AI/ML: GENAI VIDEO SEARCH ENGINE

SAMSUNG PRISM-THAPAR

Team:

Mentors:
- Ayush Srivastava
- Sanjeev Kumar
- Himanshu Shekhar

Professors:
- Dr. Sanjeev Rao
- Dr. Nitin Arora

Students:
- Prarthana Aggarwal
- Sneha
- Aadya
- Aditya Kumar Jha

INTRODUCTION

This project demonstrates the creation of a semantic search application using text embeddings and AI. Designed for an education startup, the application enables students to search through a large collection of AI-related YouTube videos by typing questions.

Using OpenAI's Embedding API, video transcripts are converted into numerical representations (vectors) for semantic search. The application returns relevant videos and directs users to specific timestamps containing the answers, enhancing learning efficiency.

KEY FEATURES

01

Understanding the **meaning of queries** (semantic search).

02

Providing **quick and precise results** with links to video segments.

03

Handles complex queries with better context matching compared to traditional keyword-based search.

04

Scalable design that can accommodate expanding video libraries and future AI enhancements.

GOALS

Understand

Gain a thorough understanding of the problem statement and the steps involved in solving it.

Research

Study semantic search concepts, embedding models and vector search tools. Explore areas already developed and collect new research material.

Design

Develop the best solution after discussing with mentors.

LITERATURE SURVEY AND STUDY

Text Embeddings and Embedding index

Text embeddings are semantic numerical representations of text. Embeddings are used to represent data in a way that is easy for a machine to understand. There are many models for building text embeddings, in this lesson, we will focus on generating embeddings using the OpenAI Embedding Model. We'd pass the text to the OpenAI Embedding API and it would return the following embedding consisting of 1536 numbers aka a vector. Each number in the vector represents a different aspect of the text.

1. The transcript for each YouTube video in the AI Show playlist is downloaded.
2. Using OpenAI Functions, an attempt is made to extract the speaker name from the first 3 minutes of the YouTube transcript. The speaker name for each video is stored in the Embedding Index named embedding_index_3m.json.
3. The transcript text is then chunked into 3 minute text segments. The segment includes about 20 words overlapping from the next segment to ensure that the Embedding for the segment is not cut off and to provide better search context.
4. Each text segment is then passed to the OpenAI Chat API to summarize the text into 60 words. The summary is also stored in the Embedding Index embedding_index_3m.json.
5. Finally, the segment text is passed to the OpenAI Embedding API. The Embedding API returns a vector of 1536 numbers that represent the semantic meaning of the segment. The segment along with the OpenAI Embedding vector is stored in an Embedding Index embedding_index_3m.json.

LITERATURE SURVEY AND STUDY

Understanding cosine similarity

Cosine similarity is a measure of similarity between two vectors, you'll also hear this referred to as nearest neighbor search. To perform a cosine similarity search you need to vectorize for query text using the OpenAI Embedding API. Then calculate the cosine similarity between the query vector and each vector in the Embedding Index. Remember, the Embedding Index has a vector for each YouTube transcript text segment. Finally, sort the results by cosine similarity and the text segments with the highest cosine similarity are the most similar to the query.

From a mathematic perspective, cosine similarity measures the cosine of the angle between two vectors projected in a multidimensional space. This measurement is beneficial, because if two documents are far apart by Euclidean distance because of size, they could still have a smaller angle between them and therefore higher cosine similarity. For more information about cosine similarity equations, see Cosine similarity.

PROPOSED SOLUTION

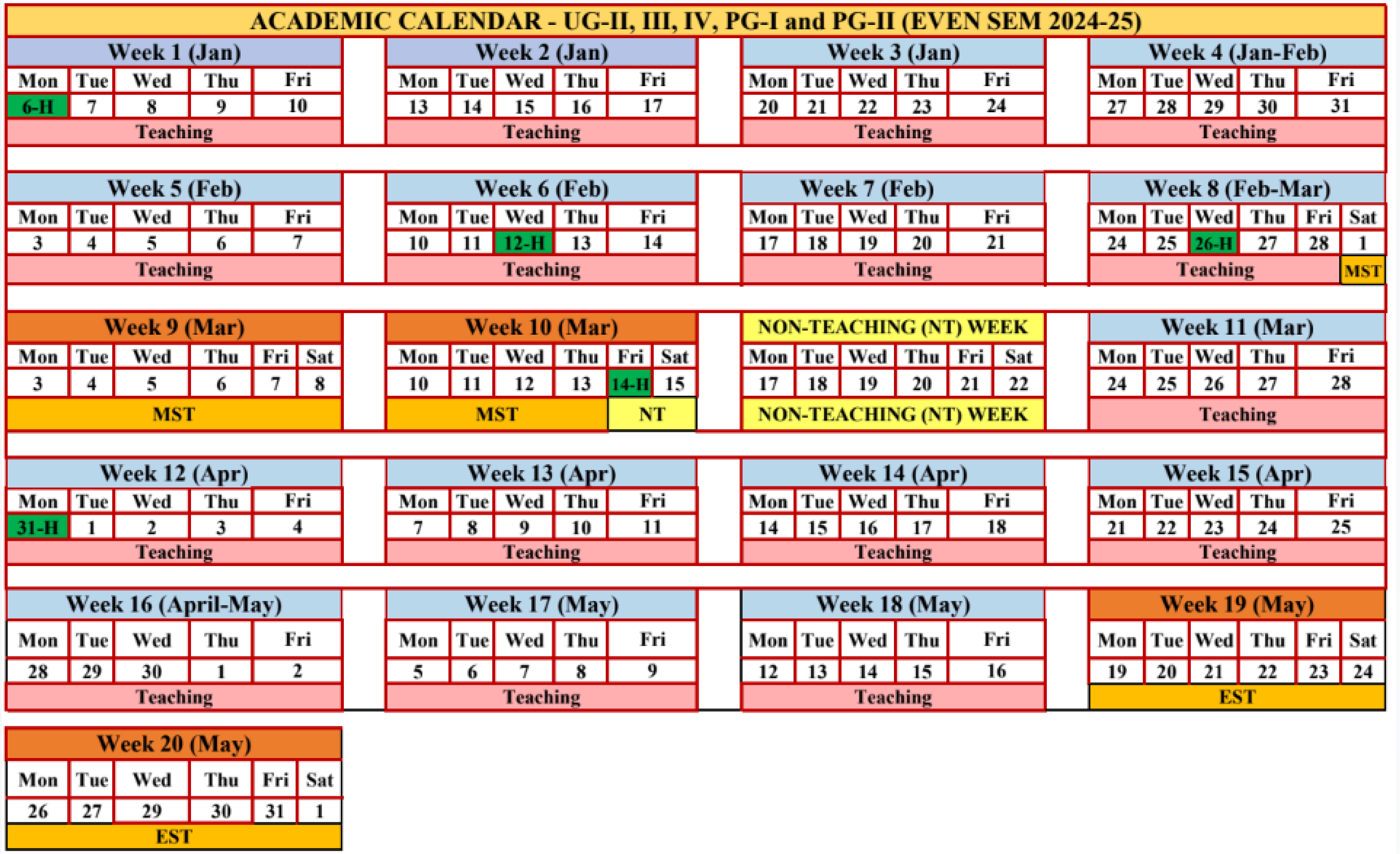
1. Input Processing
   - Goal: Understand the user's question.
   - Use Natural Language Processing (NLP) to analyze the query for semantic meaning, focusing on intent and key concepts.

2. Video Data Preparation
   - Extract Data:
     - Retrieve transcripts and metadata (e.g., titles, descriptions) from videos using YouTube APIs or similar tools.
     - Split transcripts into smaller, timestamped segments for precise targeting.
   - Generate Representations:
     - Convert text segments into numerical embeddings using text embedding models, representing their semantic meaning.

3. Indexing and Storage
   - Goal: Organize data for efficient search.
   - Store embeddings, timestamps, and metadata in a searchable index, optimized for speed using tools like a vector database or indexing service.

4. Query Matching and Ranking
   - Search Mechanism:
     - Convert the user’s question into an embedding vector.
     - Calculate similarity scores between the query vector and stored embeddings using cosine similarity or equivalent methods.
   - Rank Results:
     - Rank video segments based on relevance to the query.
     - Consider additional factors like view count, likes, video age, and metadata relevance for better ranking.

ACADEMIC CALENDER



ACADEMIC CALENDAR - UG-II, III, IV, PG-I and PG-II (EVEN SEM 2024-25)

Week 1 (Jan)

Mon	Tue	Wed	Thu	Fri
6-H	7	8	9	10
Teaching				

Week 2 (Jan)

Mon	Tue	Wed	Thu	Fri
13	14	15	16	17
Teaching				

Week 3 (Jan)

Mon	Tue	Wed	Thu	Fri
20	21	22	23	24
Teaching				

Week 4 (Jan-Feb)

Mon	Tue	Wed	Thu	Fri
27	28	29	30	31
Teaching				

Week 5 (Feb)

Mon	Tue	Wed	Thu	Fri
3	4	5	6	7
Teaching				

Week 6 (Feb)

Mon	Tue	Wed	Thu	Fri
10	11	12-H	13	14
Teaching				

Week 7 (Feb)

Mon	Tue	Wed	Thu	Fri
17	18	19	20	21
Teaching				

Week 8 (Feb-Mar)

Mon	Tue	Wed	Thu	Fri	Sat
24	25	26-H	27	28	1
Teaching					MST

Week 9 (Mar)

Mon	Tue	Wed	Thu	Fri	Sat
3	4	5	6	7	8
MST					

Week 10 (Mar)

Mon	Tue	Wed	Thu	Fri	Sat
10	11	12	13	14-H	15
MST				NT	

NON-TEACHING (NT) WEEK

Mon	Tue	Wed	Thu	Fri	Sat
17	18	19	20	21	22
NON-TEACHING (NT) WEEK					

Week 11 (Mar)

Mon	Tue	Wed	Thu	Fri
24	25	26	27	28
Teaching				

Week 12 (Apr)

Mon	Tue	Wed	Thu	Fri
31-H	1	2	3	4
Teaching				

Week 13 (Apr)

Mon	Tue	Wed	Thu	Fri
7	8	9	10	11
Teaching				

Week 14 (Apr)

Mon	Tue	Wed	Thu	Fri
14	15	16	17	18
Teaching				

Week 15 (Apr)

Mon	Tue	Wed	Thu	Fri
21	22	23	24	25
Teaching				

Week 16 (April-May)

Mon	Tue	Wed	Thu	Fri
28	29	30	1	2
Teaching				

Week 17 (May)

Mon	Tue	Wed	Thu	Fri
5	6	7	8	9
Teaching				

Week 18 (May)

Mon	Tue	Wed	Thu	Fri
12	13	14	15	16
Teaching				

Week 19 (May)

Mon	Tue	Wed	Thu	Fri	Sat
19	20	21	22	23	24
EST					

Week 20 (May)

Mon	Tue	Wed	Thu	Fri	Sat
26	27	29	30	31	1
EST					

THANK YOU