Semantic News Video Search

Multimodal RAG System for Video Retrieval

Project Documentation

Avaz Asgarov

January 2, 2026

Table of Contents

1 Introduction

The rapid accumulation of video content in news archives presents a significant challenge for efficient retrieval. Traditional search methods, which often rely on manual tagging or simple keyword matching against filenames, fail to capture the semantic depth and multimodal nature of news broadcasts. This project implements a **Multimodal Retrieval-Augmented Generation (RAG)** system designed specifically to address these limitations. By ingesting video content and analyzing it through three distinct modalities—Audio (speech transcription), Visuals (scene context), and Text (on-screen OCR)—the system makes video archives searchable via natural language queries.

1.1 Problem Statement

News organizations generate hundreds of hours of video content daily. Finding specific segments within this vast repository (e.g., *"Find the clip where the President discusses the peace treaty"*) is currently a labor-intensive process requiring manual review. Existing tools often miss context that is purely visual (e.g., a handshake) or contained only in on-screen text (e.g., a ticker headline), leading to incomplete search results and inefficient workflows.

1.2 Proposed Approach

To solve this, we developed a pipeline that processes raw video files into semantically indexed chunks. The system utilizes a sliding window approach to segment videos, ensuring context is preserved across boundaries. Each segment is then processed using state-of-the-art AI models: OpenAI's Whisper for audio transcription, GPT-4o for visual scene description, and EasyOCR for extracting on-screen text. These multimodal inputs are synthesized into a unified vector embedding stored in ChromaDB, enabling high-precision semantic search and RAG-based answer generation.

1.3 System Architecture

The core workflow follows an *Ingest-Process-Index-Retrieve* pipeline, as illustrated in Figure 1. Videos are uploaded to a local directory, where they undergo automated segmentation. The resulting metadata is indexed, allowing the RAG engine to retrieve precise moments and generate factual answers to user queries.

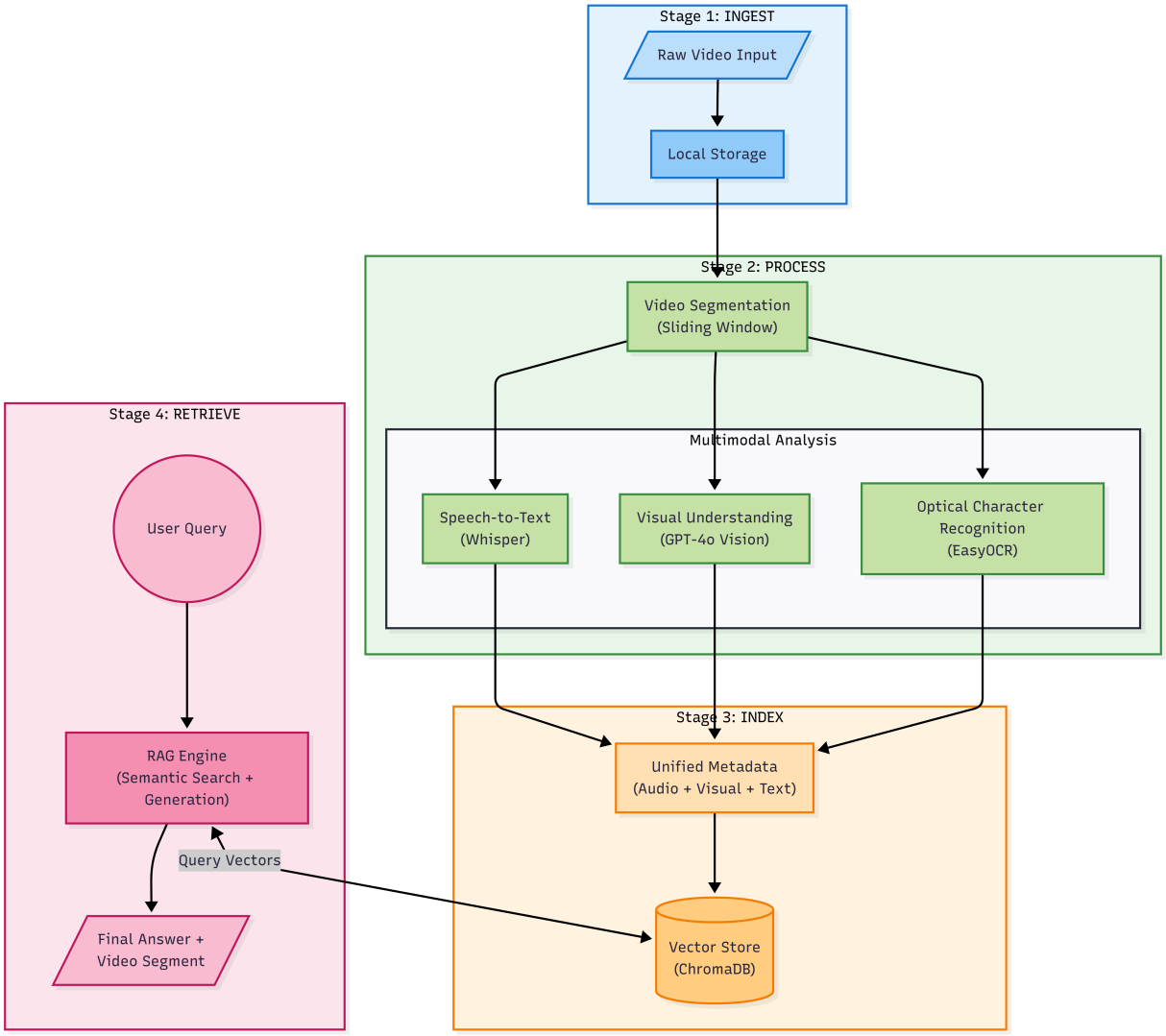

Figure 1: Architecture of the multimodal video retrieval system using Retrieval-Augmented Generation (RAG).

2 Methodology

This section details the methodologies employed in the system's architecture and the rationale for each design choice. The approach prioritizes semantic continuity and multimodal data fusion for high-precision retrieval.

2.1 Temporal Segmentation: The Sliding Window Strategy

The foundation of any video retrieval system is how it handles the temporal dimension. A raw video file is too large to serve as a single semantic unit. To address this, we implemented a **Sliding Window Segmentation** strategy.

2.1.1 Chunking Parameters

The system divides video content into segments with these parameters:

- **Window Size:** 20 seconds

- **Step Size:** 10 seconds
- **Overlap:** 10 seconds

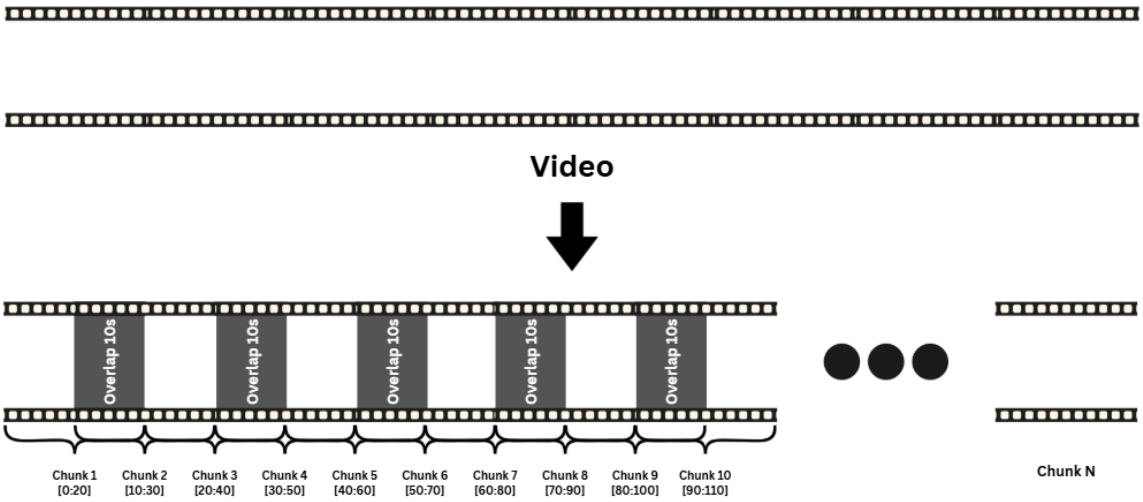

Figure 2: Visual representation of the sliding window segmentation strategy.

2.1.2 Rationale for Sliding Windows

Standard segmentation uses "hard cuts" (e.g., slicing at 0s, 20s, 40s). This introduces the *Boundary Problem*, where events at cut points are split between chunks. By using a sliding window with 50% overlap, we ensure every moment appears in the center of at least one chunk. This guarantees complete sentences and events are captured fully intact, improving embedding quality and retrieval accuracy.

2.2 Multimodal Ingestion Pipeline

To fully comprehend the context of a news broadcast, the system analyzes three distinct layers of data simultaneously: Audio, Visual, and Textual.

2.2.1 Audio Layer: Transcription

The audio component is extracted from video using the `MoviePy` library, converting it to a temporary audio file. This file is processed by OpenAI's **Whisper** model, which provides high-fidelity speech-to-text transcription with precise timestamps for alignment with temporal chunks.

Figure 3: Workflow for audio extraction using MoviePy followed by transcription via Whisper.

2.2.2 Visual Layer: Optimized Scene Understanding

For each 20-second chunk, the system generates a visual description. Analyzing every frame is redundant and costly. To optimize this, we implemented scene change detection using the

Mean Squared Error (MSE) metric with OpenCV.

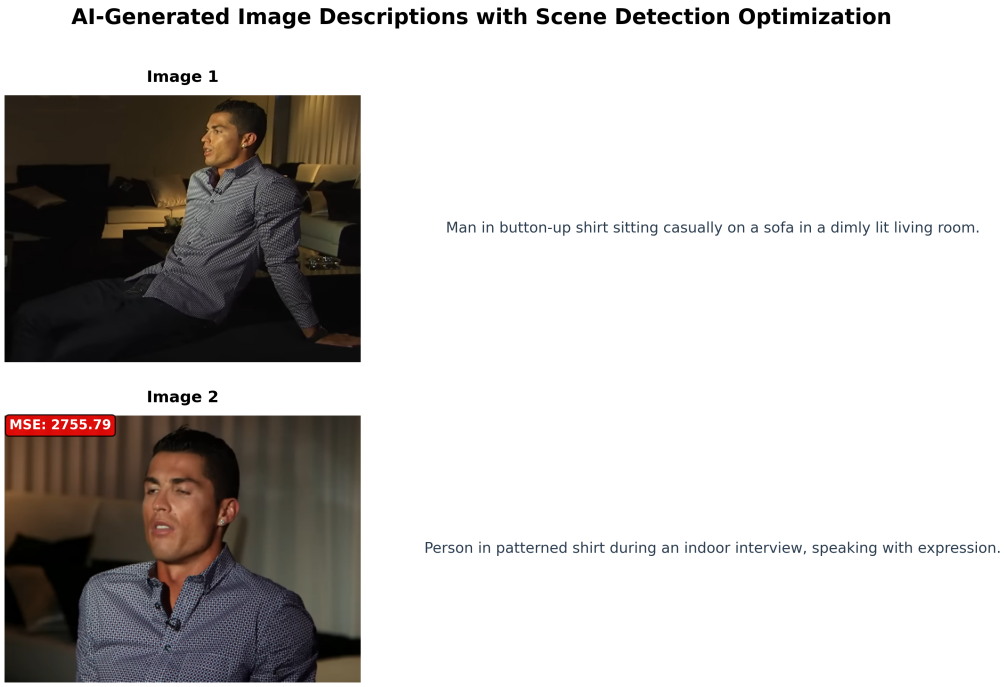

Figure 4: Visual processing. Frames are compared using MSE; only distinct scenes trigger a call to the Vision API.

The system extracts the middle frame of each chunk and compares it to the previously analyzed frame. If the MSE exceeds a predefined threshold (indicating a significant scene change), the frame is sent to **the GPT-4 Vision API** for descriptive captioning. If the MSE is low (indicating a static scene), the previous caption is reused. This reduces API costs and latency.

Note: With local multimodal models like **LLaVA**, this optimization could be eliminated, allowing frame-by-frame analysis without cost concerns.

2.2.3 Text Layer: Optical Character Recognition (OCR)

News broadcasts display critical unspoken information in tickers, banners, and chyrons. To capture this, we integrated **EasyOCR**.

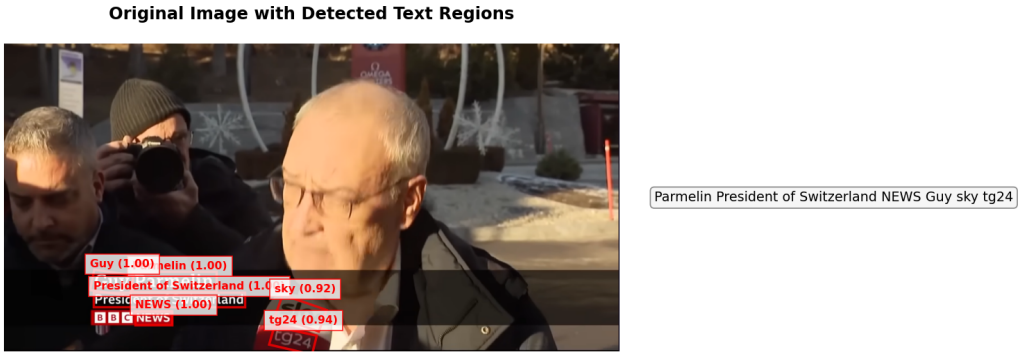

Figure 5: Example of text extraction from a news frame using EasyOCR, capturing lower-third data.

For every analyzed keyframe, EasyOCR scans for text overlays. This allows for the indexing of metadata that would otherwise be lost, including tickers, speaker names, and displayed

statistics. By including OCR output in the search index, users can retrieve videos based on visual text cues.

2.3 Metadata Enrichment

Beyond raw multimodal data, structuring the content is essential for effective retrieval in a news archive. We implemented two key enrichment processes: Named Entity Recognition (NER) and Auto-Tagging.

2.3.1 Named Entity Recognition (NER)

While vector embeddings capture semantic meaning, they can be imprecise with specific proper nouns. We use **Spacy** library to extract structured entities from video transcripts, identifying People, Organizations, and Locations.

This enables structured filtering alongside semantic search. For example, processing a video transcript can extract:

People: `Derek, Putin`

Locations: `Ukraine, Moscow, Washington`

These named entities are stored alongside vector embeddings, enabling precise retrieval of specific people or locations mentioned in videos.

2.3.2 Automatic Video Tagging

To organize the archive into a navigable taxonomy, we implemented an Auto-Tagging system that categorizes entire videos into consistent topics.

The system aggregates transcripts from the first minute of video and prompts **GPT-4o** to classify content using a predefined taxonomy: [Politics, Conflict/War, Sports, Economy, Technology, Weather, Health, Entertainment]. The output is deterministic to ensure consistency.

These tags are saved to a JSON file for frontend display, enabling instant category filtering. Example output format:

```
{
    "azerbaijan.mp4": "Conflict/War, Politics",
    "messi.mp4": "Sports, Entertainment",
    "ronaldo.mp4": "General",
    "ukraine.mp4": "Politics, Conflict/War"
}
```

2.4 Vector Storage and Retrieval-Augmented Generation (RAG)

The final stage of the methodology unifies the processed data into a queryable knowledge base using Vector Storage and RAG.

2.4.1 Synthesized Embeddings

For every 20-second chunk, the system fuses the three multimodal data streams into a single semantic text block:

`[Visual Scene]: ... [On-Screen Text]: ... [Audio Transcript]: ...`

This synthesized text is converted into a vector using OpenAI's `text-embedding-3-small` model. These vectors are stored in **ChromaDB**, a local open-source vector database optimized for high-speed similarity search. Storing the "meaning" of the video rather than just keywords allows the system to handle vague queries (e.g., "people arguing") effectively.

2.4.2 The RAG Workflow

The retrieval process is designed to answer user queries factually by grounding the LLM in retrieved evidence. The workflow proceeds as follows:

1. **Query Vectorization:** The user's natural language query is converted into a vector using the same embedding model.
2. **Semantic Retrieval:** ChromaDB performs a similarity search (Cosine Similarity) to find the **Top 3** most relevant video chunks. The system returns not just the video ID, but precise metadata:
   - **Exact Timestamp:** Start and end times for playback.
   - **Context:** The transcribed text and visual description.
   - **Entities:** The extracted People, Locations, and Organizations.
3. **Contextual Generation:** The retrieved text chunks are passed to the **GPT-4o** model as a "Context" block. The model is instructed via a strict system prompt to generate an answer *only* using this provided context, thereby minimizing hallucinations.

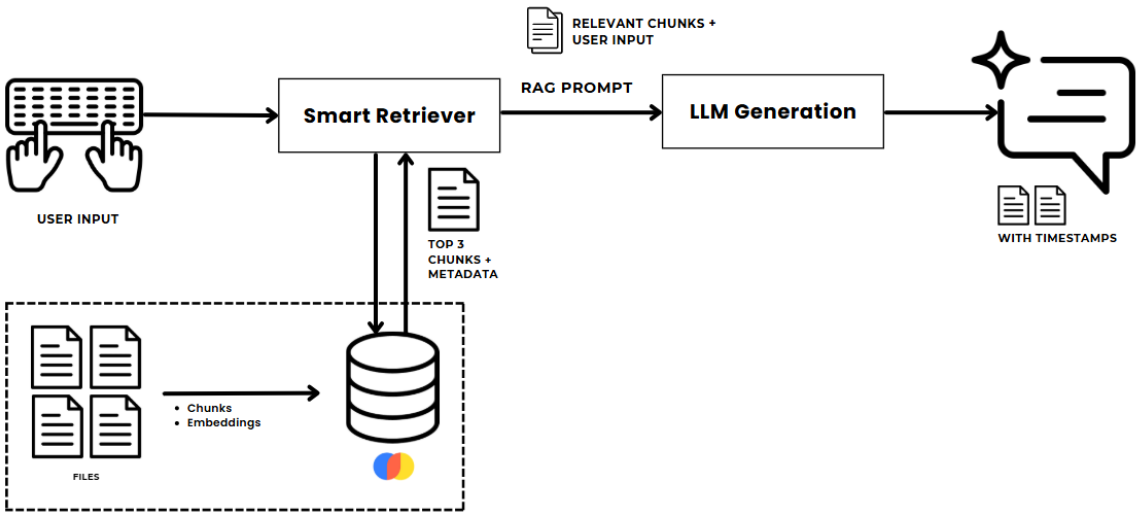

Figure 6: The Retrieval-Augmented Generation (RAG) pipeline.

3 Results

This section presents the experimental results of the semantic news video search system. The system was evaluated using a diverse set of queries targeting different modalities: factual retrieval (Audio), scene understanding (Visual), text extraction (OCR), and complex synthesis (RAG). The following examples demonstrate the system's ability to retrieve precise video segments and generate accurate AI summaries.

3.1 Factual Retrieval (Audio Transcript)

The system's ability to extract specific numerical facts from speech was tested using the query: *"What specific amount of additional aid did President Zelensky discuss with US officials?"*

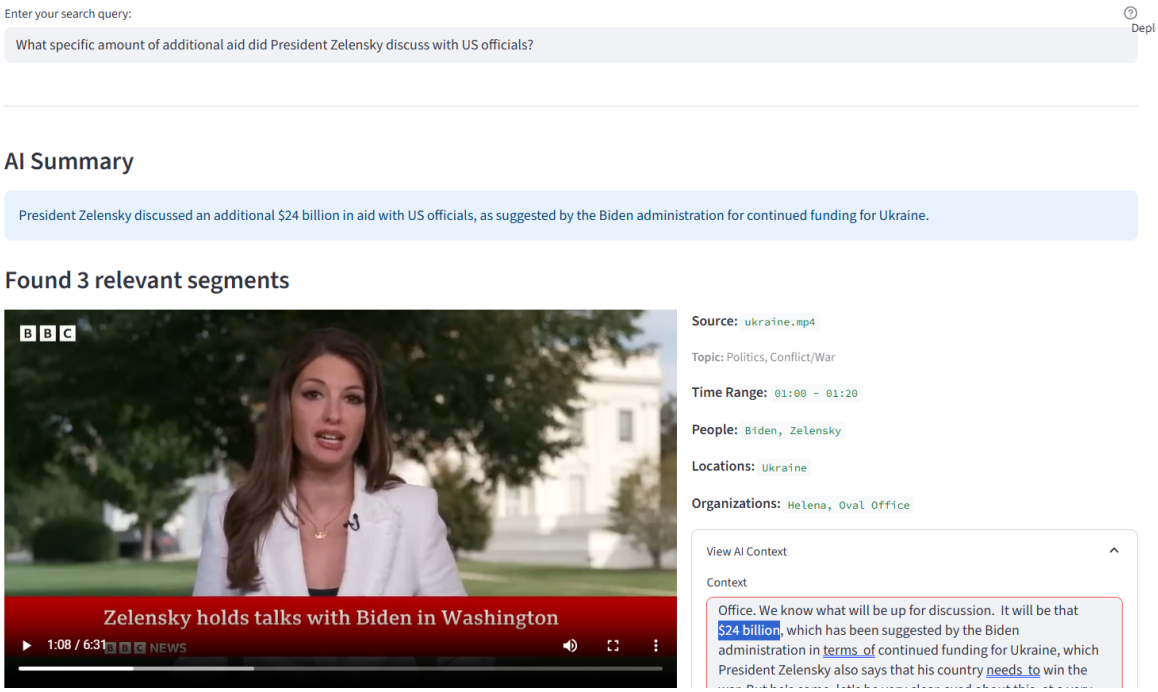

Figure 7: Result for the query on aid amount. The system correctly retrieved the segment discussing the $24 billion package and generated a precise AI answer.

The result confirms that the RAG pipeline successfully identified the keyword "$24 billion" from the transcript and aligned it with the correct video timestamp.

3.2 Visual Understanding (Scene Description)

To evaluate the Vision API's capability to describe context without explicit audio cues, we queried: *"Describe the setting of the interview with Ronaldo."*

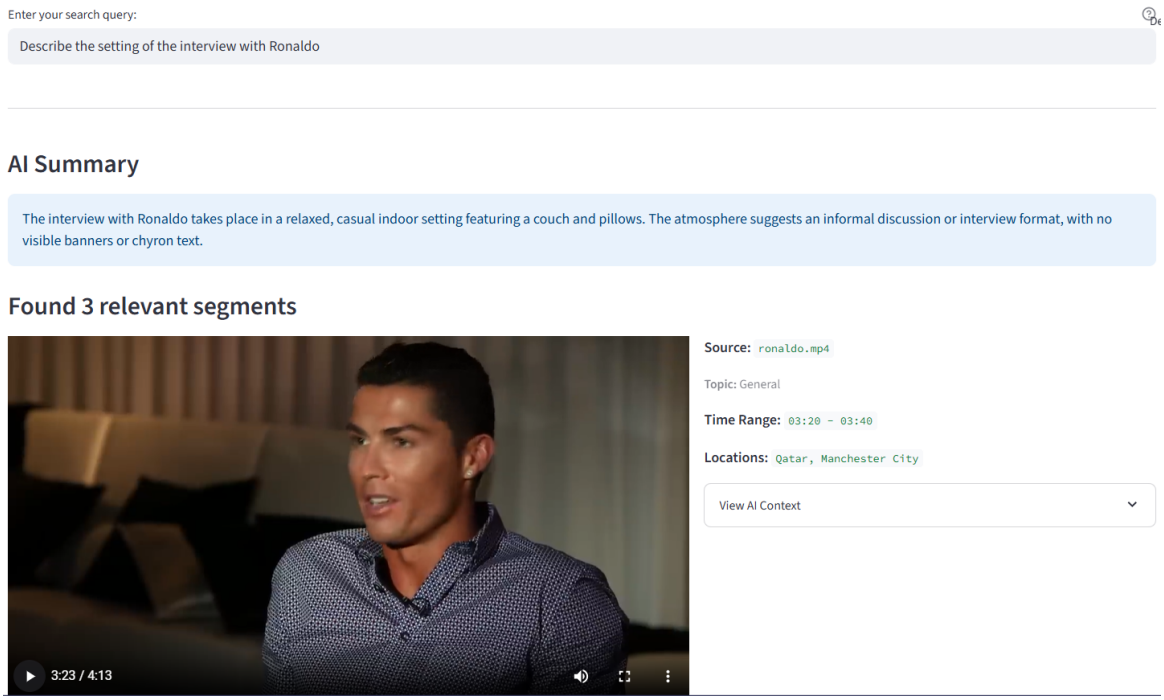

Figure 8: The system retrieved the correct interview segment. The generated answer accurately describes the visual setting (Ronaldo sitting in a formal environment), demonstrating effective multimodal understanding.

3.3 Question Answering

The system's ability to utilize text for contextual reasoning was tested with the query: *"where peace talk takes place"* targeting the "Nagorno-Karabakh" conflict coverage.

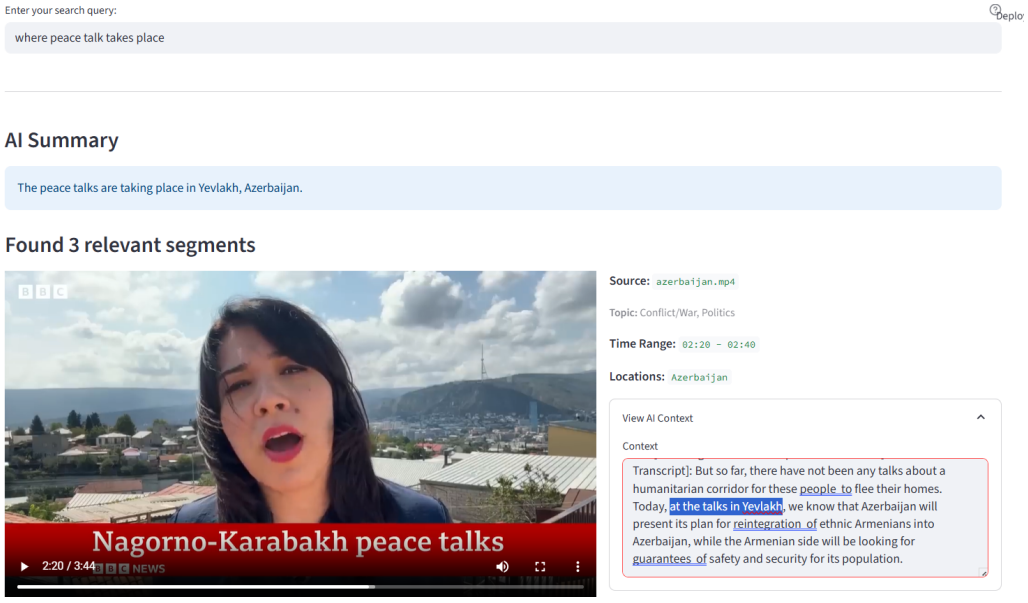

Figure 9: Retrieval based on text. The system successfully identified the location "Yevlakh" mentioned in the audio at to answer the user's question.

3.4 Complex RAG Synthesis

Finally, the system's ability to synthesize a narrative from multiple cues was tested with: *"Why did the Messi event in India descend into chaos?"*

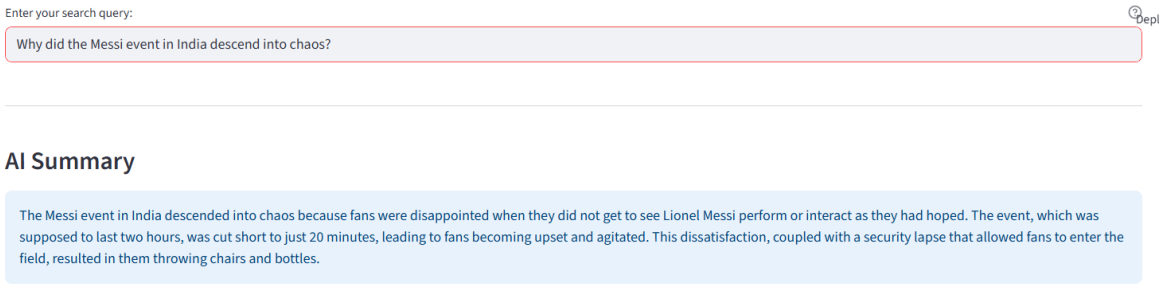

Figure 10: The Retrieval-Augmented Generation (RAG) pipeline. The user query retrieves relevant chunks from the Vector Database, which are then used by the LLM to generate a grounded answer.

3.5 Video Demonstration

To gain a clear understanding of the system's end-to-end workflow, real-time search performance, and user interface interactions, please watch the complete project video demonstration.

Complete System Walkthrough

Click the link below to watch the full demonstration:

WATCH DEMO VIDEO

Direct link: `https://drive.google.com/file/d/1hIz5RV6MguWiORnRcGJlsWFoFWdzQOAI/view`

The demonstration covers:

- **Real-Time Search**: Natural language query processing
- **Multimodal Outputs**: Audio, visual, and text-based analysis
- **User Interface**: Interactive features and navigation flow
- **Performance**: System responsiveness and retrieval accuracy

Duration: 2 minutes 46 seconds
Format: Screen recording with voice narration

4 Discussion

This section analyzes system performance, acknowledges limitations, and proposes pathways for scalability and cost optimization.

4.1 Limitations and Scalability

The current implementation demonstrates multimodal retrieval but faces challenges for production deployment:

- **Processing Latency:** Sequential audio extraction, transcription, and frame analysis should be parallelized using asynchronous workers (e.g., Celery) to reduce time.
- **API Dependency & Cost:** Reliance on OpenAI's suite creates linear cost scaling that becomes prohibitive for large archives.
- **Database Scaling:** Local ChromaDB should migrate to distributed solutions (Pinecone, Weaviate, Milvus) for enterprise-scale datasets.

4.2 Open-Source Alternatives

The architecture allows modular replacement of proprietary APIs with open-source models:

Table 1: Comparison of Current Tech Stack vs. Open-Source Alternatives

Component	**Current Implementation**	**Open-Source Alternative**
LLM	OpenAI GPT-4o (Cloud API)	Meta Llama 3
Visual Analysis	GPT-4o Vision	LLaVA (Large Language-and-Vision Assistant)
Transcription	OpenAI Whisper API	OpenAI Whisper (Local) or WhisperX (Optimized)
Embeddings	text-embedding-3-small	HuggingFace `all-MiniLM-L6-v2`
Vector DB	ChromaDB (Local Persist)	Qdrant or Milvus (Dockerized)

4.3 Future Improvements

Beyond replacing models, the system functionality could be enhanced by:

1. **Speaker Diarization:** Distinguishing between different speakers (e.g., "Anchor" vs. "Interviewee") to allow queries like *"What did the President say?"*.
2. **Face Recognition:** Replacing generic person descriptions with identity matching against a database of known public figures to improve metadata accuracy.
3. **Temporal Aggregation:** Merging consecutive 20-second chunks if they belong to the same semantic topic, creating variable-length segments for smoother playback.

5 Conclusion

This project developed a **Semantic News Video Search** system that bridges raw video data and natural language retrieval. Implementing sliding window segmentation and fusing three modalities preserves context lost in traditional search.

RAG integration transforms search from timestamp lists into intelligent question-answering. Despite API cost and speed limitations, the architecture demonstrates multimodal AI can

unlock knowledge in news archives for journalists, researchers, and media organizations.

6 References

Video Data Sources

1. **Cristiano Ronaldo Interview (BBC News):**
   `https://www.youtube.com/watch?v=86b1wygMpyM`
2. **Lionel Messi in India (BBC News):**
   `https://www.youtube.com/watch?v=F9Yl4JcjBmY`
3. **Zelensky US Visit (BBC News):**
   `https://www.youtube.com/watch?v=WVsgYUlq39M`
4. **Nagorno-Karabakh Conflict (BBC News):**
   `https://www.youtube.com/watch?v=aCQvsoWq35k`