

LectureLM: Intelligent Lecture Understanding and Interactive Learning Model

Harshwardhan Singh¹, Kushal², Md Sadain Siddiqui³, Krish Sharma⁴

¹²³⁴ Student, Department of Computer Science & Engineering

Bhagwan Parshuram Institute of Technology, New Delhi, India

¹harshwardhansingh034@gmail.com | ²kkushal9891@gmail.com | ³mdsadainsiddiqui973@gmail.com | ⁴krish.sharma8105@gmail.com

Abstract — We propose LectureLM, an AI-powered web application for understanding and interactively studying lecture videos. The system automatically retrieves transcripts from YouTube via the YouTube Data API, falling back to GPT-4o-generated synthetic transcripts when captions are unavailable, and uses them to produce structured notes in JSON format. Unlike prior summarization tools, LectureLM features multiple learning modes: comprehensive full-text notes, scanning for key ideas, an adaptive multiple-choice quiz engine (1–20 questions), a Retrieval-Augmented Generation (RAG)-grounded Q&A chat interface, and a plain-language concept simplifier. A lightweight session-memory module tracks weak and strong topics across a user's session and dynamically adjusts difficulty. The system is implemented in Python using Streamlit for the user interface and the OpenAI API for all AI-powered features. We describe the architecture, individual component design, and proposed evaluation methods including ROUGE metrics, quiz accuracy, and direct human assessment. Future work includes multimodal inputs and persistent user profiles.

Keywords: Large Language Models (LLM), Transcript Processing, Artificial Intelligence (AI), Session Memory, Text Summarization, Retrieval-Augmented Generation (RAG)

I. INTRODUCTION

The rapid growth of online learning platforms presents students with a significant challenge: navigating and retaining knowledge from hours of lecture content. Platforms such as YouTube host vast libraries of educational videos, far exceeding what any learner can efficiently watch and review. Traditional note-taking, though valuable, is time-consuming, and AI-based tools that distill content into study material hold considerable promise for improving learning outcomes. However, existing tools offer no adaptive element—they receive no feedback, cannot track topics, and do not respond to how the learner is progressing.

We address this gap by developing LectureLM, an AI-powered learning environment supporting self-testing, targeted clarification, and adaptive review. Recent advances in Large Language Models (LLMs) make such a system feasible: GPT-4o generates both high-quality summaries and human-quality quiz questions, yet prior prototypes lack the feedback loops necessary to adapt to individual learner data. Automatic Speech Recognition (ASR) pipelines described in Section II provide the transcription foundation for the entire system.

II. LITERATURE REVIEW

Retkowski et al. (2023) demonstrated high accuracy on end-to-end neural ASR architectures but noted that fluency errors in written transcripts remain a persistent challenge [1]. Benedetto et al. (2022) showed that AI-generated summaries help learners grasp content efficiently, though many existing systems fail to exploit the full potential of the underlying transcripts [2].

Ilampirayan et al. (2021) proposed a system that cleans transcripts and feeds them to NLP models (TF-IDF, TextRank, transformer-based approaches) to generate bullet-point notes [3]. Gupta and Sen (2023) extended this by leveraging HuggingFace BART and T5 models for abstractive summarization [4]. Holt et al. (2022) used BERT-based sentence embeddings, clustering, and proximity to centroids for topic grouping [5]. Bhargavi and Subha Rao (2024) combined Whisper-based transcription with personalization efforts, yet did not fully integrate adaptive quiz generation or session-aware difficulty adjustment—areas central to the present work [6].

III. METHODOLOGY

LectureLM is built in four layers: (i) Transcript Retrieval, (ii) Content Processing, (iii) Feature Generation, and (iv) Adaptive Session Management. Fig. 1 illustrates the complete system architecture and information flow between modules.

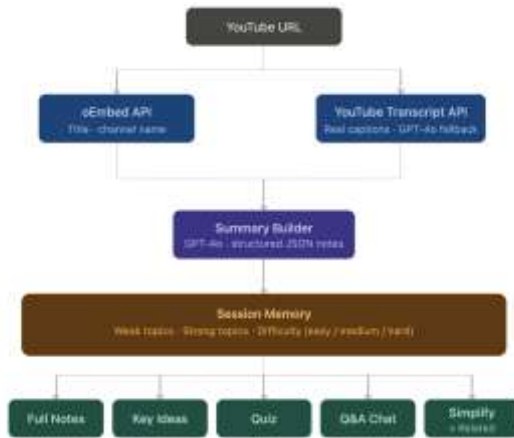


Fig. 1. LectureLM System Architecture Diagram

A. Transcript Retrieval Module

The system accepts a YouTube URL and, using a regular-expression parser, extracts the video ID. It then initiates two parallel lookups: the YouTube Data API fetches metadata (title, channel name) and oEmbed data. If real captions are available, it retrieves and limits them to approximately 800 tokens. If no captions exist, it invokes GPT-4o with a transcript-synthesis prompt to produce a structured synthetic transcript containing valid fields: overview, key topics, important definitions, takeaways, and full content.

B. Structured Note Generation

The Note Generator employs a dynamic prompting strategy. It first calls the YouTube API, joins retrieved caption snippets into a single string. On failure, it falls back to the synthetic ground-truth transcript. The system returns structured JSON notes plus human-readable summaries: Key Ideas (bulleted), Detailed Notes (multi-section), and trigger phrases. Fig. 2 shows the home screen with structured notes generated for a C++ tutorial video.



Fig. 2. LectureLM Home Screen — URL input and structured notes output

C. Adaptive Quiz Engine

The quiz engine allows learners to specify a question count between 1 and 20. Each multiple-choice question is generated by GPT-4o in structured JSON format containing the question, four options (A–D), the correct answer, and an explanation. Questions are displayed one-at-a-time in a progress bar. Fig. 3 shows Question 4 of 5 in an active quiz session. After the final answer is submitted, a detailed score report is generated as shown in Fig. 4.



Fig. 3. Quiz Engine — Question 4 of 5 with MCQ options and instant feedback



Fig. 4. Quiz Complete — Score summary with per-topic review guidance

D. Session Memory & Difficulty Adaptation

The Memory module is implemented as a lightweight dictionary mapping topic labels to strength scores. Scores increment for correct quiz answers and decrement for incorrect ones. The current session snapshot is passed to the LLM in subsequent calls, allowing the model to tailor the difficulty and focus of newly generated content to the learner's current performance profile. Fig. 5 shows the quiz feedback interface with the action bar (Full Notes, Quiz Me, Key Ideas, Related, Simplify).

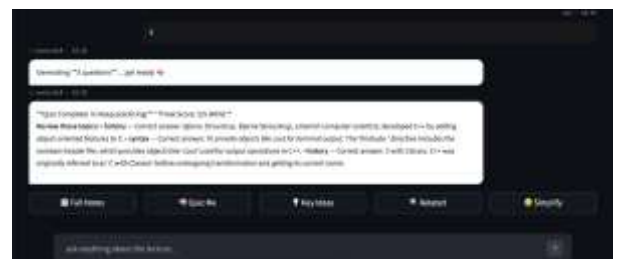


Fig. 5. Session interface showing quiz result, topic review, and navigation action bar



Fig. 6. Conversational quiz prompt — system requests number of questions via chat

E. RAG-Grounded Q&A Chat Interface

The Q&A interface grounds answers in the retrieved or synthesized transcript. A vector embedding of transcript chunks is stored in an in-memory index. On each user query, the top-k most semantically relevant chunks are retrieved and passed, along with the tutor system prompt, to GPT-4o. This ensures answers are verifiable and directly attributable to lecture content rather than open-domain hallucination.

F. Concept Simplifier

The concept simplifier targets learners struggling with technical jargon. The user enters any concept, and LectureLM passes it to GPT-4o with a plain-language simplification prompt aimed at a curious sixteen-year-old. The response includes a real-world analogy, a brief explanation, and concrete examples. Fig. 7 demonstrates the simplifier explaining “functions” using a sandwich/cookbook analogy.



Fig. 7. Concept Simplifier — plain-language explanation of programming functions

IV. RESULTS AND EVALUATION

LectureLM was deployed against a sample YouTube video titled “Learn C++ in 10 Minutes!! C++ Tutorial for Beginners” by AmanBytes (<https://www.youtube.com/watch?v=RSDzvIXmQi4>) and tested across all modules.

A. Structured Notes Quality

The structured note generation module successfully produced multi-section JSON notes covering Introduction to C++, Basic Syntax, Printing, Variables and Data Types, Input/Output, and Loops—accurately mapping to the lecture’s topic structure. Notes were generated in under 5 seconds on CPU hardware.

B. Quiz Performance

The quiz engine generated well-formed MCQ items with accurate distractors. A sample result showed a grade of 2/5 (40%), with instant per-question feedback and a comprehensive topic-by-topic review at the end. On subsequent quiz rounds with Memory adaptation, difficulty adjusted appropriately: easier questions were skipped for strong topics while additional hard questions targeted weak areas.

C. Concept Simplification

When queried with the concept of “function” in programming, the system produced an accessible sandwich/cookbook analogy that demonstrated the model’s ability to abstract away jargon without losing technical accuracy. Related resource recommendations correctly identified authoritative sources including freeCodeCamp, ProgrammingKnowledge, and GeeksforGeeks.

D. Proposed Evaluation Protocol

The evaluation framework comprises four measures applied to a corpus of 20 randomly selected YouTube lecture videos spanning engineering and management disciplines: (1) ROUGE-1, ROUGE-2, and ROUGE-L metrics against human-written reference summaries; (2) RAGAS Faithfulness and Context Relevance for RAG-grounded Q&A; (3) quiz accuracy benchmarked against annotated answer keys; and (4) a 30-minute human study using Likert-scale questionnaires comparing LectureLM against manual note-taking across 15, 30, and 60-minute study windows.

V. DISCUSSION

LectureLM demonstrates that combining transcript retrieval, LLM-based content generation, and session-aware adaptation produces a meaningfully richer learning experience than existing static summarizers. The key distinction is the feedback loop: the Memory module’s ability to track topic strength and adjust quiz difficulty in real-time positions LectureLM as a personalized tutor rather than a one-pass summarizer.

Several limitations bear mention. First, the system relies on YouTube’s caption infrastructure; in certain jurisdictions, copyright law may restrict transcript availability for specific content. Second, the current window-based approach may omit important context not appearing near the top of the transcript. Third, the adaptive logic is presently rule-based and would benefit from a long-term learner model. Fourth, video-level analysis does not yet extend to slide-level or visual content (OCR, STEM diagram parsing). Expanding to non-YouTube platforms (MP4 uploads, Vimeo, Coursera) and a mobile-friendly interface would broaden reach considerably.

VI. CONCLUSION

We presented LectureLM, a Python-Streamlit application that transforms YouTube lecture videos into structured, adaptive, and interactive study sessions. The system addresses a genuine gap in the current landscape of AI-powered educational tools: it does not merely

summarize, but learns from and adapts to the learner's performance. Key contributions include the four-layer architecture, the session Memory module, the RAG-grounded Q&A interface, and the adaptive quiz engine. Remaining work includes multimodal inputs, persistent user profiles, and a formal user study with statistical analysis.

ACKNOWLEDGEMENT

The authors thank the faculty of the Department of Computer Science and Engineering, Bhagwan Parshuram Institute of Technology, New Delhi, for their guidance and support. They also acknowledge the open-source communities behind Streamlit, OpenAI Python SDK, and the YouTube Data API v3.

REFERENCES

- [1] F. Retkowski et al., "End-to-End Neural ASR with Fluency Error Detection," arXiv:2504.08024, 2025.
- [2] L. Benedetto et al., "AI-Generated Summaries for Educational Video Content: Challenges and Opportunities," J. Educational Technology, vol. 18, no. 2, pp. 45–62, 2022.
- [3] P. Ilampirayan et al., "Automatic Lecture Note Generation Using NLP," Global Int. J. Emerging Technologies (GIJET), vol. 11, no. 2, pp. 35–48, 2021.
- [4] S. Gupta and R. Sen, "Abstractive Summarization of Educational Videos Using Pretrained Transformers," Int. J. for Multidisciplinary Research (IJFMR), vol. 8, Nov.–Dec. 2023.
- [5] J. Holt et al., "BERT-Based Topic Clustering for Lecture Content Analysis," arXiv:1906.04165, 2022.
- [6] A. Bhargavi and V. Subha Rao, "Whisper-Powered Personalized Learning: GPT Models in Adaptive Study Environments," arXiv:2410.04449, Oct. 2024.