IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511119

Syllabus Driven Question Answering Model using AI

Narasimha Dixit[1], Anushree Joshi[2], Nida F Ashrafi[3], Shifa A Solad[4], Surabhi A Raichur[5]

Assistant Professor, Computer Science and Engineering (AI & ML), KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India[1]

Student, Computer Science and Engineering (AI & ML), KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India[2]

Student, Computer Science and Engineering (AI & ML), KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India[3]

Student, Computer Science and Engineering (AI & ML), KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India[4]

Student, Computer Science and Engineering (AI & ML), KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India[5]

Abstract: Students preparing for exams often have to go through textbooks, university syllabi, lecture notes, and other study materials to find the information they need. This process can be time-consuming, especially when the required content is scattered across different documents. Although general-purpose large language models can answer academic questions, they may sometimes provide information that goes beyond the prescribed syllabus. To overcome this limitation, this paper presents a syllabus-driven question answering model based on Retrieval-Augmented Generation (RAG) [5]. The system first retrieves relevant information from the university syllabus and study materials uploaded by students, which is then used by the language model to generate answers. It uses the all-MiniLM-L6-v2 embedding model for processing academic text, ChromaDB for storing and retrieving information, and Qwen through the Groq Cloud API for generating responses. Along with question answering, the platform offers features such as revision, quizzes, flashcards, viva preparation, study planning, personal study-material integration, bookmarks, learning activity tracking, and content export. The system was tested using 30 syllabus-based questions from three engineering subjects. The RAG-based approach achieved an average semantic cosine similarity of 0.7937 compared to the reference answers, whereas the direct LLM baseline achieved 0.7247. This shows that the RAG-based system produced answers with greater semantic similarity to the reference answers.

Keywords: Retrieval-Augmented Generation, Syllabus-Driven Question Answering, Semantic Retrieval, Large Language Models, Academic Question Answering, ChromaDB.

I. INTRODUCTION

Artificial intelligence has changed the way students access academic information. Question answering systems use natural language processing to allow users to ask questions in everyday language and receive relevant responses. With the development of large language models, these systems can now handle a wide range of queries. However, ensuring that the responses are relevant, reliable and appropriate to the given context remains a challenge. [1]

Engineering students regularly refer to university syllabi, textbooks, lecture notes and presentations while preparing for examinations. Finding information related to a particular subject or syllabus topic often requires going through several documents. This can take considerable time, especially when students are looking for explanations related to specific course topics. Although general-purpose large language models can answer academic questions, their responses may sometimes include information beyond the prescribed syllabus. Retrieving relevant academic content before generating an answer can help reduce this problem.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.1511119

Previous studies have explored the use of course-specific information in academic question answering systems. Fernandez et al. introduced SyllabusQA, a dataset containing 5,078 open-ended question-answer pairs collected from 63 course syllabi across 36 majors. Their study explored language-model-based approaches and Retrieval-Augmented Generation (RAG) for answering syllabus-related questions. It also highlighted that similarity between generated answers and reference answers does not necessarily guarantee factual correctness. This is an important factor to consider when evaluating academic question answering systems. [2]

Wang et al. developed an intelligent course question answering system that combines natural language processing, semantic analysis, BERT-based retrieval, named entity recognition and knowledge graph traversal. The system was designed to handle factual, conceptual and procedural questions from students. However, the study also identified difficulties in handling ambiguous questions and procedural queries. [3]

Considering these findings, this paper presents the Syllabus Driven Question Answering Model using AI, an academic question answering system based on Retrieval-Augmented Generation [5]. The system retrieves relevant information from university syllabus and study-material collections and provides it as context to a language model for generating answers. It uses the all-MiniLM-L6-v2 embedding model for representing academic text, ChromaDB for vector storage and retrieval, and Qwen through the Groq Cloud API for response generation.

In addition to question answering, the platform provides features to support different aspects of exam preparation. These include revision, quizzes, flashcards, viva preparation, suggested questions and study planning. Students can also upload their own study materials, bookmark useful responses, export generated content and track their learning activities within the application.

The present work extends the authors' earlier mini project, which introduced a syllabus-focused RAG pipeline using semantic embeddings and ChromaDB. The current system builds on that implementation by supporting multiple study-material formats and introducing additional academic assistance features. The evaluation approach has also been updated to compare the proposed RAG system with direct language-model generation. [4]

The system was evaluated using 30 questions selected from three engineering subjects. The generated answers were compared with reference answers using semantic cosine similarity. The proposed RAG approach achieved an average similarity score of 0.7937, while the direct LLM baseline achieved 0.7247. The results show that the RAG-based approach produced answers with greater semantic similarity to the reference answers. However, these scores represent semantic correspondence and should not be considered a direct measure of factual accuracy.

II. LITERATURE REVIEW

A. AI-Based Question Answering Systems

Singh and Rasool presented a survey of AI-based question answering systems, covering their different architectures, applications, retrieval methods and evaluation techniques. The study explains how these systems have developed from traditional approaches to methods based on deep learning and large language models. It also highlights challenges such as contextual understanding, explainability, scalability and information retrieval. [1]

B. Syllabus-Grounded Question Answering

Fernandez, Scarlatos, and Lan introduced SyllabusQA, a dataset designed for answering questions based on university course syllabi. It contains 5,078 open-ended question-answer pairs collected from 63 course syllabi across 36 majors. Their study evaluated language-model prompting and Retrieval-Augmented Generation approaches. The findings showed that textual similarity scores may not always reflect factual correctness, highlighting the importance of checking generated answers against their source material. [2]

C. Intelligent Course Question Answering

Wang et al. developed an intelligent course question answering system that combines natural language processing, semantic analysis, BERT-based answer retrieval, named entity recognition and knowledge graph traversal. It was designed to answer factual, conceptual and procedural questions from students. Their evaluation reported an overall accuracy of 88.1%. However, the study also identified challenges in handling ambiguous questions and procedural explanations, as well as limitations related to knowledge graph scalability. [3]

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511119

D. Previous Syllabus-Driven QA Work

The authors' earlier mini project introduced a syllabus-driven question answering system using semantic embeddings, ChromaDB and Retrieval-Augmented Generation. It processed syllabus content and generated embeddings using SentenceTransformer models. ChromaDB was used to retrieve relevant information, which was then passed to a locally deployed Gemma3 model through Ollama to generate responses. This work established the initial retrieval and generation pipeline. It also identified future improvements, including the addition of complete textbooks, better retrieval, conversational interaction and support for syllabus updates. [4]

E. Research Gap

The reviewed studies explore different approaches to academic question answering. Singh and Rasool focused on general question answering architectures, while Fernandez et al. examined question answering based on university syllabi. Wang et al. explored the use of NLP techniques and knowledge graphs for educational queries. The authors' earlier project established a basic syllabus-driven RAG system.

However, the earlier implementation offered limited support for student-uploaded study materials and additional exam preparation activities. The present work builds on this system by maintaining separate collections for prescribed syllabus content and student-provided materials. It also introduces features such as revision, quizzes, flashcards, viva preparation, study planning and bookmarking. In addition, the generated responses are evaluated against reference answers by comparing the RAG approach with a direct LLM baseline.

III. PROPOSED METHODOLOGY

The proposed system uses Retrieval-Augmented Generation (RAG) to generate answers using relevant academic content [5]. When a student submits a question, the system searches its stored academic materials for information related to the query. The retrieved content is then provided to the language model along with the original question, helping it generate responses based on the relevant course material.

The methodology involves several stages, including document processing, semantic embedding, vector storage, content retrieval, context preparation and answer generation. It also includes additional modules to support students in their academic preparation.

A. System Architecture

The process begins when a student submits an academic question through the Streamlit interface. The question is converted into an embedding using the all-MiniLM-L6-v2 model. This embedding is then used by ChromaDB to search the stored academic content and retrieve document chunks relevant to the question.

The retrieved chunks are combined with the original question to prepare the context for the language model. The academic content is maintained in two separate ChromaDB collections. The vtu_syllabus collection stores syllabus and reference materials, while user_documents contains study materials uploaded by students. Keeping these collections separate helps distinguish prescribed academic content from student-provided materials.

The complete processing flow is illustrated in Fig. 1.

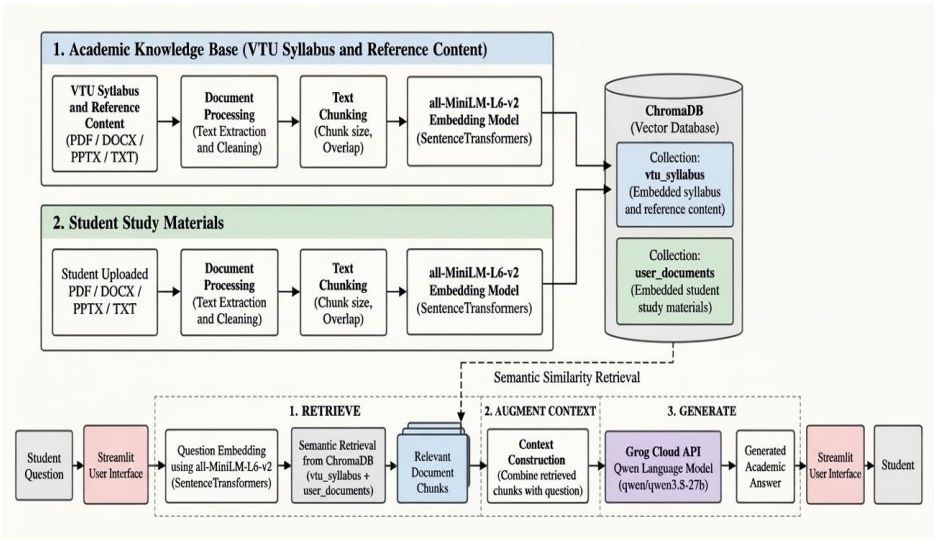


Fig. 1. System Architecture of the Syllabus Driven Question Answering Model using AI.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511119

B. Academic Document Processing

The system supports academic documents in PDF, DOCX, PPTX and TXT formats. Text is extracted from these files using format-specific libraries. PyMuPDF is used to process PDF files, while python-docx and python-pptx are used for DOCX and PPTX files. TXT files are read directly.

After extraction, the text is cleaned and divided into smaller chunks. This makes it easier to retrieve the required information without sending the entire document to the language model. Each chunk contains approximately 500 units of text, with an overlap of 80 between consecutive chunks. This overlap helps retain shared information when a topic continues from one chunk to the next.

C. Semantic Embedding and Vector Storage

The extracted and processed text chunks are converted into numerical embeddings using the all-MiniLM-L6-v2 sentence embedding model [6]. These embeddings capture the meaning of the text and help the system find related information even when the question uses different words from the original document.

ChromaDB is used to store the embeddings along with their corresponding text and metadata. The vtu_syllabus collection contains syllabus and reference materials, while the user_documents collection stores materials uploaded by students. These collections serve as the source of information during the retrieval process.

D. Semantic Retrieval and Context Construction

When a student submits a question, the all-MiniLM-L6-v2 model converts it into an embedding. ChromaDB compares this embedding with the stored document embeddings and retrieves the chunks that are most semantically similar to the question. This helps identify relevant information even when the question does not use the exact words found in the source documents.

Cosine similarity is used to measure the similarity between the question embedding and a document embedding:

$$\text{Cosine Similarity } (A, B) = A \cdot B / \|A\| \|B\|$$

Here, A represents the question embedding and B represents a document embedding. A higher value indicates greater semantic similarity between the two vectors.

The retrieved chunks are then combined with the student's question to prepare the context for answer generation. This information is passed to the language model along with the original question so that the response is based on the retrieved academic content.

E. Answer Generation Using the Language Model

The prepared context and the student's question are sent to the Qwen language model through the Groq Cloud API. Using the supplied academic content, Qwen generates a response based on the student's query and selected response requirements. The system supports different types of responses, including detailed explanations and revision-oriented answers.

After generation, the response is returned to the Streamlit interface and displayed to the student. The retrieved information serves as the academic context for generating the final answer.

F. Academic Learning-Support Features

Apart from question answering, the platform provides additional tools to help students prepare for examinations. These include revision material, syllabus-based quizzes, flashcards, viva preparation and suggested questions. A study planner helps students organize their preparation, while personal study-material integration allows them to use their own documents. Students can also bookmark useful content and export generated materials in PDF and TXT formats.

The system also records user activity within the application. It considers events such as mouse interactions, keyboard activity, scrolling, page visibility and changes in application focus. Periods of inactivity are excluded when calculating active learning time. These records provide an indication of activity within the application and are used as a proxy for learning activity. They should not be treated as a direct or validated measure of a student's concentration or cognitive engagement.

IV. SYSTEM IMPLEMENTATION

The proposed system was developed as an academic assistance platform using Python and Streamlit. It brings together document processing, semantic embedding, vector-based retrieval and language model-based response generation. The application also includes separate modules to help students with different academic activities.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511119

A. Application Interface

Streamlit is used to build the interface through which students access the application. It allows them to submit academic questions, choose the type of assistance they need, upload study materials and view previously generated or saved content. The interface includes separate sections for Chat, Q&A, Revision, Quiz, Flashcards, Viva, Study Planner and Suggested Questions. It also provides My Notes for managing personal study materials, Bookmarks for saving useful responses and a section to view learning activity. These features allow students to use the platform for both question answering and examination preparation.

B. Document Ingestion and Processing

The document ingestion module handles academic files before they are added to the retrieval system. The application supports PDF, DOCX, PPTX and TXT formats. PyMuPDF is used to extract text from PDF files, while python-docx and python-pptx are used to process DOCX and PPTX files.

Once the text is extracted, it is cleaned and divided into smaller chunks before generating embeddings. Source information is also retained with the processed chunks so that the retrieved content can be used during answer generation.

The application uses two separate ChromaDB collections. The vtu_syllabus collection stores syllabus and reference materials, while user_documents stores files uploaded by students. This separation helps keep prescribed academic content distinct from personal study materials.

C. Embedding and Vector Database Implementation

The Sentence Transformers library provides the all-MiniLM-L6-v2 model, which converts document chunks into numerical embeddings [6]. The same model is used to process student questions, ensuring that both stored documents and incoming queries are represented in the same semantic space.

ChromaDB stores these embeddings along with their corresponding document content. When a student submits a question, the application converts it into an embedding and uses it to retrieve related document chunks from the stored collections. The retrieved information is then used to prepare the context for the language model.

D. LLM Integration and Response Generation

The application connects to the Qwen language model through the Groq Cloud API. The model configured for response generation is qwen/qwen3.8-27b.

When a student submits a question, the application first retrieves relevant academic information from ChromaDB. The retrieved content is combined with the original question and sent to the Groq Cloud API. Qwen uses this context to generate the requested response. The generated answer is then returned to the Streamlit interface and displayed to the student.

E. Academic Assistance Modules

The application includes separate modules for different academic activities. The Chat module supports conversational interaction, while Q&A generates structured answers based on the selected response style.

The Revision module provides concise study material, and Quiz generates syllabus-based multiple-choice questions and evaluates student responses. Flashcards provides question-and-answer pairs for revision, while Viva helps students prepare for oral examinations.

The Study Planner helps students organize their preparation based on subjects, examination requirements and available study time. Suggested Questions provides additional questions related to the selected topic. My Notes allows students to use their personal study materials, while Bookmarks helps them save useful responses for later reference. Generated content can also be exported in PDF and TXT formats.

F. Learning Activity Tracking

The application includes an activity tracker that records student interactions while using the platform. It considers activities such as mouse movements, clicks, keyboard interactions, scrolling, touch interactions, page visibility and changes in application focus.

Periods of inactivity are excluded from the recorded active learning time, and tracking is paused when the page is no longer active. The collected data indicates activity within the application and is used as a proxy for learning activity. It does not provide a validated measurement of a student's concentration or cognitive engagement.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511119

V. EXPERIMENTAL SETUP AND EVALUATION

The proposed system was evaluated by comparing its generated answers with reference answers prepared from the prescribed academic material. The evaluation focuses on semantic similarity and compares the proposed Retrieval-Augmented Generation (RAG) approach with a direct Large Language Model (LLM) baseline.

A. Evaluation Dataset

The evaluation was carried out using 30 syllabus-based questions from three engineering subjects: Software Engineering and Project Management, Computer Networks, and Research Methodology and IPR. Ten questions were selected from the first module of each subject.

The questions were based on topics covered in the respective modules. Reference answers were prepared using the prescribed syllabus and academic reference materials. The same set of questions and reference answers was used in both evaluation settings to ensure a consistent comparison.

B. Evaluation Settings

Two different response-generation settings were considered for evaluation.

In the first setting, the questions were processed through the proposed RAG pipeline. The system retrieved relevant academic content from the vector database and provided it to the language model as context before generating an answer.

In the second setting, the same questions were given directly to the language model without retrieving any academic content. This served as the direct LLM baseline.

The results from both settings were compared to understand how the use of retrieved academic information affects semantic similarity.

C. Semantic Similarity Evaluation

The generated answers and their corresponding reference answers were converted into vector representations using the all-MiniLM-L6-v2 embedding model [6]. Cosine similarity was then calculated between each pair of embeddings to measure how closely their meanings matched.

The cosine similarity is calculated using the following formula:

$$\text{Cosine Similarity } (A, B) = A \,.\, B / \|A\| \, \|B\|$$

Here, A represents the embedding of the generated answer, while B represents the embedding of the reference answer. A higher value indicates greater semantic similarity between the two representations.

The RAG-based system achieved an average semantic cosine similarity of 0.7937, while the direct LLM baseline achieved 0.7247. The difference between the two scores was 0.0690, representing an approximate relative improvement of 9.52% with the RAG approach.

These results indicate better semantic alignment between the generated answers and reference answers when retrieved academic context was used. However, semantic similarity scores alone do not establish the factual accuracy of the generated responses.

D. Subject-Wise Evaluation

The RAG results were also grouped according to the three selected engineering subjects to examine how the system performed across different subject areas.

TABLE I SUBJECT-WISE SEMANTIC SIMILARITY OF THE RAG SYSTEM

Subject	RAG Semantic Similarity
Software Engineering and Project Management	0.7806
Computer Networks	0.8098
Research Methodology and IPR	0.7906

Computer Networks achieved the highest average similarity score of 0.8098, followed by Research Methodology and IPR with 0.7906. Software Engineering and Project Management recorded an average score of 0.7806.

The results show some variation in similarity scores across the three subjects. This may be due to differences in subject terminology, the available reference materials and the concepts covered by the selected questions.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511119

E. Evaluation Procedure

Each question was processed separately using both the RAG and direct LLM settings. The generated answers were converted into embeddings using the same all-MiniLM-L6-v2 model and compared with their corresponding reference answers using cosine similarity.

The individual similarity scores were then combined to calculate the average score for each evaluation setting. The RAG results were also grouped by subject to obtain the subject-wise average scores.

Both evaluation settings used the same questions, reference answers, embedding model and similarity measure. This ensured that the comparison was carried out under consistent conditions.

VI. RESULTS AND DISCUSSION

The evaluation results show how the answers generated using the proposed Retrieval-Augmented Generation (RAG) approach compare with those generated directly by the language model. The comparison includes the overall similarity scores as well as the results for individual subjects.

A. Overall Evaluation Results

For the 30 evaluation questions, the proposed RAG system achieved an average semantic cosine similarity of 0.7937, while the direct LLM baseline recorded an average score of 0.7247.

The difference between the two scores was 0.0690, representing an approximate relative improvement of 9.52% with the RAG approach. This shows that providing retrieved academic content helped generate answers that were more semantically similar to the reference answers.

TABLE II OVERALL SEMANTIC SIMILARITY COMPARISON

Evaluation Setting	Average Cosine Similarity
Direct LLM	0.7247
RAG	0.7937
Absolute Difference	0.0690

The results suggest that retrieving relevant syllabus content and study materials before generating answers improved their semantic similarity to the reference answers. However, cosine similarity only measures how closely the meanings match and does not directly confirm the factual correctness of the responses.

B. Subject-Wise Results

The subject-wise evaluation shows differences in the average similarity scores achieved by the RAG system. Computer Networks recorded the highest score of 0.8098, followed by Research Methodology and IPR with 0.7906. Software Engineering and Project Management recorded an average score of 0.7806.

TABLE III SUBJECT-WISE SEMANTIC SIMILARITY RESULTS

Subject	Average Cosine Similarity
Software Engineering and Project Management	0.7806
Computer Networks	0.8098
Research Methodology and IPR	0.7906

The scores show some differences in semantic similarity across the selected subjects. These variations may be due to differences in subject terminology, concepts, reference materials and the nature of the questions.

C. Discussion

The RAG approach achieved a higher average semantic similarity score than the direct LLM baseline. This suggests that providing relevant academic content as context helped the language model generate answers that were closer in meaning to the prepared reference answers.

The subject-wise results also showed variations across the three engineering subjects. Computer Networks recorded the highest average score, while the other two subjects had slightly lower scores. This shows that the similarity between generated answers and reference answers was not the same across all subject areas.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511119

These findings should be considered in the context of the evaluation. The experiment was conducted using 30 questions from three engineering subjects, making it a limited evaluation. Cosine similarity measures how closely the meanings of two texts match, but it cannot independently confirm whether every statement in a generated answer is factually correct. Overall, the results suggest that using retrieved academic content improves semantic alignment between generated and reference answers. However, further evaluation using more questions and additional subjects would help understand how the system performs across a wider range of academic topics.

VII. LIMITATIONS AND FUTURE SCOPE

A. Limitations

The proposed system was evaluated using 30 questions from three engineering subjects. Although this provides an initial understanding of its semantic similarity performance, the evaluation does not cover all subjects, modules, question types or academic materials included in a university curriculum.

The evaluation relies on semantic similarity to compare generated answers with reference answers. While this helps determine how closely the answers match in meaning, it does not directly verify the factual correctness of every statement. Therefore, the reported scores should be understood within the scope of the experiment rather than treated as direct accuracy measurements.

The quality of the responses also depends on the academic materials available to the system. Incomplete, unclear or poorly structured documents may affect the retrieval process and the answers generated from them. The system may also face difficulties with questions that require information not available in the stored documents.
The learning activity component records interactions within the application. However, these records only indicate activity on the platform and cannot be used as a validated measure of student concentration, understanding or cognitive engagement.

B. Future Scope

The system can be improved by adding complete textbooks, more reference materials and updated syllabus documents. Support for regular syllabus updates could also help keep the academic content current.
Future evaluations could include a larger set of questions covering more subjects, semesters, modules and question types. Additional evaluation methods could also be introduced to check factual correctness, retrieval quality and the usefulness of generated answers.

The retrieval process can be further improved by experimenting with different embedding models, retrieval methods and ranking techniques. Better handling of information spread across multiple document sections could also help the system answer more complex academic questions.

The platform could also be extended with more personalized learning features. Revision materials and study recommendations could be adjusted based on students' previous interactions and academic requirements. These improvements could help develop the system into a more complete syllabus-aware academic learning assistant.

VIII. CONCLUSION

This paper presented a syllabus-driven question answering model based on Retrieval-Augmented Generation (RAG). The system retrieves relevant information from university syllabi and study materials uploaded by students and uses it as context for generating answers through Qwen via the Groq Cloud API. It uses the all-MiniLM-L6-v2 embedding model and ChromaDB for semantic retrieval. Along with question answering, the platform provides features such as revision, quizzes, flashcards, viva preparation, study planning, personal study-material integration, bookmarks, learning activity tracking and content export.

The system was evaluated using 30 questions from three engineering subjects. The RAG approach achieved an average semantic cosine similarity of 0.7937, compared with 0.7247 for the direct LLM baseline. Among the three subjects, Computer Networks recorded the highest average score of 0.8098. These results show that providing relevant academic content as context helped generate answers that were closer in meaning to the reference answers.

The evaluation was limited to three subjects and 30 questions. Since cosine similarity measures how closely the meanings of two texts match, the scores alone cannot confirm the factual correctness of every generated response. Further

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511119

evaluation using larger datasets, additional academic subjects and methods that assess factual correctness and retrieval quality would help provide a better understanding of the system's performance.

Overall, the proposed system provides a foundation for syllabus-based question answering and academic assistance through a single platform. Future improvements can focus on expanding the available academic content, improving retrieval methods and providing more personalized learning support.

REFERENCES

[1]. N. Singh and A. Rasool, "AI Enabled Question Answering Systems: A Survey," in *Proceedings of the 2025 IEEE World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS)*, 2025, doi: 10.1109/WorldSUAS66815.2025.11199298.

[2]. N. Fernandez, A. Scarlatos, and A. Lan, "SyllabusQA: A Course Logistics Question Answering Dataset," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, 2024, pp. 10344–10369, doi: 10.18653/v1/2024.acl-long.557.

[3]. Y. Wang, M. Yang, Y. Chen, and M. Yu, "Research on Intelligent Course Q&A Systems Based on NLP Models, Knowledge Graphs, and Deep Learning Methods," in *Proceedings of the 2024 4th International Symposium on Big Data and Artificial Intelligence (ISBDAI)*, 2024, doi: 10.1145/3723366.3723370.

[4]. N. Dixit, A. Joshi, N. F. Ashrafi, S. A. Solad, and S. A. Raichur, "Syllabus Driven Question Answering Model Using AI," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 07, no. 12, Dec. 2025, doi: 10.56726/IRJMETS86698.

[5]. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[6]. Sentence-Transformers, "sentence-transformers/all-MiniLM-L6-v2," *Hugging Face Model Repository*. [Online]. Available: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

 This work is licensed under a Creative Commons Attribution 4.0 International License