to the learning content. The RAG module was added to the Ample LMS platform to ensure that all the processes including content extraction and question creation were smooth, scalable, and effective in bulk question generation.

RAG-based systems face several challenges that can limit their effectiveness. They require significant computational resources, which can slow down the process and make them less suitable for environments with limited capabilities. The quality of the questions they generate depends heavily on the relevance and accuracy of the information they retrieve. If the sources are outdated, irrelevant, or poorly written, the results will likely be flawed too. Additionally, these systems struggle with unclear or conflicting information, which can lead to confusing questions that may not meet the needs of learners. Overcoming these issues requires better reasoning and more reliable information handling.

4. Results and Discussion

Retrieval-Augmented Generation (RAG) was used in question generation to create multiple-choice questions for our study. Ample LMS was employed for content extraction and question generation as well. For evaluating the AI-generated content, a study was carried out using a data set taken from a Java question paper. A teacher constructed 25

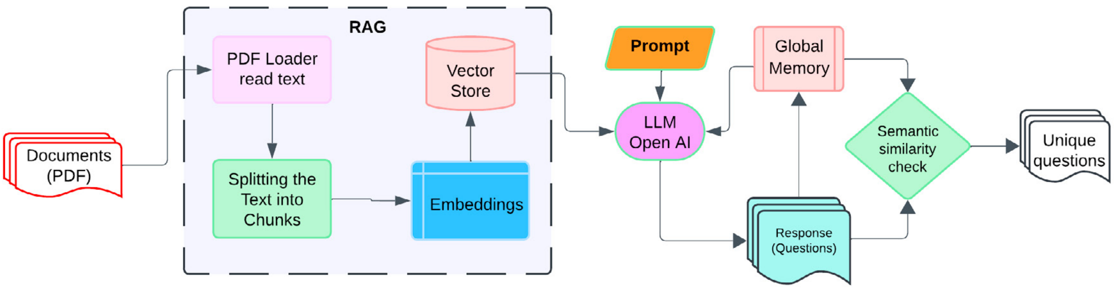


Fig. 1. Process Flow of Proposed Model

questions over five topics (Encapsulation, Class, Data Hiding, Acess Modifiers and Getters & Setters) in which the topics were the basis of this study.This provided us with a clearer framework for our comparison.

Using the same content from Ample LMS, we generated 75 additional questions through three AI models: ChatGPT, Gemini, and Perplexity. The questions produced by the AI were made utilizing RAG which went into the LMS to retrieve relevant information and applied generative representations to make contextually appropriate questions.

Table 1. Table showing evaluation of the performance of different models (C- ChatGPT, G- Gemini, P-Perplexity)in Question Generation.

Concept	XLNET			BERT			Code BERT		
	C	G	P	C	G	P	C	G	P
Encapsulation	**0.92**	0.878	0.888	**0.76**	0.738	0.733	**0.984**	0.968	0.968
Class	**0.927**	0.924	0.909	**0.782**	0.575	0.608	**0.978**	0.953	0.971
Data Hiding	**0.886**	0.882	0.876	0.739	**0.779**	0.719	**0.992**	0.981	0.972
Access Modifiers	**0.898**	0.882	0.873	**0.933**	0.838	0.774	**0.976**	0.975	0.974
Getters and Setters	0.894	**0.905**	0.895	**0.934**	0.854	0.803	**0.991**	0.985	0.979

The quality of these questions was evaluated based on relevance, grammatical accuracy, and diversity (refer Table 1). Among the three models, ChatGPT stood out by producing the most accurate and natural-sounding questions,closely aligning with the teacher-generated ones. Its capacity to grasp the subtleties of Java concepts and produce questions that sound natural made it the best performer in our analysis.

Integrating RAG into Ample LMS created a smooth process for generating questions. Educators could easily upload documents and automatically create a wide range of questions with little effort, greatly lowering their manual workload. This system ensures that the questions are varied and relevant to the context, providing an efficient way for educators to develop high-quality learning materials.

Table 1 showing that ChatGPT achieved a similarity score of **0.92**, surpassing Gemini (0.88) and Perplexity (0.76). ChatGPT's strength lies in its ability to create more nuanced and well-structured questions, with fewer grammatical and semantic errors. Gemini showed promise, but frequently produced questions that were less diversified and sometimes repetitious. Perplexity, on the other hand, was useful but lacked clarity and relevancy. The LMS integration

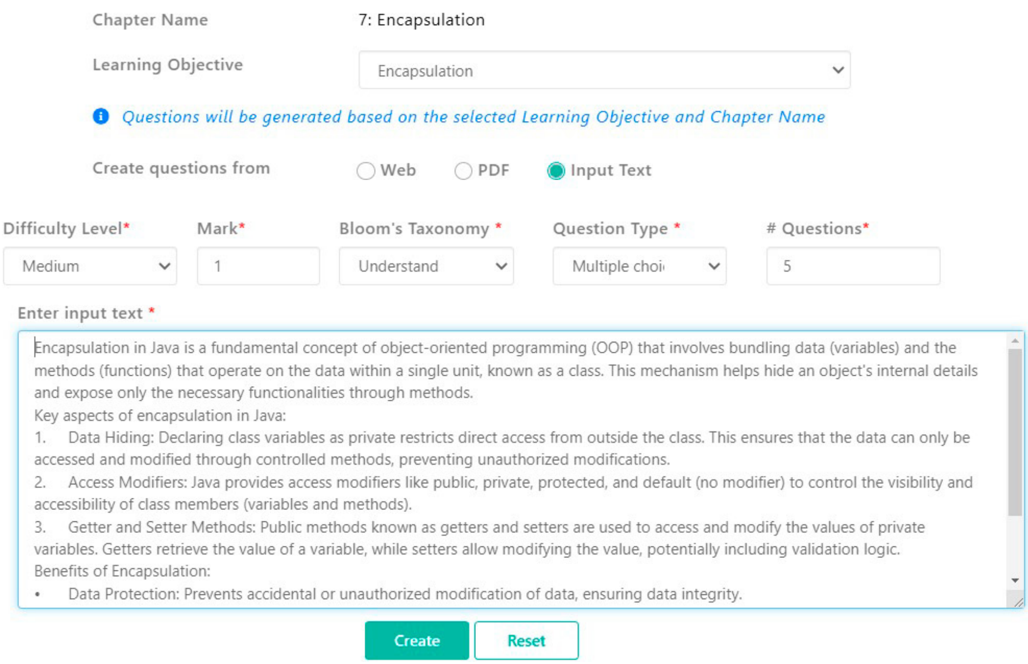
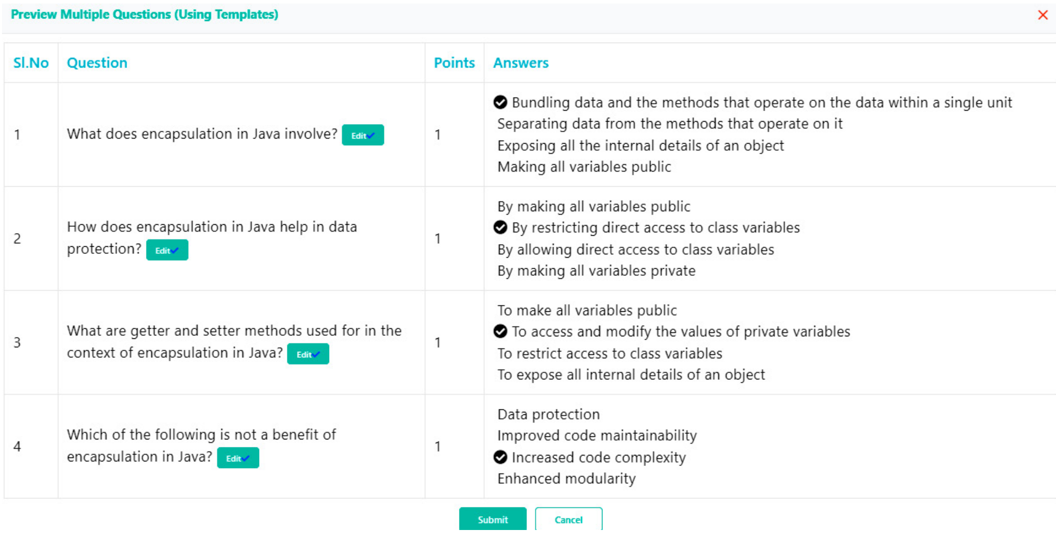


Fig. 2. Ample LMS Output Screen showing how input is given for QP Generation

made the process even simpler, allowing educators to upload a document, text, or link to a webpage and automatically generate a set of multiple-choice questions. With RAG, the platform quickly retrieved the most relevant content from the uploaded materials and used it to create diverse questions. Figures 2 and 3 show the user-friendly interface of the LMS, demonstrating how easy it is to input documents and generate questions with minimal effort. This smooth integration with Ample LMS not only sped up content creation but also ensured the questions were of high quality and variety.

According to our research, RAG can successfully automate the creation of instructional materials, particularly when paired with ChatGPT. By using AI to create questions, teachers may devote more of their time to instruction and provide students with more interesting and varied content.


Preview Multiple Questions (Using Templates)

Sl.No	Question	Points	Answers
1	What does encapsulation in Java involve? Edit	1	Bundling data and the methods that operate on the data within a single unit Separating data from the methods that operate on it Exposing all the internal details of an object Making all variables public
2	How does encapsulation in Java help in data protection? Edit	1	By making all variables public By restricting direct access to class variables By allowing direct access to class variables By making all variables private
3	What are getter and setter methods used for in the context of encapsulation in Java? Edit	1	To make all variables public To access and modify the values of private variables To restrict access to class variables To expose all internal details of an object
4	Which of the following is not a benefit of encapsulation in Java? Edit	1	Data protection Improved code maintainability Increased code complexity Enhanced modularity

Submit Cancel

Fig. 3. Ample LMS Sample Output Screen on the QP Generated

In addition to the above-positive outcomes of this research, several novel aspects enhance the process of question generation. Overlapping text chunks in methodology ensure that the questions generated maintain context and therefore have more coherent and meaningful assessments. This is because of better contextual understanding that makes a student more deeply engaged with the material. In addition, Retrieval Augmented Generation (RAG) within the ample LMS drastically reduces educator time and effort in formulating quality questions. As a result, this streamlines the process of generating these questions and frees up additional time for the teachers for teaching and student engagement, thus enhancing the learning experience. To adequately assess multiple-choice questions generated, a panel of educators does a thorough review of the sampling set, with a high emphasis on the quality of distractors associated with every question. Each distractor was rated carefully against the three critical criteria: relevance, clarity, and potential to mislead. This structured evaluation is essential for effective distractor design, as it guarantees that the options presented are not only appropriate but also contribute meaningfully to the learning objectives by appropriately challenging students.. The average scores obtained from these ratings gave us some quantitative feedback about the quality of distractors created in general. Apart from this formal review, we have a feedback facility in our LMS, which enables the learners to raise any confusion about the questions or the appropriateness of the options offered. This system enables the students to raise feedback for each question, which instructors can later check for quality. This process not only makes the questions better in quality but also creates a dynamic learning environment where students' input is appreciated and used to make the whole process better. This way of both educator assessment and learner feedback ensures that the generation of questions is dynamic and responsive to the needs of both instructors and students

5. Conclusion

This study highlights the great potential of Retrieval-Augmented Generation (RAG) for automating the creation of multiple-choice questions in education. By integrating RAG into Ample LMS, we made the usually time-consuming task of generating questions easier, reducing the workload for teachers. The system was indeed able to generate grammatically correct, varied, and contextually relevant questions from education-related documents

Our experiments showed that, of the models tested, ChatGPT consistently came up with questions that best matched those developed by teachers in natural language and structure, making it a dependable tool for educators seeking quick but effective development of quality content. In sum, RAG can be easily adapted to various topics and learning settings; it is scalable and will keep yielding improvement in teaching and student learning.

5.1. Future Direction

In the future, RAG research has opened exciting opportunities on several fronts regarding question generation. Improvement is needed in how retrieval and comprehension of the content are performed by the system to construct more accurate and contextually relevant questions, especially on complex subjects. Moreover, the system being expanded to create any sort of question, such as an open-ended or essay question, makes it more versatile in an educational assessment. The second very promising path refers to personalization; in that, it adjusts the difficulty level of questions according to each student's performance, hence making possible a personalized learning experience. This should make the generated questions compatible with what educators want to happen, and this could come about as a combination of human oversight and automation.

In our future directions, we are aiming to create a more general model that increases the flexibility and utility of our question generation system over a wider variety of computer science topics. A complete dataset from a computer science curriculum including a wide range of subjects and resources will be used in this model. We train our huge language model using this varied dataset in order to build a reliable system that can provide excellent multiple-choice questions (MCQs) that are pertinent and contextually appropriate for every discipline of computer science. By using examples from other areas of computer science, this method will overcome the present constraints on the diversity of datasets. This, in our opinion, will support our findings and increase the general efficacy of the question creation procedure, ultimately providing educators and students with a more comprehensive and captivating educational experience.

Finally, integrating RAG with real-time analytics in LMS platforms could foster a more interactive learning environment, where questions are continuously refined based on student feedback. These improvements could significantly enhance education by making learning more engaging and personalized.

References

[1] Ackerman, R., Balyan, R., 2023. Automatic multilingual question generation for health data using llms, in: International Conference on AI-generated Content, Springer. pp. 1–11.

[2] Basaran, A.E., Güresir, A., Knoch, H., Vychopen, M., Güresir, E., Wach, J., 2024. Beyond traditional prognostics: Integrating rag-enhanced atlasgpt and chatgpt 4.0 into aneurysmal subarachnoid hemorrhage outcome prediction .

[3] Dhole, K.D., Manning, C.D., 2020. Syn-qg: Syntactic and shallow semantic rules for question generation. arXiv preprint arXiv:2004.08694 .

[4] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.E., Lomeli, M., Hosseini, L., Jégou, H., 2024. The faiss library. arXiv preprint arXiv:2401.08281 .

[5] Du, X., Shao, J., Cardie, C., 2017. Learning to ask: Neural question generation for reading comprehension. arXiv preprint arXiv:1705.00106 .

[6] Fabbri, A.R., Ng, P., Wang, Z., Nallapati, R., Xiang, B., 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. arXiv preprint arXiv:2004.11892 .

[7] Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.S., Li, Q., 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 6491–6501.

[8] Gabajiwala, E., Mehta, P., Singh, R., Koshy, R., 2022. Quiz maker: Automatic quiz generation from text using nlp, in: Singh, P.K., Wierzchoń, S.T., Chhabra, J.K., Tanwar, S. (Eds.), Futuristic Trends in Networks and Computing Technologies, Springer Nature Singapore, Singapore. pp. 523–533.

[9] Gangadharan, V., Gupta, D., Amritha, L., Athira, T., 2020. Paraphrase detection using deep neural network based word embedding techniques, in: 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), IEEE. pp. 517–521.

[10] Gutjahr, G., Chandrashekar, P., Nair, M.G., Haridas, M., Nedungadi, P., 2020. Comparison of hidden markov models and the fast algorithm for feature-aware knowledge tracing, in: Machine Learning for Predictive Analysis: Proceedings of ICTIS 2020. Springer, pp. 269–276.

[11] Hrvacević, L., Cincović, J., Milaković, A., Jocović, V., Matvejev, V., Drašković, D., 2024. Development of a web system with an automated question generator based on large language models, in: 2024 11th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN), pp. 1–6. doi:10.1109/IcETRAN62308.2024.10645106.

[12] Joseph, N., Pradeesh, N., Chatterjee, S., Bijlani, K., 2017. A novel approach for group formation in collaborative learning using learner preferences, in: 2017 international conference on advances in computing, communications and informatics (ICACCI), IEEE. pp. 1564–1568.

[13] Kumar, A.P., Nayak, A., K, M.S., Chaitanya, Ghosh, K., 2024. A novel framework for the generation of multiple choice question stems using semantic and machine-learning techniques. International Journal of Artificial Intelligence in Education 34, 332–375.

[14] Kusuma, S.F., Siahaan, D.O., Fatichah, C., 2022. Automatic question generation with various difficulty levels based on knowledge ontology using a query template. Knowledge-Based Systems 249, 108906.

[15] Nedungadi, P., Menon, R., Gutjahr, G., Raman, R., 2024. Evaluating holistic education and digital learning model for advancing sdg4: a longitudinal mixed-effects modeling approach. Cogent Social Sciences 10, 2299134.

[16] Omnex, . E-learning. URL: https://www.omnex.com/e-learning.

[17] Pawar, C.S., Makwana, A., 2022. Comparison of bert-base and gpt-3 for marathi text classification, in: Singh, P.K., Wierzchoń, S.T., Chhabra, J.K., Tanwar, S. (Eds.), Futuristic Trends in Networks and Computing Technologies, Springer Nature Singapore, Singapore. pp. 563–574.

[18] Rajesh, S., Rao, V.V., Thushara, M., 2024. Comprehensive investigation of code assessment tools in programming courses, in: 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), IEEE. pp. 1–6.

[19] Rao, P.R., Jhawar, T.N., Kachave, Y.A., Hirlekar, V., 2022. Generating qa from rule-based algorithms, in: 2022 International Conference on Electronics and Renewable Systems (ICEARS), IEEE. pp. 1697–1703.

[20] Scaria, N., Dharani Chenna, S., Subramani, D., 2024. Automated educational question generation at different bloom's skill levels using large language models: Strategies and evaluation, in: International Conference on Artificial Intelligence in Education, Springer. pp. 165–179.

[21] Subramanian, S., Wang, T., Yuan, X., Zhang, S., Bengio, Y., Trischler, A., 2017. Neural models for key phrase detection and question generation. arXiv preprint arXiv:1706.04560 .

[22] Tiwari, S.P., Prasad, S., Thushara, M., 2023. Machine learning for translating pseudocode to python: A comprehensive review, in: 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE. pp. 274–280.

[23] Wang, Z., Lan, A.S., Nie, W., Waters, A.E., Grimaldi, P.J., Baraniuk, R.G., 2018. Qg-net: a data-driven question generation model for educational content, in: Proceedings of the fifth annual ACM conference on learning at scale, pp. 1–10.

[24] Zhang, R., Guo, J., Chen, L., Fan, Y., Cheng, X., 2021. A review on question generation from natural language text. ACM Transactions on Information Systems (TOIS) 40, 1–43.

[25] Zhu, Y., Ren, C., Xie, S., Liu, S., Ji, H., Wang, Z., Sun, T., He, L., Li, Z., Zhu, X., et al., 2024. Realm: Rag-driven enhancement of multimodal electronic health records analysis via large language models. arXiv preprint arXiv:2402.07016 .