Usability Evaluation of a Course-Based Content Retrieval Application

Jan Carlos Ramos
jramos61@gatech.edu

***Abstract*—**The emergence of the internet has led to the proliferation of vast amounts of online instructional content. In order to deliver education at scale, Massive Open Online Courses have emerged. MOOCs use learning management systems (LMS) like Canvas and Blackboard to deliver their instructional content, which are not always organized for optimal learning. From a students perspective, content delivery on these LMS platforms can feel disjointed and difficult to navigate. Video lectures often lack detailed timestamps, and textbooks may require extensive skimming to locate specific information. The irony of having vast amounts of information is that it can become difficult to access efficiently. This paper introduces EmbedEdu, a system that leverages vector databases to store pre-labeled online course content, such as video lectures, textbooks, and discussion forum posts. The system enables multi-modal semantic search, allowing students to query questions or keywords and retrieve the most relevant course content—whether from textbooks, slide decks, video lectures, or discussion forums—stored in the database. This facilitates efficient and focused learning by addressing the organizational challenges of traditional LMS platforms.

1 INTRODUCTION

Video lectures, discussion forums, slide decks, and online textbooks have become essential components of online education. Whether the information is delivered via a learning management system (LMS) like Canvas or Blackboard, or openly available on the web through platforms such as YouTube, open-access repositories, or personal blogs, the amount of online educational resources is bountiful. Properly managing and structuring these resources can significantly enhance learners' ability to access and engage with the content effectively. The proposed system in this paper, EmbedEdu, leverages vector databases to store pre-labeled course content from Georgia Tech's CS1301 - Intro to Computing course, taught

by Dr. David Joyner.

EmbedEdu transforms traditional course content—including videos, text, and forums—into a unified, searchable resource using vector database architecture that enables multi-modal search across multiple formats. By addressing the organizational and retrieval challenges of traditional MOOC materials, this work seeks to bridge the gap between the vast potential of online education and the practical needs of learners in a digital environment.

2 RELATED WORKS

Recent advancements in multi-modal processing and transformer architectures have inspired innovative methods to make educational content, particularly video lectures, more accessible, searchable, and meaningful for learners. Ghauri, Hakimov, and Ewerth [7] introduced a multi-modal neural network for classifying educational video segments, utilizing a dual transformer architecture to capture semantic context. Wu et al. [12] further extended these advancements by addressing the impact of varying video quality on retrieval performance. Their pipeline combined embeddings from video frames, speech-to-text transcriptions, and acoustic features, demonstrating the robustness of multi-modal approaches when one modality is impaired. Oliveira et al. [9] complemented this work with a six-step pipeline for video summarization using transformer models. This pipeline involved transcribing the video, followed by preprocessing, chunking, and summarization. The approach generated concise summaries aligned with video segments using text similarity techniques, a feature akin to a Retrieval-Augmented Generation (RAG) approach. Li et al. [8] further expanded on this by demonstrating how RAG could enhance educational content processing, achieving impressive accuracy. Alawwad et al. [1] advanced this direction by specifically addressing challenges in textbook question answering using large language models (LLMs) and RAG, with a focus on multi-modal video search capabilities. Building on this foundation, the proposed system employs vector databases to store decomposed knowledge components—spanning videos, discussion forums, textbooks, and slide decks—as embeddings, enabling precise semantic search and facilitating a more personalized and efficient learning experience.

3 METHODOLOGY

3.1 Data Preparation

EmbedEdu is able to process various mediums of instructional content. The system handles two primary content categories: video lectures and online textbooks, each requiring distinct preprocessing pipelines. The pipelines use a combination of state-of-the-art models and efficient processing techniques:

3.1.1 *Audio Extraction*

The system uses yt-dlp to download and extract high-quality audio (192kbps MP3) from source videos. Audio extraction is parallelized using asyncio for efficient processing of multiple videos.

3.1.2 *Speech-to-Text Transcription*

OpenAI's Whisper base model is used for speech recognition and transcription. Transcription output includes timestamped segments with text and temporal alignment

3.1.3 *Text Processing and Embedding*

Transcribed text is processed in batches of 64 segments to manage memory efficiently. The sentence-transformer model "multi-qa-mpnet-base-dot-v1" generates 768-dimensional dense vector embeddings. This model was selected as it is specifically optimized for retrieval tasks and semantic similarity search.

3.1.4 *Visual Processing and Embedding*

Video frames are processed in batches of 32 frames to manage memory efficiently. The visual-transformer model "clip-vit-base-patch32" generates 512-dimensional dense vector embeddings for visual content. This model was selected because it is able to align visual and textual representations which enables visual content retrieval using textual queries.

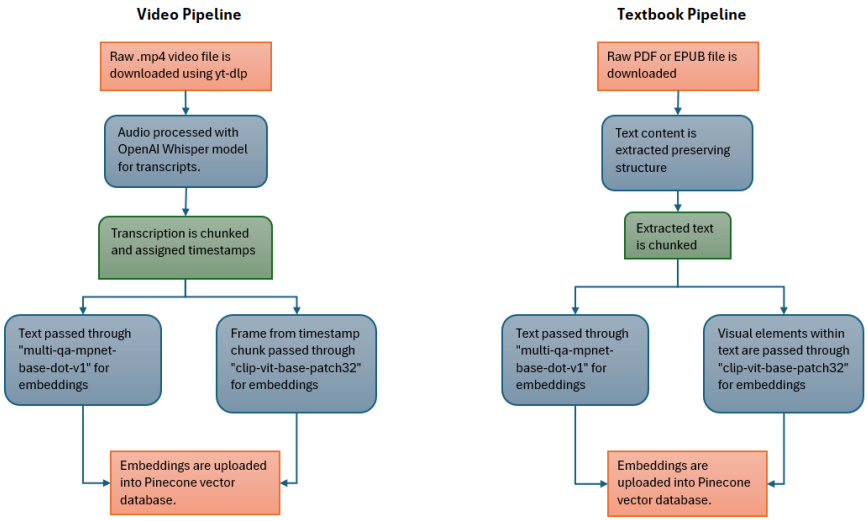

Figure 1—Video and Textbook Processing Pipeline Architecture

3.2 Video Pipeline

The online video lecture content is processed following this sequence of steps:

- The raw mp4 video file is downloaded with the source platform and assigned a unique video ID.
- The text transcription with timestamps is generated by processing the video's audio content.
- The text transcription is divided into manageable chunks.
- Each chunk is then passed through a text embedding model to produce its dense vector representation.
- Concurrently, one frame from each chunk will be passed through a visual embedding model to generate a vector representation of the visual context.
- Finally, these embeddings are uploaded into a vector database, creating a searchable repository that supports multi-modal queries.

3.3 Textbook Pipeline

The online textbook content is processed following this sequence of steps:

- The raw PDF or EPUB file is downloaded from the source platform and assigned a unique textbook ID.
- The text content of the textbook is extracted, preserving the structural elements such as chapters, sections, and page numbers.

- The extracted text is divided into manageable chunks, with each chunk linked to its corresponding chapter, section, and page numbers to retain contextual alignment.
- Each chunk is passed through a text embedding model to produce its dense vector representation.
- Concurrently, associated visual elements (e.g., images, diagrams, or charts) within the text are extracted and processed through a visual embedding model to generate vector representations.
- Finally, these embeddings are uploaded into a vector database, creating a searchable repository that supports multi-modal queries, enabling students to retrieve text, diagrams, or other visual materials efficiently.

3.4 Database Design

The vector database integration leverages Pinecone's serverless architecture on AWS, initialized with a dual-index system: a 768-dimensional index for text and a 512-dimensional index for visual content, both optimized for cosine similarity search. Content is processed through a parallel pipeline where textual segments are batched in groups of 64 using multi-qa-mpnet-base-dot-v1, while visual frames are processed in batches of 32 using the CLIP ViT-B/32 model. The upload process creates unique identifiers for each segment by combining video IDs with timestamps, then packages both text and visual embeddings with rich metadata including frame timestamps, video titles, and direct URLs. These multimodal documents are upserted to Pinecone in batches, creating a searchable database that supports both text and visual queries with optional metadata filtering.

Key	Description
id	Unique identifier for the video segment.
thumbnail	URL to the thumbnail image of the video segment.
title	Title of the video segment.
views	Number of views for the video.
length	Total length of the video in seconds.
url	URL to the video starting at the segment's start time.
start	Start time (in seconds) of the segment within the video.
end	End time (in seconds) of the segment within the video.
text	Transcript or text associated with the segment.
video_id	Identifier for the video.
visual_embeddings	Numerical representation (vector) capturing the visual content of the thumbnail or video frame.
textual_embeddings	Numerical representation (vector) capturing the semantic meaning of the text.

Table 1—Description of Keys in the Pinecone Vector Database for Video

Key	Description
id	Unique identifier for the textbook section.
chapter_title	Title of the chapter containing the section.
section_title	Title of the section within the chapter.
page_range	Range of pages (e.g., 15–20) covered by the section.
url	URL to the digital version of the textbook or section, if available.
text	Full text content of the section, including headings and subheadings.
start_page	The starting page number of the section.
end_page	The ending page number of the section.
textual_embeddings	Numerical representation (vector) capturing the semantic meaning of the text in the section.
visual_embeddings	Numerical representation (vector) capturing the visual content of any figures or illustrations in the section.
keywords	List of key terms or concepts covered in the section.

Table 2—Description of Keys in the Pinecone Vector Database for Textbooks

3.5 User Interface

At the top of the interface, an input bar allows students to type their questions in natural language. Upon submission, the application retrieves the most relevant video lecture by computing the cosine similarity score between the embedded query and entries in the vector database. The selected video is displayed prominently in the center of the interface and automatically begins playback at the precise segment that addresses the student's query. Directly below the video, the interface presents a Python coding example related to the student's question. This example is dynamically generated using a Retrieval-Augmented Generation (RAG) process that summarizes the highest-ranked embeddings from the database. Beneath the code example, the application highlights the most relevant textbook section, also stored as vector embeddings, providing comprehensive learning resources in one view. This layered structure ensures that students receive a multi-modal response—combining video, code, and text—for their queries, maximizing the utility of the platform.

4 FUTURE WORK

Future directions for this project include conducting a qualitative experiment aimed at evaluating the usability and effectiveness of the educational application. The study has been approved by the Georgia Tech Institutional Review Board (IRB) and will involve 10-20 participants who are currently enrolled OMSCS graduate students based in the United States. Participants will attend a single session lasting approximately 90 minutes, during which they will interact with the application to assess its ability to retrieve relevant course content and navigate the system intuitively.

Participants will be briefed on the study's purpose and procedures at the start of the session, including the use of screen recording to capture their interactions and behaviors. They will perform tasks such as querying the application for course-related information and switching between recommended content. Feedback will be collected through structured questions and scales, allowing participants to rate the relevance of retrieved content, the helpfulness of coding examples, and the ease of navigation. Improvised follow-up questions may also be asked to gain deeper insights. Additionally, participants will share thoughts on missing features or unmet expectations to refine the application further.

All recordings and data will be stored securely and de-identified to ensure participant confidentiality. The results of this study will contribute valuable insights into improving the educational application's design and functionality, ultimately benefiting both students and educators. Participants are informed that their involvement is voluntary, with no compensation provided, and they may withdraw at any time without penalty. These protocols are outlined in a detailed consent form, which ensures transparency and compliance with ethical standards.

REFERENCES

[1] Alawwad, H. A., Alhothali, A., Naseem, U., Alkhathlan, A., & Jamal, A. (2024). Enhancing Textbook Question Answering Task with Large Language Models and Retrieval Augmented Generation. *arXiv preprint arXiv:2402.05128*.

[2] Arnold, S., Schneider, R., Cudré-Mauroux, P., Gers, F. A., & Löser, A. (2019). SECTOR: A Neural Model for Coherent Topic Segmentation and Classification. *arXiv preprint arXiv:1902.04793*.

[3] Becker, B. A., & Fitzpatrick, T. (2019). What Do CS1 Syllabi Reveal About Our Expectations of Introductory Programming Students? In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1011-1017). ACM.

[4] Boyd, K., McAllister, P., Mulvenna, M. D., Bond, R., Wang, H., Spence, I., Wu, G., Haider, A., Cooper, R., & Wood, A. (2023). Designing Multimodal Video Search by Examples (MVSE) user interfaces: elicitation of UX requirements and insights from semi-structured interviews. In *European Conference on Cognitive Ergonomics 2023* (pp. 1-8). ACM.

[5] Das, A., & Das, P. P. (2020). Incorporating Domain Knowledge To Improve Topic Segmentation Of Long MOOC Lecture Videos. *Journal of LaTeX Class Files*, 14(8), 1–12.

[6] Dhulipala, L., Hadian, M., Jayaram, R., Lee, J., & Mirrokni, V. (2024). MUVERA: Multi-Vector Retrieval via Fixed Dimensional Encodings. *arXiv preprint arXiv:2405.19504*.

[7] Ghauri, J. A., Hakimov, S., & Ewerth, R. (2020). Classification of Important Segments in Educational Videos using Multimodal Features. In *Proceedings of the CIKM 2020 Workshops*. Galway, Ireland: CEUR Workshop Proceedings.

[8] Li, X., Henriksson, A., Duneld, M., Nouri, J., & Wu, Y. (2024). Supporting Teaching-to-the-Curriculum by Linking Diagnostic Tests to Curriculum Goals: Using Textbook Content as Context for Retrieval-Augmented Generation with Large Language Models. In *International Conference on Artificial Intelligence in Education* (pp. 118-132). Springer.

[9] Oliveira, L. M. R., Shuen, L. C., Cruz, A. K. B. S., & Neto, C. S. S. (2023). Summarization of Educational Videos with Transformers Networks. In *Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia '23)* (pp. 137-143). ACM, Ribeirão Preto, Brazil. https://doi.org/10.1145/3617023.3617042.

[10] Soares, E. R., & Barrére, E. (2018). Automatic Topic Segmentation for Video Lectures Using Low and High-Level Audio Features. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web* (pp. 189-196). ACM.

[11] Tuna, T., Joshi, M., Varghese, V., Deshpande, R., Subhlok, J., & Verma, R. (2015). Topic Based Segmentation of Classroom Videos. In *International Conference on E-Learning & Teaching*. IEEE.

[12] Wu, G., Haider, A., Tian, X., Loweimi, E., Chan, C. H., Qian, M., Muhammad, A., Spence, I., Cooper, R., Ng, W. W. Y., Kittler, J., Gales, M., & Wang, H. (2024). Multi-modal video search by examples—A video quality impact analysis. *IET Computer Vision*, *18*(7), 1017-1033.