



0 stars

18 forks

1 watching

Branches

Activity

Custom properties

Tags

Public repository

1 Branch

0 Tags

Go to file

t

Go to file

+

Add file

Code

sandrabosk

initial commit

c3d297c · last year

README.md

initial commit

last year



Project III | Business Case: Building a Multimodal AI ChatBot for YouTube Video QA

Building a chatbot that can translate YouTube videos into text and allow for natural language querying offers several compelling business cases.

- Firstly, it improves accessibility for users with hearing impairments or those who prefer reading over watching videos, thereby broadening the audience reach and enhancing brand reputation.
- Secondly, it enables efficient indexing and searching of video content, allowing users to quickly find specific information within videos, which is particularly useful for educational content and tutorials.
- Thirdly, it improves customer support by leveraging existing video content to provide instant, accurate responses to customer queries, thus reducing support costs and improving response times.
- Additionally, it serves educational and training purposes by enhancing the learning experience, enabling easy querying and access to specific segments of instructional videos. From an SEO and content marketing perspective, video transcripts can significantly boost website traffic and video discoverability by improving search engine indexing. Lastly, supporting multiple languages allows the chatbot to cater to a global audience, expanding market reach and enhancing user engagement.

Project Overview

The goal of this final project is to develop a RAG system or AI bot that combines the power of text and audio processing to answer questions about YouTube videos. The bot will utilize natural language processing (NLP) techniques and speech recognition to analyze both textual and audio input, extract relevant information from YouTube videos, and provide accurate answers to user queries.

Key Objectives

1. Develop a text-based question answering (QA) model using pre-trained language models. You may find it useful to fine-tune your model.
2. Integrate speech recognition capabilities to convert audio/video input (user questions) into text transcripts.
3. Build a conversational interface for users to interact with the bot via text or voice input. The latter is not a must.
4. Retrieve, analyze, and store into a vector database (pinecone, chromabd...) YouTube video content to generate answers to user questions.
5. Test and evaluate the bot's performance in accurately answering questions about YouTube videos.
6. Your AI must use agents with several tools and memory.

Incorporating LangChain & LangSmith

To enhance the project with LangChain, you can utilize LangChain agents and functions for various tasks:

1. Text Preprocessing:

- Use LangChain functions for tokenization, lemmatization, and other text preprocessing tasks as you see fit.

2. QA Model Development:

- Utilize LangChain agents for fine-tuning pre-trained language models from HuggingFace or OpenAI for question answering tasks. If you use OpenAI api key provide by Ironhack, be mindful of the limited credits available for the entire class.

3. Speech Recognition Integration:

- Incorporate LangChain agents for integrating speech recognition capabilities into the bot, allowing it to process audio and/or text inputs.

4. Conversational Interface:

- Design conversational flows using LangChain agents to handle user interactions and route queries to the appropriate processing modules.

5. YouTube Video Retrieval:

- Develop LangChain agents for accessing YouTube video content and extracting relevant metadata for analysis.

6. Make use of a vector database of your choice

7. Make use of LangSmith platform for testing, evaluation, and deployment of your AI

Deliverables

1. Source code for the multimodal bot implementation, including LangChain integration.
2. Documentation detailing the project architecture, methodology, and LangChain usage.
3. Presentation slides summarizing the project objectives, process, and results.

4. This must be deployed as a web/mobile app.

Project Timeline

- Day 1-2: Project kickoff, data collection, and text preprocessing using LangChain functions.
- Day 3-4: QA model development with LangChain agents and speech recognition integration.
- Day 5-6: Conversational interface development with LangChain agents and YouTube video retrieval.
- Day 7: Testing, evaluation, documentation, and presentation preparation.

Resources

Below are some useful resources, but you don't have to use them.

- [YouTube](#) Python [module](#).
- [Whisper LLM](#)
- Pre-trained language models available in libraries like [HuggingFace](#) Transformers.
- [LangChain](#) for text preprocessing, model development, and conversational interface design.
- [LangSmith](#) for testing, performance checks, and [deploying](#) your model and app.

Evaluation Criteria

- Accuracy of the bot in answering user questions about YouTube videos.
- Usability and responsiveness of the conversational interface (latency).
- Documentation quality and clarity of presentation slides.

Conclusion

This final project offers an exciting opportunity to explore the intersection of NLP, speech recognition, and multimedia analysis in building a multimodal bot for YouTube video QA. By leveraging state-of-the-art techniques and technologies, including LangChain, you will gain valuable hands-on experience in developing innovative AI applications with real-world impact. You may find it to focus on my topic like health, nutrition, astrophysics...but this is an open ended project in some ways. So, feel free to build something you will be

Releases

No releases published

Packages

No packages published