VidHarbor: Your Tailored YouTube Voyage

Aaraiz Hassan
23100130

Bisma Nawaz
24100277

Arooba Maqsood
24100235

M. Talha Tariq
25100041

Ayza Shuja
24100106

Shaheer Akhtar
24100203

Abstract

This project is a specialized search engine that lets users ask inquiries of particular YouTube channels. Users can ask any questions to any of the several well-known educational channels from the extensive list that will be provided to them. The responses will take into account the user's selected channel's unique style and context, as well as the content of the videos on that channel. As a result, the initiative significantly improves user interaction by effectively obtaining data from dependable YouTube sources.

1 Introduction

The project aims to transform the way users engage with YouTube channels by creating a specialized educational search platform. By offering an extensive compilation of reliable and carefully selected YouTube channels, the site seeks to close the gap between user searches and pertinent content. Users are guaranteed access to excellent educational resources that are in line with their learning objectives thanks to this carefully curated selection.

The platform's sophisticated search feature, which enables users to ask queries in natural language or use keywords to discover precise answers, is its main selling point. The platform employs cutting-edge natural language processing (NLP) techniques to precisely comprehend user inquiries and associate them with videos that offer the most comprehensive and pertinent responses.

The chat-based interface of the platform simulates a real-world conversation, hence improving user interaction. With the platform, users can interact as they would with an experienced instructor, asking follow-up questions or requesting further explanation. The user's query is catered to by the platform's response, guaranteeing a customized and engaging learning process.

The creative approach of the project has various advantages:

- It gives customers the ability to quickly gather knowledge from reputable YouTube creators. The platform guarantees the quality and reliability of the information by compiling content from reliable sources.
- It provides consumers with a fluid and user-friendly interface, which improves the entire content consumption experience. Users may easily navigate among movies and find the precise information they need thanks to the chat-based interface.
- By giving users equitable access to excellent educational resources, the platform has the ability to democratize education. Through the platform, people may learn at their own pace and according to their own schedules, as time and place are no longer constraints.

2 Implementation Details

2.1 Architecture Overview

Our platform has a microservices architecture, with multiple interconnected components that are specifically designed to process and obtain YouTube videos and transcripts in an efficient manner:

- *Data Ingestion Service:* This service is in charge of obtaining and analyzing YouTube video content from certain channels. The service retrieves transcript and video metadata by utilizing the YouTube Transcript API and the Google API Client Library.
- *Chunking:* Each video's transcript is retrieved sequentially, and then split into

chunks. The chunks have a maximum token size (so they never get too big for the API to handle later).

- *Embeddings Generation:* After the chunks are made, their embeddings are generated using OpenAI's 'text-embedding-ada-002' model. These embeddings, which are dense vector representations of textual data, are used to capture semantic commonalities between the different chunks, which will be relevant later when we have to retrieve specific chunks based on a user's query.

- *Storage in ChromaDB:* ChromaDB is a database client designed specifically for storing and querying high-dimensional vector data, and it is where the created embeddings are kept. ChromaDB was chosen since it facilitates quick and scalable similarity searches by effectively storing document embeddings and their associated metadata.

- *Query Handling Service:* This system processes and handles user queries. It first generates an embedding for the queries using the same 'text-embedding-ada-002' model that was used for storing our data (for the sake of consistency). Then, it compares this embedding with the ones currently in our ChromaDB using similarity search techniques via the ChromaDB client's pre-built 'query' method. It then returns a specific number (which can be changed) of chunks that are closest to the query vector and thereby will be the most relevant for GPT to answer with.

- *User Interface:* The User Interface offers the platform's front-end user interface through which users can interact. Constructed utilizing contemporary web technologies like React, the interface provides user-friendly search features and shows search results that are derived from Query Handling Service-performed similarity searches.

2.2 Backend Implementation

The following are the tools used for our backend implementation:

- *Programming Languages:* Python

- *Framework:* Flask

- *Database:* MongoDB

- *YouTube Data API:* Used for fetching video content, metadata, and transcripts.

- *Embeddings:* OpenAI's text-embedding-ada-002 model for generating embeddings.

- *Model:* GPT-4 for generating responses to user queries.

- *Backend Functionality:* Following the creation of embeddings and their database storage, the backend uses a certain procedure to obtain pertinent data in response to user queries:
  - *Database query:* When a user submits a question, the backend first uses the user query to generate an embedding for the query and then query the database in order to obtain relevant chunks from the transcripts that have been saved.
  - *Model Interaction:* The application specifies a predetermined prompt format in which the retrieved paragraphs are provided to the model. This is a system level prompt for GPT, and it gives the model instructions on how to use the data from the retrieved paragraphs to provide a succinct and educational response to the user's query.
  - *Response Generation:* To produce a final response, the model analyzes the request and the paragraphs it has retrieved. The purpose of this response is to fully address the user's query by giving pertinent information in a clear and comprehensible manner.
  - *Response Delivery:* Lastly, the query-response cycle is concluded by sending the created response back to the user via the frontend interface. By using the content of the transcripts that have been stored, this procedure guarantees that consumers will obtain precise and educational responses to their inquiries.

2.3 Frontend Implementation

The following are the tools used for our frontend implementation:

- *Programming Languages:* Javascript

- *Framework:* Flask

2.4 Data Extraction

Our video transcripts (5 from each channel) were extracted from 9 popular YouTube channels, including:

- Programming with Mosh (https://www.youtube.com/@programmingwithmosh)
- TEDx Talks (https://www.youtube.com/@TEDx)
- Y Combinator (https://www.youtube.com/@ycombinator)
- BroCodez (https://www.youtube.com/@BroCodez)
- freeCodeCamp (https://www.youtube.com/@freecodecamp)
- Clever Programmer (https://www.youtube.com/@CleverProgrammer)
- Net Ninja (https://www.youtube.com/@NetNinja)
- Lex Fridman (Podcasts on Tech) (https://www.youtube.com/@lexfridman)
- Huberman Lab (Neuroscience and Science-based Tools) (https://www.youtube.com/@hubermanlab)

3 Experiments and Iterations

3.1 Prototype Development

- *Functional Prototype Creation:* Our project started with the creation of a functional prototype that showcased the platform's primary features adapted for the educational content industry. The prototype was designed to showcase key functionalities like transcript processing, data intake from specific educational YouTube channels, and search functions tailored to educational content.
- *Feature Prioritization:* Features were arranged in order of importance to the user's demands and the educational setting. To assure the platform's fundamental capabilities were not compromised, core features such as creating embeddings for semantic search and retrieving video transcripts from well-known educational channels were built first.
- *Technical Feasibility Evaluation:* During the prototype development phase, a thorough assessment of technical feasibility was conducted, taking into account elements like the integration of the YouTube API, the extraction of transcripts, and the creation of embeddings for instructional content. Through iterative testing and technical development, issues like transcript availability, content diversity, and domain-specific search relevance were resolved.
- *Proof of Concept Validation:* The prototype was used to verify the viability and efficacy of the suggested solution in the field of education. Early testing with transcript processing, customized search functions, and data importation from educational YouTube channels gave important insights into how the platform may expedite content discovery and knowledge acquisition for instructors and students.

3.2 Experimentation and Refinement

- *Model Selection:* The brains of our software is the model we use that will be responsible for understanding the reference passages provided and generating an answer. Choosing the right model was thus paramount. Initially we used Gemini, but it was not able to answer some questions even when the context was available in the transcripts (this had a lot to do with how the embeddings were stored as well, mentioned below). Switching to GPT-4 with OpenAI's ecosystem made things significantly better and improved results.
- *Embeddings Generation:* A big problem with the embeddings used with Gemini was that the embeddings were not being chunked properly. Entire transcript documents of long YouTube videos were being turned into a single embedding, and then being retrieved and sent to the model. This obviously caused problems with the model not being able to handle such huge token sizes. Therefore, when we switched to GPT, we also started properly tokenizing documents, and then chunking these tokens together (with maximum size limits), and generating embeddings for the chunks. This meant that chunks retrieved later would have much more relevant context to the query and be easier for the API to handle.

- *Model Fine-tuning with Prompts:* Through experimentation, several prompts were tried in an effort to get the best possible replies from the model. We experimented with several prompts that were customized for the educational domain and user queries in order to maximize the caliber and pertinence of the model's output. The final prompt format that we settled on included the following instructions:

  ”You are a helpful and informative bot that answers questions primarily using information from the reference passages provided. Please note that the reference passages might have some typos and incorrect grammar. Focus on the information provided in the reference passages, but when needed, you can use your own knowledge too. Your audience may be non-technical, so try to break down complicated concepts and use analogies where possible. In your answer, do not mention referring to any passages or a database. If the question is unclear or there are multiple possible interpretations, ask the user for clarification.”
  Question: (add query asked by the user)
  Reference Passages: (include all relevant passages fetched by querying our database)

  In addition to ensuring consistency in model responses, this structured prompt style supplied contextual information needed to produce pertinent and educational responses that were customized to user queries and reference passages.

3.3 Future Directions

- *Initial Prototype Refinement:* Since the application is still in its nascent phases, the main priority will be to address important usability concerns, boost functionality, and improve user experience by improving upon our initial prototype. Early user feedback will be gathered, crucial pain points will be identified, and product upgrades will be prioritized to meet user expectations and needs through iterative refinement cycles.

- *Extensive Testing:* To verify the application's performance, dependability, and functionality across a range of use cases and user situations, extensive testing will be carried out. This includes performance testing to evaluate system stability and responsiveness under various load scenarios, compatibility testing across various devices and browsers, and functional testing to make sure all features function as intended.

- *User Feedback Collection:* An important part of our future development activities will be talking to users to find out their experiences in using our program. In order to obtain information about user preferences, problems, and suggestions for improvement, feedback gathering tools like surveys, feedback forms, and user testing sessions will be used. Iterative improvements and the priority of further feature development will be guided by user feedback.

- *Addition of Chat History:* The inclusion of a chat history function will allow users to go back and examine their prior exchanges with the application, including the questions and responses they sent during those sessions. By allowing users to track their progress, going back and easily picking up where they left off in past chats, this feature increases user engagement.

- *Implementation of Timestamp Feature:* The incorporation of a timestamp function will facilitate user access to particular timestamps within video content that correspond directly to their search queries. By enabling rapid access to pertinent video content areas, this feature improves user experience and content navigation while promoting effective content consumption and knowledge acquisition.

- *User-Centric Design:* Throughout the development process, it will be crucial to uphold a user-centric design philosophy. User research, usability testing, and feedback analysis will inform design choices in order to maintain the application's intuitiveness, accessibility, and appeal to its intended user base.

- *Iterative Development Approach:* Adopting an iterative development approach will enable ongoing innovation and improvement in response to changing market dynamics and consumer needs. Every development iteration will concentrate on improving the application's long-term vision and goals, providing users with incremental value, and responding to feedback from earlier iterations.

- *Advanced NLP Techniques:* In order to improve the platform's comprehension and processing of instructional content, future versions will investigate the incorporation of sophisticated natural language processing (NLP) techniques. Using cutting-edge natural language processing (NLP) models, such as transformer-based architectures like BERT, GPT, or variations designed for educational text interpretation, is one way to achieve this. Furthermore, investigating methods like sentiment analysis, entity recognition, and summarizing could improve the platform's capacity for content summarization and semantic comprehension.

4 Reflections

4.1 Successes

- *Efficient Content Discovery:* In order to solve the problem of finding relevant content quickly on YouTube, a specialized search platform was created. This aims to provide consumers with a quick and easy way to find important information from producers of educational content in a matter of minutes.

- *Enhanced User Interaction:* The incorporation of question-answering or chat-based features will enable users to obtain knowledge and perspectives from reputable YouTube creators in a more participatory and captivating way.

- *Improved Information Retrieval:* The platform promises to improve the accuracy of information retrieval by utilizing embeddings and sophisticated natural language processing algorithms. This will allow users to find pertinent segments of video content that directly answer their inquiries with ease.

4.2 Challenges

- *Model Fine-tuning Complexity:* It proved quite difficult to capture the subtleties of user inquiries and instructional content, therefore it took repeated trial and refinement to fine-tune the model to produce relevant results.

- *Scalability Considerations:* Developing the architecture and infrastructure of the platform presented problems in anticipating future scalability requirements and guaranteeing optimal performance under growing user demands.

- *Data Coverage:* Although transcripts from a variety of channels have been extracted, it is still difficult to provide thorough coverage of all pertinent subjects. Increasing the number of channels and subjects in the dataset is crucial to enhancing the platform's utility.

4.3 Lessons Learned

- *Iterative Development Approach:* Using an iterative development method made it possible to remain adaptive and flexible and fine-tune platform features in response to experimentation and feedback from the TAs.

- *Early Planning and Prototyping:* In the early phases of development, it became clear how important it was to plan ahead and conduct comprehensive prototyping. The construction of prototypes and a well-defined project roadmap facilitated the visualization of the features and functionalities of the application, providing a solid basis for subsequent iterations of development.

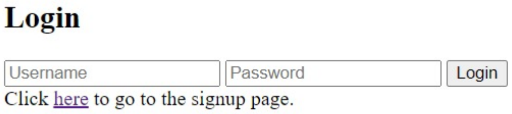
- *Ongoing Education:* The software development process functioned as an educational tool, offering chances to investigate novel technologies, approaches, and optimal techniques in the field. Team members' shared knowledge and ongoing education helped to improve the project and create new skills.

- *User-Centric Design:* In order to create a platform that connected with its target audience and catered to their unique needs and preferences, it proved essential to prioritize user feedback and include user-centric design concepts.

- *Collaborative Development:* Over the course of the development lifecycle, our team members' cooperation and communication were essential to overcoming obstacles and advancing the project.

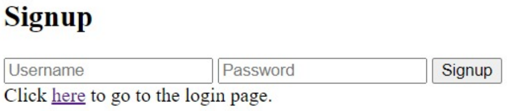
5 Project Demonstration

The following is a brief walkthrough of our application.



Click here to go to the signup page.

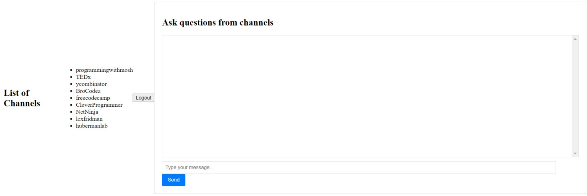
This is the login screen which allows an already registered user to login using their credentials.



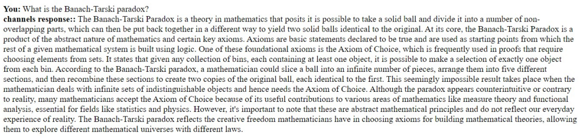
Click here to go to the login page.

This is the signup screen which allows a new user to register.

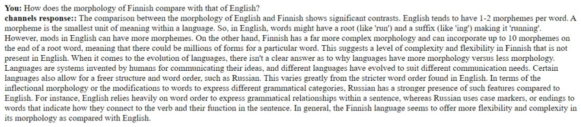
Once the user has successfully logged in, the user is redirected to the main screen.



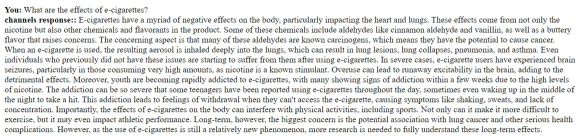
This screen shows the list of channels and a window where prompts can be written and answers will be displayed.



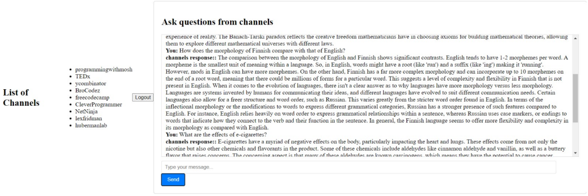
You: What is the Banach-Tarski paradox?
channels response:: The Banach-Tarski Paradox is a theory in mathematics that posits it is possible to take a solid ball and divide it into a number of non-overlapping parts, which can then be put back together in a different way to yield two solid balls identical to the original. At its core, the Banach-Tarski Paradox is a product of the abstract nature of mathematics and certain key axioms. Axioms are basic statements declared to be true and are used as starting points from which the rest of a given mathematical system is built using logic. One of these foundational axioms is the Axiom of Choice, which is frequently used in proofs that require choosing elements from sets. It states that given any collection of bins, each containing at least one object, it is possible to make a selection of exactly one object from each bin. According to the Banach-Tarski paradox, a mathematician could slice a ball into an infinite number of pieces, arrange them into five different sections, and then recombine these sections to create two copies of the original ball, each identical to the first. This seemingly impossible result takes place when the mathematician deals with infinite sets of indistinguishable objects and hence needs the Axiom of Choice. Although the paradox appears counterintuitive or contrary to reality, many mathematicians accept the Axiom of Choice because of its useful contributions to various areas of mathematics like measure theory and functional analysis, essential for fields like statistics and physics. However, it's important to note that these are abstract mathematical principles and do not reflect our everyday experience of reality. The Banach-Tarski paradox reflects the creative freedom mathematicians have in choosing axioms for building mathematical theories, allowing them to explore different mathematical universes with different laws.



You: How does the morphology of Finnish compare with that of English?
channels response:: The comparison between the morphology of English and Finnish shows significant contrasts. English tends to have 1-2 morphemes per word. A morpheme is the smallest unit of meaning within a language. So, in English, words might have a root (like 'run') and a suffix (like 'ing') making it 'running'. However, mods in English can have more morphemes. On the other hand, Finnish has a far more complex morphology and can incorporate up to 10 morphemes on the end of a root word, meaning that there could be millions of forms for a particular word. This suggests a level of complexity and flexibility in Finnish that is not present in English. When it comes to the evolution of languages, there isn't a clear answer as to why languages have more morphology versus less morphology. Languages are systems invented by humans for communicating their ideas, and different languages have evolved to suit different communication needs. Certain languages also allow for a freer structure and word order, such as Russian. This varies greatly from the stricter word order found in English. In terms of the inflectional morphology or the modifications to words to express different grammatical categories, Russian has a stronger presence of such features compared to English. For instance, English relies heavily on word order to express grammatical relationships within a sentence, whereas Russian uses case markers, or endings to words that indicate how they connect to the verb and their function in the sentence. In general, the Finnish language seems to offer more flexibility and complexity in its morphology as compared with English.



You: What are the effects of e-cigarettes?
channels response:: E-cigarettes have a myriad of negative effects on the body, particularly impacting the heart and lungs. These effects come from not only the nicotine but also other chemicals and flavorants in the product. Some of these chemicals include aldehydes like cinnamon aldehyde and vanillin, as well as a buttery flavor that raises concerns. The concerning aspect is that many of these aldehydes are known carcinogens, which means they have the potential to cause cancer. When an e-cigarette is used, the resulting aerosol is inhaled deeply into the lungs, which can result in lung lesions, lung collapses, pneumonia, and asthma. Even individuals who previously did not have these issues are starting to suffer from them after using e-cigarettes. In severe cases, e-cigarette users have experienced brain seizures, particularly in those consuming very high amounts, as nicotine is a known stimulant. Overuse can lead to runaway excitability in the brain, adding to the detrimental effects. Moreover, youth are becoming rapidly addicted to e-cigarettes, with many showing signs of addiction within a few weeks due to the high levels of nicotine. The addiction can be so severe that some teenagers have been reported using e-cigarettes throughout the day, sometimes even waking up in the middle of the night to take a hit. This addiction leads to feelings of withdrawal when they can't access the e-cigarette, causing symptoms like shaking, sweats, and lack of concentration. Importantly, the effects of e-cigarettes on the body can interfere with physical activities, including sports. Not only can it make it more difficult to exercise, but it may even impact athletic performance. Long-term, however, the biggest concern is the potential association with lung cancer and other serious health complications. However, as the use of e-cigarettes is still a relatively new phenomenon, more research is needed to fully understand these long-term effects.



experience of reality. The Banach-Tarski paradox reflects the creative freedom mathematicians have in choosing axioms for building mathematical theories, allowing them to explore different mathematical universes with different laws.
You: How does the morphology of Finnish compare with that of English?
channels response:: The comparison between the morphology of English and Finnish shows significant contrasts. English tends to have 1-2 morphemes per word. A morpheme is the smallest unit of meaning within a language. So, in English, words might have a root (like 'run') and a suffix (like 'ing') making it 'running'. However, mods in English can have more morphemes. On the other hand, Finnish has a far more complex morphology and can incorporate up to 10 morphemes on the end of a root word, meaning that there could be millions of forms for a particular word. This suggests a level of complexity and flexibility in Finnish that is not present in English. When it comes to the evolution of languages, there isn't a clear answer as to why languages have more morphology versus less morphology. Languages are systems invented by humans for communicating their ideas, and different languages have evolved to suit different communication needs. Certain languages also allow for a freer structure and word order, such as Russian. This varies greatly from the stricter word order found in English. In terms of the inflectional morphology or the modifications to words to express different grammatical categories, Russian has a stronger presence of such features compared to English. For instance, English relies heavily on word order to express grammatical relationships within a sentence, whereas Russian uses case markers, or endings to words that indicate how they connect to the verb and their function in the sentence. In general, the Finnish language seems to offer more flexibility and complexity in its morphology as compared with English.
You: What are the effects of e-cigarettes?
channels response:: E-cigarettes have a myriad of negative effects on the body, particularly impacting the heart and lungs. These effects come from not only the nicotine but also other chemicals and flavorants in the product. Some of these chemicals include aldehydes like cinnamon aldehyde and vanillin, as well as a buttery flavor that raises concerns. The concerning aspect is that many of these aldehydes are known carcinogens, which means they have the potential to cause cancer.

These are three examples of prompts given to our model. It shows that our model is working perfectly fine and is integrated with the frontend.

6 Member Contribution

- **Aaraiz Hassan:** Worked on the frontend implementation and integrating the backend and frontend.
- **Arooba Maqsood:** Worked on the backend implementation and the project report.
- **Ayza Shuja:** Worked on the backend implementation and the project report.
- **Bisma Nawaz:** Worked on the backend implementation and the presentation poster.
- **M. Talha Tariq:** Worked on the frontend implementation and fixing issues in integration.
- **Shaheer Akhtar:** Worked on fixing issues in the backend implementation and integration with the frontend.