

# **Shazam Clone for Subtitle Search**

## **Building an Intelligent Media Search Engine**

### **Introduction: The Vision Behind the Project**

Imagine watching a movie or listening to a podcast, and suddenly, you hear a line that grabs your attention. You want to find out where else that line has been used or find similar dialogues from your favorite shows. Inspired by the magic of Shazam for music, we set out to build a Shazam Clone for Subtitle Search - a powerful application capable of identifying spoken content from audio or video files and instantly linking it to matching subtitle chunks from a vast database.

Our goal was to create a tool that could take any spoken content, analyze it, and retrieve the most relevant subtitles from a database of subtitle files. This required a blend of natural language processing (NLP), machine learning (ML), and audio processing technologies.

### **Technology Stack: Libraries That Brought Our Vision to Life**

#### **Python:**

Purpose: The main programming language due to its flexibility and vast ecosystem of libraries for NLP, ML, and data processing.

#### **Streamlit:**

Purpose: Building interactive and responsive web applications.

#### **Pandas:**

Purpose: Data manipulation and analysis.

#### **Sentence-Transformers:**

Purpose: Generating embeddings for semantic search.

#### **Transformers:**

Purpose: Transformer models for natural language processing.

#### **ChromaDB:**

Purpose: Vector database for storing and searching embeddings.

### **Whisper (OpenAI Whisper):**

Purpose: Speech-to-text conversion.

### **ONNX Runtime:**

Purpose: Optimizing ML model performance.

### **Torch (PyTorch):**

Purpose: Core framework for deep learning models.

### **MoviePy:**

Purpose: Handling video processing tasks.

### **NumPy:**

Purpose: Numerical operations and data manipulation.

## **System Architecture: Connecting the Dots**

- 1. Data Extraction and Preprocessing:** Subtitle files are loaded and cleaned. Text is embedded for fast retrieval.
- 2. Embedding Generation:** Subtitle chunks are embedded using Sentence-Transformers.
- 3. Speech-to-Text Conversion:** Whisper transcribes audio to text.
- 4. Search and Retrieval:** Transcribed text is matched with stored embeddings in ChromaDB.
- 5. Results Display:** Top matching subtitles are displayed with similarity scores.

## **Challenges and Solutions**

- 1. Handling Large Subtitle Files:** Optimized memory usage with batch processing.
- 2. Real-Time Search Efficiency:** Used ChromaDB for fast vector indexing.
- 3. Accurate Speech Transcription:** Used Whisper for robust transcription and ONNX Runtime for CPU efficiency.

## **Future Enhancements**

- 1. GPU Acceleration:** To enhance embedding generation speed.
- 2. Multilingual Support:** Expanding Whisper's capabilities to support more languages.
- 3. Dynamic Collection Management:** Allowing users to upload and index their subtitle files.

## **Conclusion: A Smart Search Engine for Media Content**

The Shazam Clone for Subtitle Search marks a significant leap in media search technology, combining state-of-the-art NLP, robust audio transcription, and real-time search capabilities. By leveraging powerful libraries and efficient processing techniques, this project achieves fast and accurate subtitle matching - a true blend of speech recognition and semantic search.

Whether you're a movie enthusiast, a podcast lover, or a developer interested in cutting-edge media processing, this tool opens up exciting possibilities for content discovery and media indexing.