Clip Scholar: Leveraging LLMs for AI-Driven Educational Content Generation

Bharath L[1], Gnanesh A R[1], Gopal [1], Madhan S[1], Nachiket Ganesh Apte [1]
[1]Department of Data Science and Artificial Intelligence,
(22bds013, 22bds023, 22bds025, 22bds036, 22bds041)@iiitdwd.ac.in

***Abstract*—This paper presents Clipscholar, an application designed to automate the extraction of transcripts from YouTube videos, generating structured notes and customizable quizzes. Developed using Flask, the YouTube Transcript API, and the Mistral-Large-Instruct-2411 model, Key features include Markdown-formatted note exports and support for multiple quiz formats, such as multiple-choice questions (MCQs), fill-in-the-blanks, and question-answer pairs.This report details the technical framework, methodology, cloud integration strategy, and potential applications of Clip Scholar as an educational tool. The platform aims to enhance learning efficiency by allowing users to process video-based information in textual formats tailored to individual learning preferences.**

***Index Terms*—e-learning, YouTube, Large Language Models, Mistral, web applications, quiz generation**

I. Introduction

In today's digital education landscape, YouTube hosts an extensive library of educational content spanning diverse academic disciplines [1] and skill areas. However, video-based learning presents inherent challenges [2] [3]: information retrieval is time-consuming, content density varies significantly, and learners with different preferences may struggle with purely audiovisual formats. This challenge highlights the need for an automated system [4] that can transform raw video transcripts into structured and interactive learning materials. Clip Scholar addresses this gap by leveraging Large Language Models (LLMs) [5] and natural language processing (NLP) [6] to automate the process of notes generation, quiz creation, and flashcard development from YouTube videos. This AI-powered platform extracts and processes video transcripts to generate structured notes and assessment materials, making educational content more accessible, searchable, and adaptable to diverse learning styles. The core innovation of Clip Scholar lies in its integration of transcript extraction technology with state-of-the-art language models. By utilizing Mistral-Large-Instruct-2411, the platform achieves high-quality content summarization and educational material generation with minimal human intervention. This approach not only enhances accessibility but also enables personalized learning experiences through automatically generated quizzes that reinforce key concepts. This report outlines the technical architecture, methodological approach, implementation strategies, and potential applications of Clip Scholar within educational ecosystems.The complete codebase for our system is available at https://github.com/madhans476/Clip-Scholar.git.

II. Preliminaries

Clipscholar is built upon the following core technologies:

- **Supadata AI Transcript Extractor**: A Python library that retrieves video subtitles as text [7]. Its functionality relies on caption availability.
- **Mistral-Large-Instruct-2411**: An advanced language model from Mistral AI, selected for its efficacy in generating summaries and quiz content [8] [9].
- **Flask**: A lightweight Python web framework facilitating efficient API request handling [10] [11].

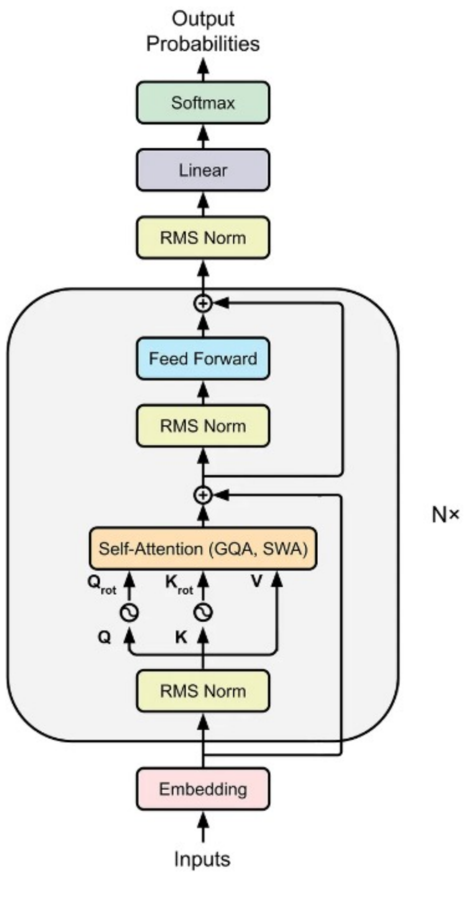

Fig. 1. Mistral-Large-Instruct-2411 Architecture Diagram

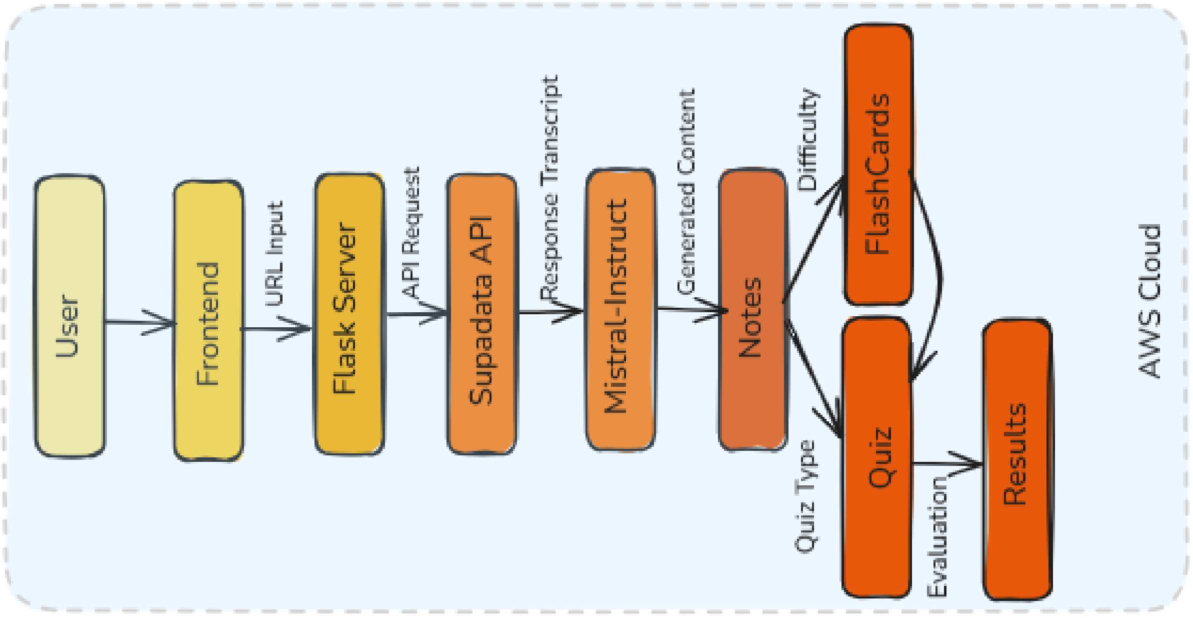

Fig. 2. Clipscholar Architecture Diagram

III. Methodology

A. YouTube Video Processing

Clip Scholar employs a systematic approach to process YouTube videos, focusing on accuracy and efficiency in transcript extraction.

1) Supadata AI Transcript Extractor Integration: The platform leverages the Supadata AI Transcript Extractor, a dedicated library that provides access to automatically generated subtitles from YouTube videos. This API offers several advantages:

1) **Direct Access**: The API enables direct programmatic access to transcripts without the need for browser automation or scraping.
2) **Language Support**: Multiple language transcripts can be accessed, enhancing the platform's global applicability.
3) **Timestamp Integration**: Each text segment includes precise timestamps, allowing for synchronized note referencing.

B. System Architecture

Clipscholar's architecture comprises a back end for content processing and a front end for user interaction, as illustrated in figure 2. Implemented on a Flask server hosted on an AWS t3.micro instance with Ubuntu 24.04 LTS, it performs three primary functions: transcript extraction, notes generation, quiz production, and flashcard generation. The workflow begins with user input and concludes with rendered outputs. The system prompts for structured notes, quiz (MCQ) generation, and flashcard generation are as follows:

1) **The Prompt for Notes Generation**: *"I have a YouTube video transcript of an education related video. Please create comprehensive, structured notes that:*
1. Identify and organize the main topics and subtopics with clear headings
2. Incorporate relevant notes, context, or text from external sources to enrich the educational content
3. Include key points, definitions, examples, and detailed explanations of the content
4. Format the notes in proper markdown with headings, bullet points, and emphasis where appropriate
5. Add relevant citations to academic papers, books, or authoritative sources where appropriate using APA format
6. Include a References section at the end with the full citations
Transcript:
{transcript text}"
2) **The Prompt for Quiz Generation (MCQ)**: *"Based on only the main topic of the following notes, generate {num questions} multiple-choice questions (MCQs) with {difficulty level} difficulty level.*
For each question:
1. Create a clear question about an important concept
2. Provide 4 options (A, B, C, D)
3. Indicate the correct answer
4. Format as a JSON array of objects with keys: question, options (array), correctAnswer (index)
Notes:
{notes}"
3) **The Prompt for Flashcard Generation**: *"Based on the main topic of the following notes, generate {num_flashcards} flashcards with {difficulty_level} difficulty level. For each flashcard:*

1. Create a front side with a question, concept name, or term
2. Create a back side with the answer, explanation, or definition
3. Enhance the content with additional context from external resources for broader understanding where appropriate
4. Include relevant examples or analogies where helpful
5. Format the back side with proper markdown with bullet points, emphasis, and clear structure where appropriate
6. If applicable, add citations to academic papers, books, or authoritative sources using APA format
Format as a JSON array of objects with keys: front and back
Notes:
{notes}"

C. Notes Generation

1) **Input**: Users submit a YouTube video URL through Clipscholar's interface (e.g., https://www.youtube.com/watch?v=VIDEO_ID), as shown in Figure 3.
2) **Processing**: The video ID is extracted, and the Supadata ai transcript extractor retrieves the transcript. Mistral-Large-Instruct-2411 converts this into structured bullet-point notes.
3) **Output**: Notes are displayed on the interface and available for download as Markdown files via JavaScript.

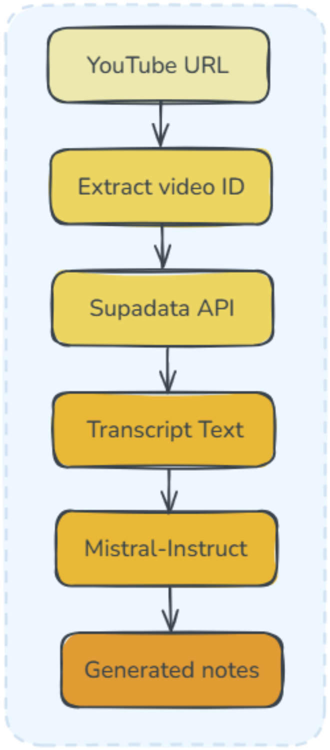

Fig. 3. Notes Generation Dataflow Diagram

D. Quiz Creation

1) **Input**: Generated notes and a user-specified question count (5–20), entered via Clipscholar's interface, as shown in Figure 4.
2) **Processing**: Mistral-Large-Instruct-2411 produces quizzes in formats such as MCQs, fill-in-the-blanks, and Q&A, based on the notes.
3) **Output**: Quizzes are presented as text on the interface, with plans for future interactivity.
4) **Automatic Evaluation**: Generated quizzes support automated scoring and feedback based on user responses.

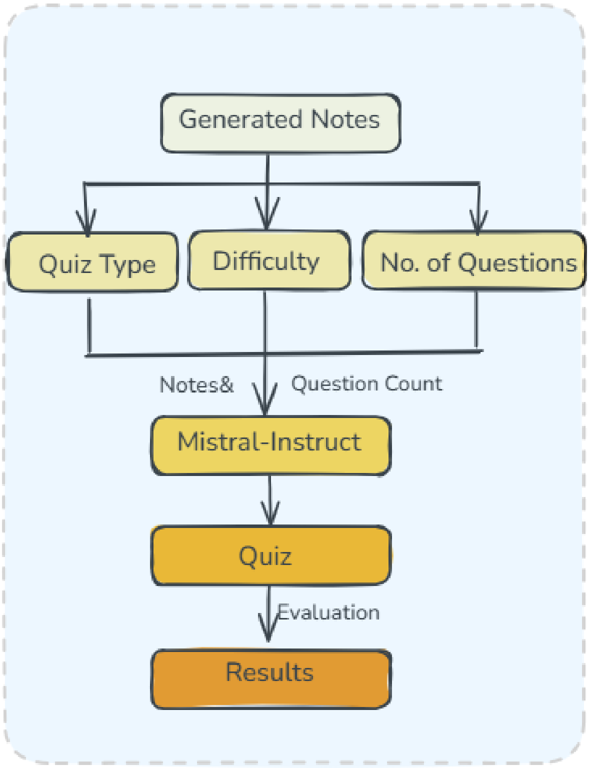

Fig. 4. Quiz Generation Dataflow Diagram

E. Flashcard Generation

1) **Input**: Generated notes or quiz content, entered via Clipscholar's interface, as shown in Figure 5.
2) **Processing**: Mistral-Large-Instruct-2411 converts the notes or quiz questions into flashcard format, featuring a term or question on one side and the definition or answer on the reverse, formatted in Markdown.
3) **Output**: Flashcards are displayed on the interface and available for download as Markdown files via JavaScript, supporting self-paced review.

F. Deployment

Clipscholar is deployed on an Amazon Web Services (AWS) EC2 instance [12] to ensure scalable and reliable access. The deployment process involves launching a t2.micro instance with Ubuntu 20.04 LTS, configuring the Flask application, and

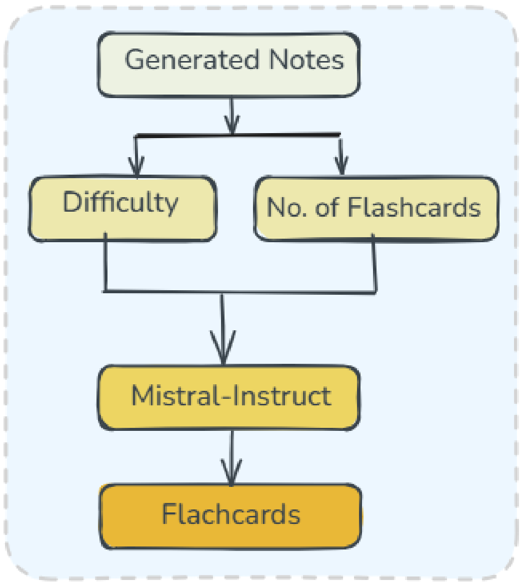

Fig. 5. Flashcards Generation Dataflow Diagram

setting up a Gunicorn WSGI server with Nginx as a reverse proxy for efficient request handling. Security is enhanced through AWS security groups, restricting access to ports 80 (HTTP) and 22 (SSH). The Supadata AI Transcript Extractor and Mistral-Large-Instruct-2411 model are integrated into the instance, with dependencies managed via a virtual environment. This cloud-based hosting enables Clipscholar to process YouTube video transcripts and deliver notes and quizzes to users seamlessly, with potential for future scalability.

The frontend, developed with HTML, JavaScript, and Bootstrap 5.3, provides a responsive interface featuring URL submission, notes display with download functionality, and quiz generation controls. Implemented and tested on a local server (Intel i5, 16GB RAM, no GPU), Clipscholar's resource demands highlight the need for optimization of Mistral-Large-Instruct-2411.

IV. Conclusion

This paper details Clipscholar, a significant advancement in bridging video-based and text-based educational content delivery. By leveraging state-of-the-art language model technology, specifically Mistral-Large-Instruct-2411, and a robust Flask backend, the platform transforms passive video consumption into active learning experiences through structured notes and interactive assessments. The implementation of Notes Generation and Quiz Generation delivers fast, accurate, and structured outputs, enhancing learning efficiency.

The technological framework established in this project demonstrates the viability of automated educational content transformation at scale. Its cloud-based architecture ensures performance reliability and cost-effectiveness, while the modular design allows for future integration of additional features and AI capabilities. Test results affirm its reliability and user satisfaction, despite minor challenges with transcript-dependent accuracy.

Key contributions of the Clip Scholar platform include:

1) **Information Accessibility**: Converting video-based knowledge into searchable, referenceable text formats.
2) **Learning Efficiency**: Reducing the time required to extract and internalize key concepts from educational videos.
3) **Assessment Integration**: Seamlessly connecting content consumption with knowledge validation through automated quiz generation.

As a versatile platform, Clip Scholar holds significant potential to support students, educators, and lifelong learners, paving the way for future enhancements to broaden its impact in educational technology.

V. Limitations

- **Processing Time**: Clipscholar requires extended time to process lengthy videos, impacting efficiency for longer content.
- **Math-Related Performance**: The system exhibits comparatively lower performance when handling math-related videos, limiting its effectiveness in this domain.
- **Language Restriction**: Clipscholar currently supports only English videos, restricting its applicability to multilingual content.

References

[1] M. N. Giannakos, K. Chorianopoulos, and N. Chrisochoides, "Making sense of video-based learning analytics: Lessons learned from youtube," *IEEE Transactions on Learning Technologies*, vol. 8, no. 4, pp. 356–369, 2015.

[2] P. J. Guo, J. Kim, and R. Rubin, "How video production affects student engagement: An empirical study of mooc videos," in *Proceedings of the ACM Conference on Learning at Scale*, pp. 41–50, 2014.

[3] S. Burgess and K. Green, "The impact of digital technology on learning: A review of the evidence," *Education Endowment Foundation*, 2020.

[4] S. Khan, M. Babiker, and A. Hussein, "Using large language models for automated educational content generation," *IEEE Access*, vol. 10, pp. 120456–120470, 2022.

[5] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2021.

[6] J. D. Williams, "Natural language processing in educational applications: Opportunities and challenges," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[7] "Supadata ai transcript extractor." [Online]. Available: https://supadata.ai/.

[8] Mistral AI, "Mistral-large-instruct-2411 model," 2024. [Online]. Available: https://mistral.ai/.

[9] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[10] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2nd ed., 2018.

[11] H. Hosseini, T. Doran, and S. Kim, "Building scalable web apis with flask and python," *IEEE Computing in Science Engineering*, vol. 23, no. 2, pp. 46–55, 2021.

[12] Amazon Web Services, *Amazon EC2 Documentation*, 2024. [Online]. Available: https://docs.aws.amazon.com/ec2/.