MSML 641 Final Project - YouTube Video Q&A System

December 16, 2025

Rohan Dawkhar, Leonidas Fafoutis, Kaiwei Hsu, Noah Shaw

1 Introduction

This project builds a natural language processing (NLP) system that allows users to ask questions about the content of a YouTube video by providing its URL and a question.[1] The system retrieves the video transcript, preprocesses it using NLP techniques, and answers queries using both a keyword-based baseline and an improved LLM-powered retrieval model.

The goal of the project includes:

- Compare Q&A performance using a baseline approach with an LLM-based retrieval model.
- Analyze system errors and limitations.
- Evaluate how well models handle long, noisy transcripts.
- Demonstrate usability through an optional interface.

The product is delivered via a web interface, which allows users to interact with the model and obtain generated answers to their queries. The report is organized as follows: Section 2 explains the architecture and the design of the system. Section 3 discusses the details of NLP models used in the project. Section 4 explains the web interface and features.

2 Related Work

Question-answering systems have evolved significantly over the past decade, with two dominant paradigms emerging: sparse retrieval methods and dense semantic retrieval approaches.

2.1 Sparse Retrieval Methods

Sparse retrieval techniques like TF-IDF and BM25 have been standard in information retrieval for decades. These methods represent documents and queries as high-dimensional vectors where each dimension corresponds to a term. TF-IDF measures the term importance by combining Term Frequency (how often a word appears in a document) with Inverse Document Frequency.

2.2 Dense Retrieval and Semantic Understanding

The introduction of transformer-based models and BERT enabled dense retrieval approaches. These models encode text into fixed-dimensional vector representations that capture semantic meaning.

2.3 Retrieval-Augmented Generation

Recent work has combined retrieval with generative language models to improve answer quality. Lewis et al. (2020) introduced Retrieval-Augmented Generation (RAG), which retrieves relevant documents and feeds them to a sequence-to-sequence model.

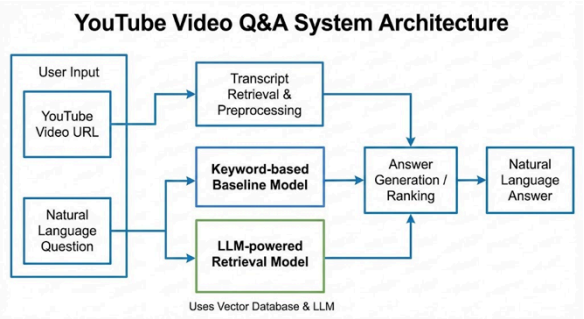
2.4 Our Contribution

This project directly compares sparse retrieval (TF-IDF baseline) with dense retrieval augmented with LLM generation on YouTube video transcripts. Unlike prior work focusing on single methods, our comparative analysis provides empirical evidence of when each approach succeeds or fails. Our work emphasizes practical considerations like handling noisy automatic transcripts and quantifying hallucination in LLM responses.

[1] The GitHub for the project is available at https://github.com/sachelsout/youtube-video-qa?tab=readme-ov-file

3 Architecture

The system architecture is illustrated in the figure below. First, a user provides a URL link of a video of interest and a prompt for a question to be answered about the video. Then, the model retrieves relevant information from the transcript of the video and processes the query using a keyword-based model (TF-IDF) as well as an LLM retrieval model. Using the generated answers, the model ranks the results and outputs the best final answers in response to the query.



The detailed features of the model include

- Automatic YouTube transcript extraction
- Transcript cleaning and segmentation
- Baseline keyword-based QA
- Improved LLM + embeddings retrieval QA
- Quantitative and qualitative evaluation
- Error analysis and ethical considerations
- Modern web interface with dual QA modes, dark mode, and user controls
- Conversational Q&A with separate conversation histories per mode
- Configurable retrieval parameters (1-20 chunks) for flexible results

4 NLP models

4.1 Baseline TF-IDF model

The baseline mode uses the TF-IDF vectorization approach. It is essentially a QA system using keyword matching and TF-IDF retrieval. The central idea is that the model transforms the text in transcripts into a high-dimensional term-frequency vector. This allows the model to identify possible answers by comparing the cosine similarity between the query and the vectorized text content through retrieving the top-k most relevant chunks. Specifically, TF-IDF represents Term Frequency–Inverse Document Frequency. It is a statistical metric that measures the importance of a term within a document relative to a collection of documents. This baseline model provides a benchmark to understand how well a simple NLP can be used to answer questions for YouTube videos.

4.2 LLM model

This model includes an LLM embedding pipeline that prepares video transcripts for semantic retrieval and an LLM-powered model to answer questions.

First, the text from raw transcripts is encoded or segmented into semantically meaningful chunks via a transformer-based embedding model. The pipeline converts unstructured text into structured semantic vectors and allows the LLM-powered model to answer questions in video content. These structured embeddings capture the semantic meaning of each text segment and allow for information search by similarity and context retrieval as opposed to pure keyword matching.

The retrieved context is then converted into a structured prompt that is used to instruct the LLM to answer questions based on the provided relevant context. The prompt includes context, questions, and detailed instructions. Specifically, the prompt first informs the LLM that “You are a retrieval-based assistant answering questions about a YouTube video using ONLY the provided text.” Then, the prompt includes context identified before and the question(s) users provide. Finally, the prompt includes specific instructions to ensure content is well-generated.

In the LLM model, generated content is controlled by the model parameters, such as temperature to control for diversity of generated text and max tokens to control for the length of responses.

5 Web Interface

The product is delivered via a web interface. The features and designs are explained below.

5.1 Features and controls

The model provides a modern Fatsia demo interface. This allows the user to enter a YouTube video URL, and a load button triggers

transcript extraction and preprocessing. The chat interface provides the features below:

- Dual QA Modes: Switch between Baseline (TF-IDF) and LLM (embeddings + model) seamlessly.
- Separate Conversations: Each mode maintains its own conversation history.
- Configurable Retrieval: Use the chunks slider (1–20) to control how many transcript segments are retrieved for each question.
- Video Thumbnail: Displays the video's thumbnail with YouTube link, making it easy to switch between the Q&A and the video.
- Message History: View your questions and the system's responses in a clean chat interface.
- Dark Mode: Toggle between light and dark themes (preference saved to your browser).

The model also provides users with the following controls: An Ask Button allows users to send a question to the current QA mode. A clear Button can be used to clear chat history for the current mode. A New Video Button loads a different YouTube video if users would like to change the video. Finally, a Dark Mode Toggle allows users to switch between light and dark themes.

The API endpoints support the standard GET and POST methods as well:

- GET / - Main web interface
- POST /api/transcribe - Fetch and process YouTube transcript
  - Request: { "video_url": "..." }
  - Response: Video metadata and processing status
- POST /api/ask - Get QA response
  - Request: { "video_id": "...", "question": "...", "mode": "baseline|llm", "chunks_k": 5 }
  - Response: { "answer": "...", "chunks": [...], "mode": "..." }

5.2 Conversation Management

The interface supports rich conversation management.

First, the baseline and LLM modes maintain completely separate conversation histories. Each mode tracks its own question-answer pairs independently. Therefore, switching between modes does not erase users' conversation context.

Second, the interface allows users to control the chunks slider with a value of 1–20, which controls how many transcript segments are retrieved for better or broader context. A lower value, e.g., 1–5, focuses on most relevant answers, whereas a high value, e.g., 10–20, includes more context for comprehensive understanding.

Furthermore, users can reset conversation for the current mode to clear chat histories. They can also load a different video and start fresh (clears both conversation histories).

Finally, the conversation is currently stored in users' browser only (local storage and session memory). The conversations persist during a session. However, closing the browser will clear all conversation data for privacy reasons.

5.3 Technical Details

The frontend is built with:

- HTML5 with Jinja2 templating
- Vanilla JavaScript with state management (appState object)
- CSS3 with CSS variables for light/dark theme support
- Responsive design (mobile-first, supports 320px–1920px widths)

The backend is based on:

- FastAPI with async/await support
- Session-based video storage
- Real-time transcript processing with progress feedback
- Integrated embedding and LLM inference

The styling includes:

- Modern gradient buttons with hover effects
- Smooth dark mode transitions with localStorage persistence
- Animated message appearances
- Mobile-optimized layout with proper spacing and typography

6 Results and Evaluation

6.1 Evaluation Setup and Metrics

We evaluated both systems on 25 question-answer pairs from 4 YouTube videos:

- Video 1: Physics/Science education (6 questions)

- Video 2: Educational content (5 questions)
- Video 3: Educational content (7 questions)
- Video 4: Educational content (7 questions)

Total transcript length: ~8,000 words across all videos. Gold answers were manually curated to be concise (average length: 20 words) and accurate based on video content.
We use three standard metrics:

1. Exact Match (EM): Binary score (1 if normalized prediction equals gold answer, 0 otherwise). Normalized means: lowercase, remove punctuation and articles (a, an, the), remove extra whitespace.
2. F1 Score: Token-level overlap between prediction and gold answer. Computed as: F1 = 2 * (Precision * Recall) / (Precision + Recall) where Precision = # matching tokens / # prediction tokens and Recall = # matching tokens / # gold tokens. Ranges from 0 to 1.
3. ROUGE-L: Longest common subsequence-based metric measuring sequence-level similarity. Less sensitive to token reordering than F1.

6.2 Quantitative Results

Table 1 presents the performance comparison:

Metric	Baseline	LLM	Improvement	% Change
Exact Match	0.0%	4.0%	+4.0 pts	N/A
F1 Score	0.081	0.395	+0.314	+387%
ROUGE-L	0.037	0.302	+0.265	+716%

The LLM model substantially outperforms the baseline across all metrics. The F1 improvement of 387% indicates that while both methods struggle, the LLM produces answers with significantly better token overlap with gold answers. The 4% EM score for LLM (vs 0% for baseline) suggests that semantic understanding enables occasional exact matching, though this remains rare.

6.3 Error Analysis

Beyond quantitative metrics, we conducted detailed error analysis on all 50 predictions (25 questions × 2 models). We categorized errors by severity:

- Critical (8 errors, 16.3%): Complete failure to answer or severe hallucination with no grounding in text
- Major (33 errors, 67.3%): Incomplete answer, missing key information, or hallucination with partial grounding
- Minor (8 errors, 16.3%): Correct semantically but penalized by metrics due to paraphrasing or synonyms

Error distribution by type:

- Incomplete Answer (19, 38.8%): Model retrieves related content but omits key information
- Hallucination (14, 28.6%): Model adds extra or ungrounded information
- Paraphrase Penalty (8, 16.3%): Semantically correct but different wording
- Retrieval Failure (6, 12.2%): Retrieved chunks not relevant to question
- Semantic Mismatch (2, 4.1%): Fundamental misunderstanding

6.4 Metrics Limitation and Paraphrasing

An important observation: 8 errors (16.3% of all errors) are classified as "Minor" because the LLM produced semantically correct answers with different wording than the gold answer. For example:

Gold: "You should run to minimize rain exposure."
LLM: "Running is better because it reduces the time spent in precipitation."

Token overlap metrics (F1, ROUGE-L) penalize such paraphrases as errors even though a user would consider the answer correct. This suggests that the reported metrics (F1=0.395) may underestimate the practical utility of the LLM system. A more nuanced evaluation using semantic similarity (e.g., BERTScore) would better reflect actual answer quality.

7 Future Work

Several directions could improve system performance:

1. Improved Transcription: Using a fine-tuned automatic speech recognition (ASR) model or human-verified transcripts would eliminate transcription errors, likely reducing error rates significantly.
2. Confidence Thresholding: Implement confidence estimation where the system outputs "I don't know" if confidence falls below a threshold, reducing hallucinations.
3. Source Attribution: Require the LLM to cite specific timestamps or transcript excerpts supporting each answer, enabling user verification and reducing hallucination.
4. Larger Evaluation Set: Our evaluation used only 25 questions from 4 videos. A dataset of 500+ questions across 50+ videos spanning diverse genres (education, entertainment, tutorials, news) would enable more robust conclusions.

8 Limitations

The system has some clear limitations that stem mainly from the YouTube transcript quality and our LLM-powered retrieval approaches.

Our system relies solely on automatically generated transcripts that YouTube generates for each video, and it is possible for these transcripts to have errors like incorrect words or missing punctuation etc. Our error analysis has shown that our system does tend to fail if the provided transcript is missing valuable information or is incorrectly represents it, which may lead to incorrect answers even if the video itself contains the correct answer.

Our TF-IDF baseline also has hard failures because it relies on token overlap and cannot capture semantic similarities or other forms of implicit relationships within the tokens. It can only utilize transcripts that contain exact keywords. As a result of this, when faced with more complex or abstract questions, the baseline fails to generate useful responses or reason effectively. These hard failures can even occur when the transcript and information given in the video are clearly semantically relevant to the question being asked.

9 Ethical Considerations

While our model provides a useful function of answering questions submitted by the user on YouTube videos, there are ethical concerns we must take into account as a result of the inherent problems with both the model itself as well as the data. The first issue arises from the automatic transcription process that we receive from YouTube. Another potential issue comes from the way most LLMs generate answers, especially when prompts are vague or the information is not contained within the dataset. These LLM hallucinations can provide incorrect information to the user while presenting it as facts. The dataset used may not reflect all views on certain subjects, resulting in a model that struggles to provide clear and factual information. To mitigate some of these issues, we can either create tools or structure the model to provide the most accurate information. For the automatic transcription issues, the creation of our own, more accurate, automatic transcription tool or a data preprocessing tool would be appropriate.

10 Conclusion

In this project, we design a QA system for YouTube videos. We use several NLP techniques, including TF-IDF, embeddings, prompt engineering, and information retrieval. We deploy the model using a web interface with many useful features to allow users to interact with the system easily. We compare the performance of the TF-IDF baseline model with the LLM retrieval model and find that the LLM-based retrieval model outperforms the traditional baseline model, demonstrating its more powerful semantic understanding of text from video transcripts. We also discuss the limitations and ethical considerations. Using a simple architecture, this project demonstrates the applications of NLP to real-world problems.

11 Recorded Presentation and GitHub Repo Links

- Recorded Presentation: https://www.youtube.com/watch?v=zEiSVOZIPOg
- GitHub Repo: https://github.com/sachelsout/youtube-video-qa

12 References

- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Nayak, N., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems (NeurIPS), volume 33, pages 1877–1901.

- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

- Lewis, P., Perez, E., Piktus, A., Schwenk, H., Schwab, D., Kiela, D., & Schwenk, H. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv preprint arXiv:2005.11401.

- Rajpurkar, P., Zhang, J., Liang, P., & Liang, P. S. (2016). SQuAD: 100,000+ Questions for Machine Reading Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2383–2392.

- Reimers, N., & Gupta, U. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

- Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval, 3(4), 333–389.

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS), volume 30.