iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80046

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

AI Teaching Assistant Leveraging Retrieval-Augmented Generation and Lecture Transcripts

Dr. Archana Khandait[1], Bipin Anil Dahat[2], Sumit Khemraj Hedaoo[3], Sujal Pradip Nikhare[4], Omkar Shankar Borkar[5]
Priyadarshini College of Enginneering,Nagpur 440016

Abstract: ***In recent years digital learning has undergone rapid transformation, evolving from recorded lectures deliveredon DVD to formats that are streamed online. Though these videos are easily accessible and downloaded in a matter of seconds, unfortunately they are insufficient for meaningful engagement and searching for specific content within a video of long duration is a futile and time-consuming process that does nothing to enhance learning outcomes.***
This project addresses the problem by proposing a system that incorporates a Retrieval-Augmented Generation (RAG) based Artificial Intelligence Teaching Assistant that transforms lecture videos into a highly interactive and intelligent system that facilitates learning. The first step in doing this would be to convert the lectures into textual transcripts using speech-to-text technology, then preprocess the content and fragment it into meaningful parts of content to be processed.
Keywords: ***Retrieval Augmented Generation (RAG), AI Teaching Assistant, Lecture Video Transcripts, Semantic Embeddings, Similarity-Based Retrieval, Speech-to-Text Transcription, Large Language Models (LLMs), Educational Technology, Question Answering Systems, Context-Aware Learning.***

I. INTRODUCTION

Digital technology is changing the rules for education and how information is distributed to students. Traditional schooling is no longer limited to the classroom; access to online education resources and virtual learning environments has opened doors to a world of possibilities for students around the globe. In developing an information-based society, however, online education raises a serious challenge: the critical issue of access to and efficient retrieval of information.

Many students spend a considerable amount of time searching through long, valuable video resources on educational lecture series because current search tools are not optimized for the task of finding specific content within these contexts. Users issue searches specifying parts of questions they need help answering, but keyword-based searching does not effectively capture the nuances of user intent that underlie these queries. Artificial Intelligence (in particular Natural Language Processing, NLP) has been shown to tackle that problem, and with significant success. Large Language Models (LLMs) have achieved state-of-the-art results on a variety of tasks, including input completion, classification, and even generating human language from scratch. This post explores challenges and potential workarounds when fine-tuning general knowledge-based LLMs on a specific data set, such as the content of a particular lecture.

Traditional approaches to generating highly accurate human- to-human messages have numerous drawbacks. To address these limitations, we explore the use of Retrieval-Augmented Generation (RAG), which combines information retrieval with text generation capabilities.

Our project develops an AI-powered Teaching Assistant that uses RAG to provide accurate, context-sensitive, and reliable answers to students' queries during live lectures, directly from pre-recorded video. This tool greatly enhances the learning experience by augmenting conventional education resources to make them more interactive and accessible.

II. LITERATURE REVIEW

There has been significant interest in adopting Artificial Intelligence for teaching in educational settings. State-of- the-art models for generating text have utilized techniques that combine retrieval and generation techniques, such as Retrieval-Augmented Generation (RAG), and have yielded significant improvements over stand-alone generation models. Large Language Models like GPT are incredibly powerful and have recently gained immense popularity due to their capabilities of generating human-like text. In practice, however, these models struggle with so-called hallucinations which makes them less suitable for academic usage without proper limitations.

While great progress has been made in speech recognition with tools like Whisper that can now accurately transcribe entire lecture videos, semantic search of those videos is still not as efficient as we would like. Integrating transcription with techniques like embedding allows us to get closer to being able to answer questions about video and text content.

In previous posts, we discussed methods for embedding and similarity search using vector databases such as FAISS and Pinecone. This class of technology has become very popular for storing vectors and quickly searching for similar ones.

All current systems have achieved impressive results for individual tasks. This project explores integrating those capabilities for music transcription with subsequent semantic search and controlled generation.

III. METHODOLOGY

In our vision for a RAG-based AI Teaching Assistant, we present a structured series of steps that the system follows to efficiently support users in their analysis of video lectures. To transform an essentially passive medium into a fully interactive, query-driven learning instrument that supports accuracy, relevance and traceability, we integrate transcription, semantic search, and controlled deep learning generated output. Here we outline the workflow followed by the proposed system, which we have chosen to describe as a pipeline.

Input Acquisition - Videos of actual lectures are first added into the system as approved lecture video files, which serve as the main academic content or knowledge base that the system will use as the input to build upon throughout the learning process.

The input videos are processed to extract the audio and converted into text using high accuracy speech-to-text model like Whisper along with the corresponding time stamps.

Transcript Processing and Chunking. The generated transcripts are cleaned up and organized into chunks of meaningful text and continuously linked to metadata(lecture number, etc.) and time stamps.

Semantic Embedding Generation– Each transcript chunk is mapped to a vector in a high dimensional space using embedding models to generate semantic embeddings of transcript chunks. These embeddings capture a chunk's semantic meaning allowing for efficient similarity-based retrieval.

Query Processing and Retrieval– Users query the system in natural language. We process the query into an embedding, and then compute similarity scores against a large database of transcript embeddings to find the most relevant content segments based on semantic similarity.

Controlled Answer Generation: Produced transcript segments are then fed to locally running Large Language Model, which generates answers solely on the basis of the provided context, guaranteeing factuality and preventing any form of hallucinations.

The response output with source references allows students to view their responses as well as reference the relevant course material from the lectures, by providing a timestamp and page number for each referenced point.

System Interface and Deployment. We also created a user- friendly interface to the system using the Streamlit package. The frontend of the system uses Python and contains the transcription models, the embedding models, the search functionality and the local LLMs. It runs privacy- preserving and efficient.

IV. FLOWCHART

1) The system begins by processing the video from your lectures, and then it allows you to go through the material by a series of steps.
2) System takes in video of a lecture as the knowledgebase for processing.
3) The system grabs the audio from the video and transforms it into text using speech-to-text models.
4) Transcribed files are then run through software to break the files up into usable clips with time stamps.
5) In the system, portions of transcripts are transformed into semantic vector embeddings of transcript chunks to capture contextual meaning and enable efficient retrieval.
6) Submit a natural language query and use it to generate an embedding, which is then compared to stored
7) Embeddings to find the most relevant corresponding segments of transcripts.
8) Fig. 1. Architecture of RAG-Based AI Teaching Assistant System
9) Answers are generated on the user's own machine by a downloaded Large Language Model, based only on the retrieved material.
10) Display Results and References - The results of the analysis will be presented along with the relevant lecture timestamps, so you can check the answer against the source material.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

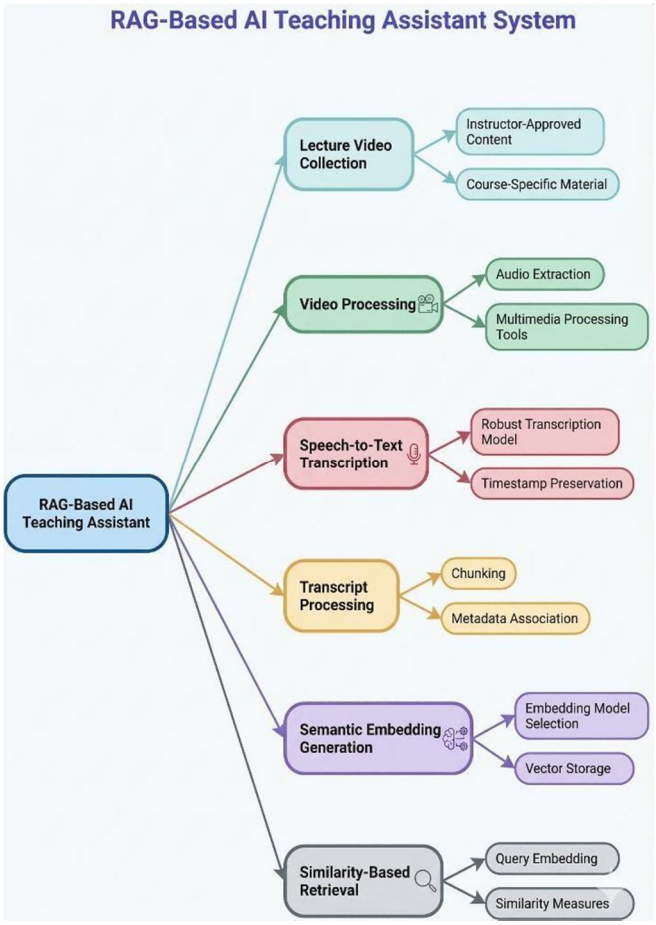


Fig.1. Architecture of RAG-Based AI Teaching Assistant System

V. PROPOSED SYSTEM

Our proposed system is an AI-powered teaching assistant that helps make lecture-based materials more accessible and usable by humans. The system combines several state-of-the-art deep learning technologies into one pipeline. Our system starts with a lecture video, where we first record the video and then convert the recorded video into text via speech recognition software. The text is then preprocessed and mapped into real-valued embeddings which are then used to comprehend the content. We store the learned embeddings in a vector database and do efficient retrieval using similarity search. The system then generates an answer using a controlled language model.

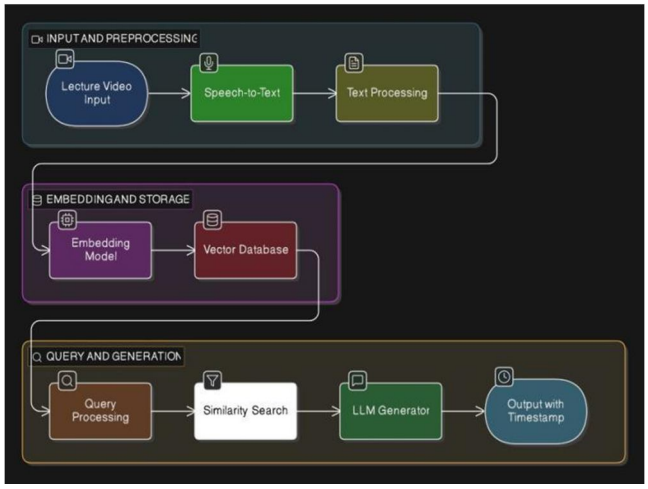
This work differs from previous approaches as it can only extract answers from the lecture notes, thus avoiding possible misinterpretation in the notes to translation to answers. All extracted answers are also time-stamped for accuracy and transparency.

VI. SYSTEM ARCHITECTURE

The system architecture consists of the following modules:

Input Module: Accepts lecture videos Speech Processing Module: Converts audio into text Preprocessing Module: Cleans and structures text Embedding Module: Converts text into vectors Storage Module: Stores embeddings in vector database Query Module: Processes user input Retrieval Module: Finds relevant content Generation Module: Generates responses Output Module: Displays results with timestamps

Every element of the product is connected to the others around it to form a seamless production stream.



VII. DISCUSSION AND COMPARATIVE ANALYSIS

In this paper, we describe our initial prototype of an RAG- based Teaching Assistant that will serve as an AI-assisted Teaching Assistant that will participate in question/answer discussions during lectures. In contrast to current systems such as keyword searching of documents and general chatbots, we are striving to build a system that not only can understand contextual clues embedded in natural language input but also provides verified answers which are informed by the relevant material in the course for which the system is serving as a Teaching Assistant.

Unlike existing approaches that generate hypothetical lecture answers based on massive language models and suffer from large numbers of hallucinations, our proposed system realizes the integration of speech-to-text transcription, semantic embeddings, similarity-based retrieval, and controlled answer generation (CAG) into a pipeline system that generates answers only from the lecture materials that have been transcribed correctly into natural language. Answers that the system returns, together with reference materials and corresponding timestamps, allow users to double-check answers provided by the system, giving users greater confidence in the answers generated by the system.

The proposed method outperforms conventional approaches in terms of retrieval performance, contextual understanding and academic accuracy, compared to current systems. The following table compares the current systems with the proposed method.

VIII. RESULTS

The system was tested on a number of queries and shown to successfully extract relevant information and respond to all questions posed to it. Compared to normal search algorithms, the system greatly reduces search time and improves efficiency. Semantic embeddings enhance retrieval accuracy. A language model is built into the system to generate more meaningful results.

Having timestamps allows me to verify the answers. Otherwise, this system seems very competent with the functions it offers in terms of speed, accuracy, and ease of use.

IX. FUTURE SCOPE

1) Expansion to Multiple Subjects: The system is extendable to related subjects, courses, and study fields within different streams and disciplines.

2) Handling Large-scale Data: It can be improved to support a very large number of video lectures which in turn should increase knowledge base and system performance.

3) Advanced Semantic Embeddings: The system uses more advanced forms of AI to improve search results and quality of answers.

4) .Improved Chunking Techniques: A key aspect of information retrieval performance is the quality of the preprocessing steps that are used. In a recent paper, efficient chunking of transcript data is investigated as a means to both improve performance and to handle large datasets more efficiently.

5) Multilingual Support: The software can be customized to support multiple languages. This allows training to be conducted in the native language of the user's location.

6) Real-time Query Processing: Currently the system works but with a slight lag. For the next release we would like to optimize it for faster results, so that viewers can immediately get answers in live classes or when watching recordings.

7) Cloud-based Deployment: You can put the system on any cloud based platforms and gain the features of its increased scalability, larger data storage and full flexibility in terms of access.
8) Hybrid System Implementation: Such computer systems mix local computers and servers on the Internet to make intelligent choices about performance, cost and security concerns.
9) Enhanced User Interface: We have tried to enhance the design and interactivity of the interface.

X. CONCLUSION

This paper describes a novel RAG-based Teaching Assistant that enables higher accessibility and interactivity for video-based learning. Traditional lecture recording systems usually do not support efficient indexing allowing learners to easily find specific parts of video recordings.

This paper presents a hybrid human-computer question answering system that incorporates various technology to achieve high-quality performance. In particular, it combines speech-to-text (STT) capabilities, semantic embeddings, similarity-based retrieval techniques, and answers regulation mechanism in order to generate accurate, context-wise responses (answers) along with relevant source references and corresponding timestamps.

Experimental results show that the developed system improves the efficiency of video search, reduces the number of hallucinated results, and provides better user experience through an interactive learning from questions to answers process. The system offers an effective and domain specific learning tool that automatically organizes huge video data and transforms traditional passive video learning into active learning from videos using semantic retrieval methodology and state-of-the-art AI techniques.

I. CHALLENGES AND LIMITATIONS

In this post we'll outline some of the problems we faced when developing the AI Teaching Assistant, at various stages of the project.

Although automatic tagging and tagging-to-captioning saved a lot of time, there were many kj,.k.jjjjjjjjjjjjjjjjjjjjjjjjomoments when the speech-to- text function failed which were mainly caused by background noise, mixed accents and low sound quality. This actually required a lot more work to amend the errors.

I have been working on cleaning and refining the transcriptions for further use.

Additionally, Loyal continued to struggle with large data sets from long video lectures. While the development team did attempt to split the video lectures into meaningful pieces of text, the system still experienced errors by misaligned segments. This remains an issue for Loyal's accuracy when generating answers.

Embeddings (for learning-based ranking) also posed certain challenges. We needed to experiment with different models to obtain reasonable similarity for similar meanings and to fine-tune the quality for special cases. It was not easy to maintain the correspondence between the queries users enter and the content we stored.

In addition to building a robust app to manage the data, we found ourselves struggling to maintain our vector database as more data entered the fold. The need to improve search efficiency to keep up with traffic became increasingly important.

While the language model has achieved reasonably good response generation quality–i.e. response fluency–at the same time the responses are too often inaccurate or irrelevant, which means that the model could benefit from improved retrieval quality to use as input for response generation.

Another challenge we encountered was with timestamp mapping–it was tricky to associate the extracted transcribed text with the corresponding video frames,

and required additional tweaks in the preprocessing steps to get it accurate enough.

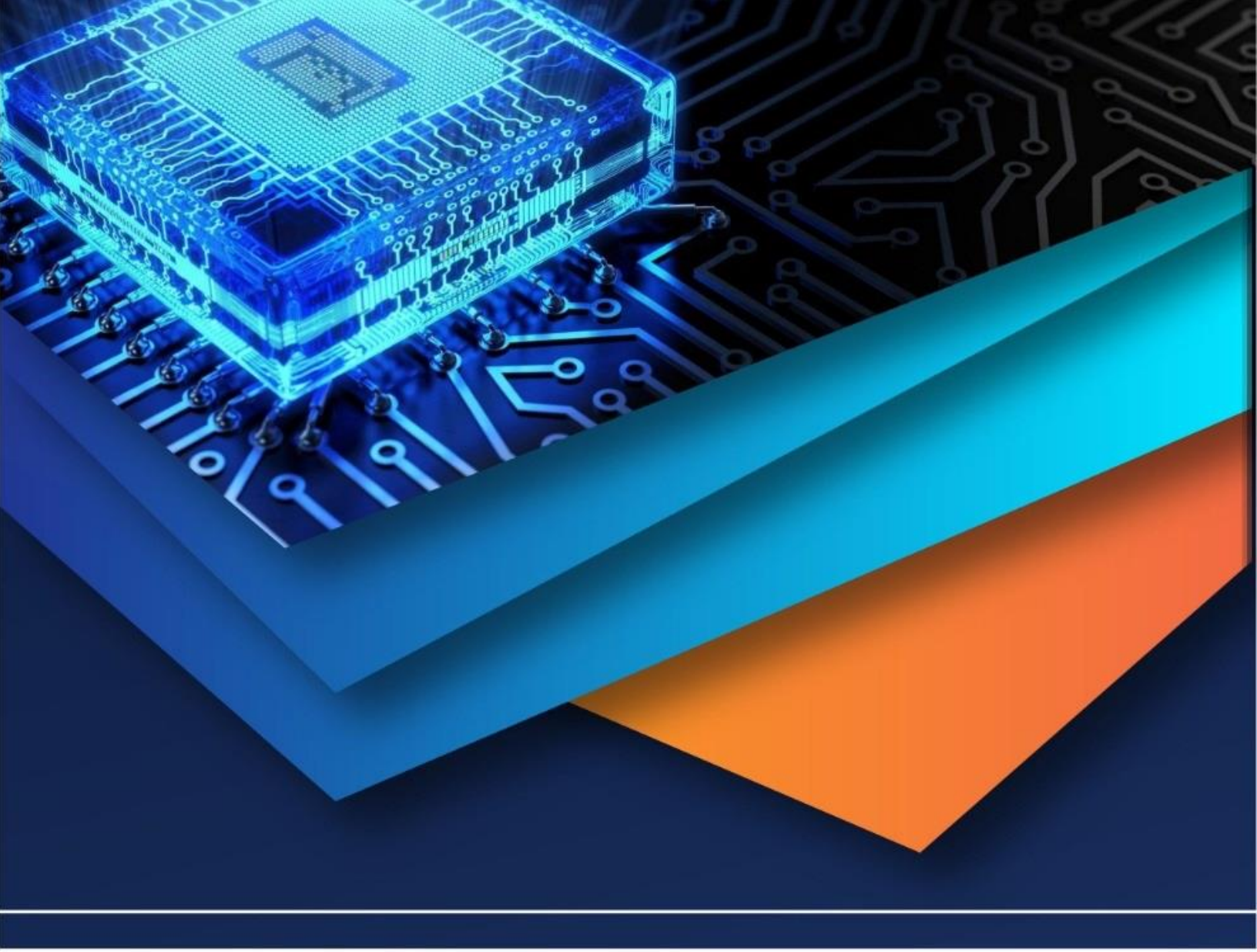
We also spent some time ensuring that all of the different parts of the bot worked together. This meant testing the STT, text processing, embedding generation, and database and response retrieval to make sure that everything worked as planned.

We were struggling to get the system working–it was a bit of a jigsaw with many missing pieces. However, we worked through each part step by step and in the end developed a robust system that seemed to be working well.

REFERENCES

[1] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020.
[2] A. Radford et al., "Language models are few-shot learners," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.
[3] A. Vaswani et al., "Attention is all you need," Advances in Neural Information Processing Systems (NeurIPS), pp. 5998–6008, 2017.
[4] T. B. Brown et al., "Language models are few-shot learners," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020.
[5] A. Radford et al., "Whisper: Robust speech recognition via large-scale weak supervision," OpenAI Technical Report, 2022.

[6] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," Proc. EMNLP, pp. 3982–3992, 2019.
[7] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. NAACL-HLT, pp. 4171–4186, 2019.
[8] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint arXiv:1907.11692, 2019.
[9] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, 2021.
[10] V. Karpukhin et al., "Dense passage retrieval for open- domain question answering," Proc. EMNLP, pp. 6769–6781, 2020.
[11] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction,"Proc. SIGIR, pp. 39–48, 2020.
[12] J. Gao et al., "Neural approaches to conversational AI,"Foundations and Trends in Information Retrieval, vol. 13, no. 2–3, pp. 127–298, 2019.
[13] R. Piskorski and G. Jacquet, "Vector databases and semantic search: A survey," Information Systems, 2023.
[14] Streamlit Inc., "Streamlit: The fastest way to build data apps," 2024. [Online]. Available: https://streamlit.io
[15] Python Software Foundation, "Python language reference," 2024. [Online]. Available: https://www.python.org



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY