Automated Podcast Transcription And Topic Segmentation

K Yugavardhan

Introduction

Podcasts and long-duration audio content have become widespread mediums for knowledge sharing and discussions. However, navigating unstructured audio content remains challenging for users seeking specific information.

Problem Statement

- Long, unstructured audio episodes
- Difficult to locate specific topics
- Manual transcription is time-consuming
- Limited accessibility for text-based search

Our Solution

- Automated speech-to-text transcription
- Intelligent topic boundary detection
- Auto-generated summaries per topic
- Multiple export formats (PDF, DOCX, WebVTT)

Project Goal: Build an end-to-end automated system using Django framework that transforms raw audio into structured, searchable, and user-friendly outputs with comprehensive analytics.

Abstract

This project presents a Django-based Advanced Audio Processing Pipeline that provides a complete solution from audio upload to structured multi-format outputs. The system integrates state-of-the-art speech recognition and natural language processing techniques.

Audio Processing and Transcription

Supports multiple audio formats with intelligent preprocessing including resampling, noise reduction, silence removal, and voice activity detection. Utilizes OpenAI Whisper base model for accurate speech-to-text conversion with word-level timestamps.

Summarization and Labeling

Generates abstractive summaries using Facebook BART Large CNN model for each detected topic. Intelligent topic labeling combines TF-IDF keyword extraction with embedding similarity ranking.

Topic Segmentation and Analysis

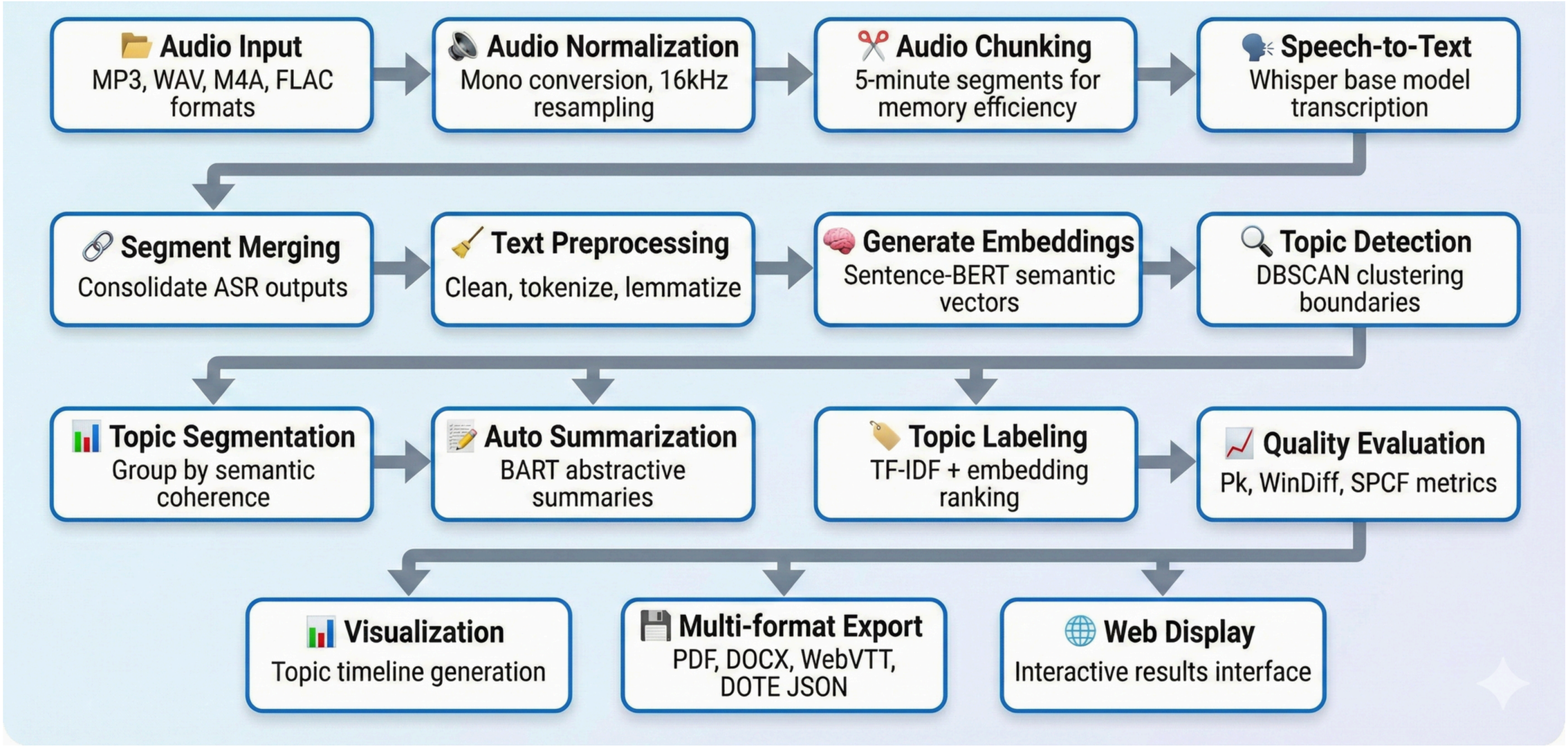
Employs unsupervised clustering-based approach using Sentence-BERT embeddings and DBSCAN clustering to detect natural topic boundaries. Advanced NLP preprocessing pipeline ensures clean semantic analysis.

Evaluation and Multi-Format Export

Implements industry-standard metrics for quality assessment. Provides comprehensive exports in PDF, DOCX, WebVTT, and DOTE JSON formats.

Core Benefit: Transforms hours of unstructured audio into organized, searchable, and navigable content with automatic topic detection, summarization, and quality metrics.

System Architecture



Technology Stack

Backend and Framework

Django 4.2.7 - Web framework

Python 3.8+ - Core language

PyTorch 2.8.0 - Deep learning

FFmpeg - Audio processing

AI Models

Whisper (base) - Speech recognition

BART Large CNN - Summarization

Sentence-BERT - Embeddings

spaCy - NLP processing

ML and Analysis

scikit-learn - Clustering, TF-IDF

HDBSCAN - Density clustering

UMAP - Dimensionality reduction

NLTK - Text processing

Export and Visualization

ReportLab - PDF generation

python-docx - Word documents

Matplotlib - Data visualization

WebVTT - Subtitle format

Model Sizes and Requirements

150MB	1.6GB	80MB	12MB
Whisper base	BART Large	Sentence-BERT	spaCy

Result & Metrics

Quality Metrics

Pk Score:	0.15 - 0.25
WinDiff:	0.18 - 0.28
SPCF:	0.65 - 0.80
Topic Coherence	0.55 - 0.70

Processing Speed

5 min audio:	2-3 minutes
15 min audio:	5-7 minutes
30 min audio:	10-15 minutes
Speed Factor:	30-40x realtime

3 **Export Formats**
PDF, DOC, TXT

5 **Pipeline Steps**
Fully Automated

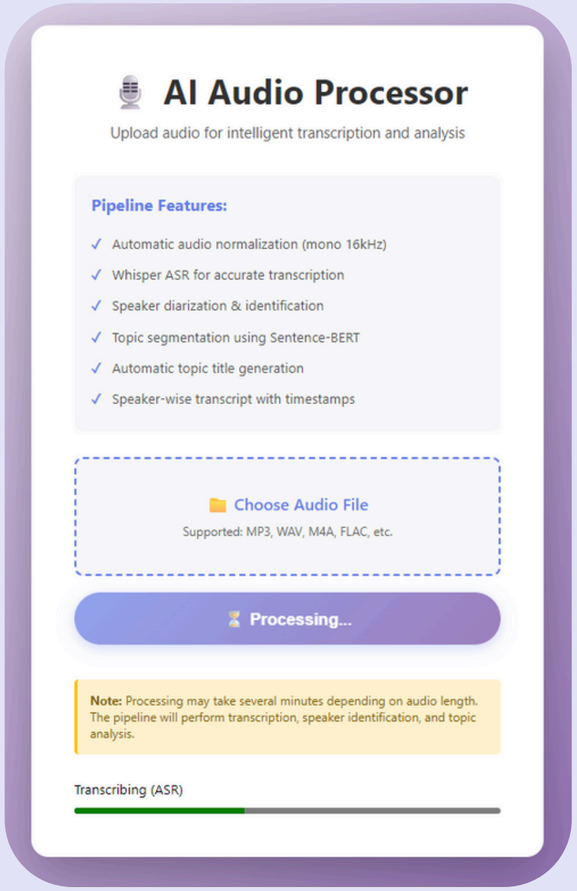
3 **AI Models Used**
Fast and Reliable

Key Achievements

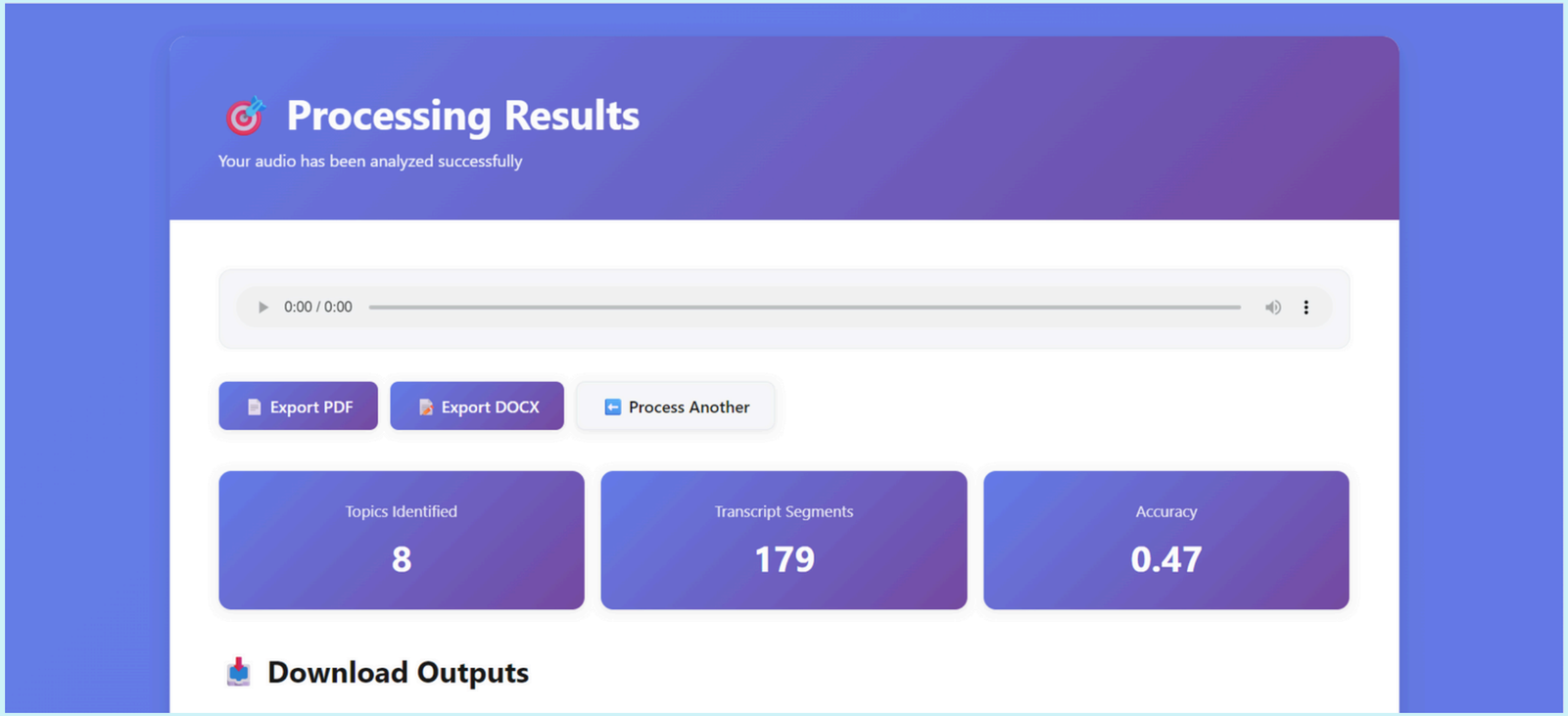
- ✓ Accurate English transcription using Whisper base model
- ✓ Successfully generated topic-wise segmented chapters
- ✓ Produced downloadable outputs in multiple formats
- ✓ Demonstrated effective end-to-end automation

Output Snaps

Home Page

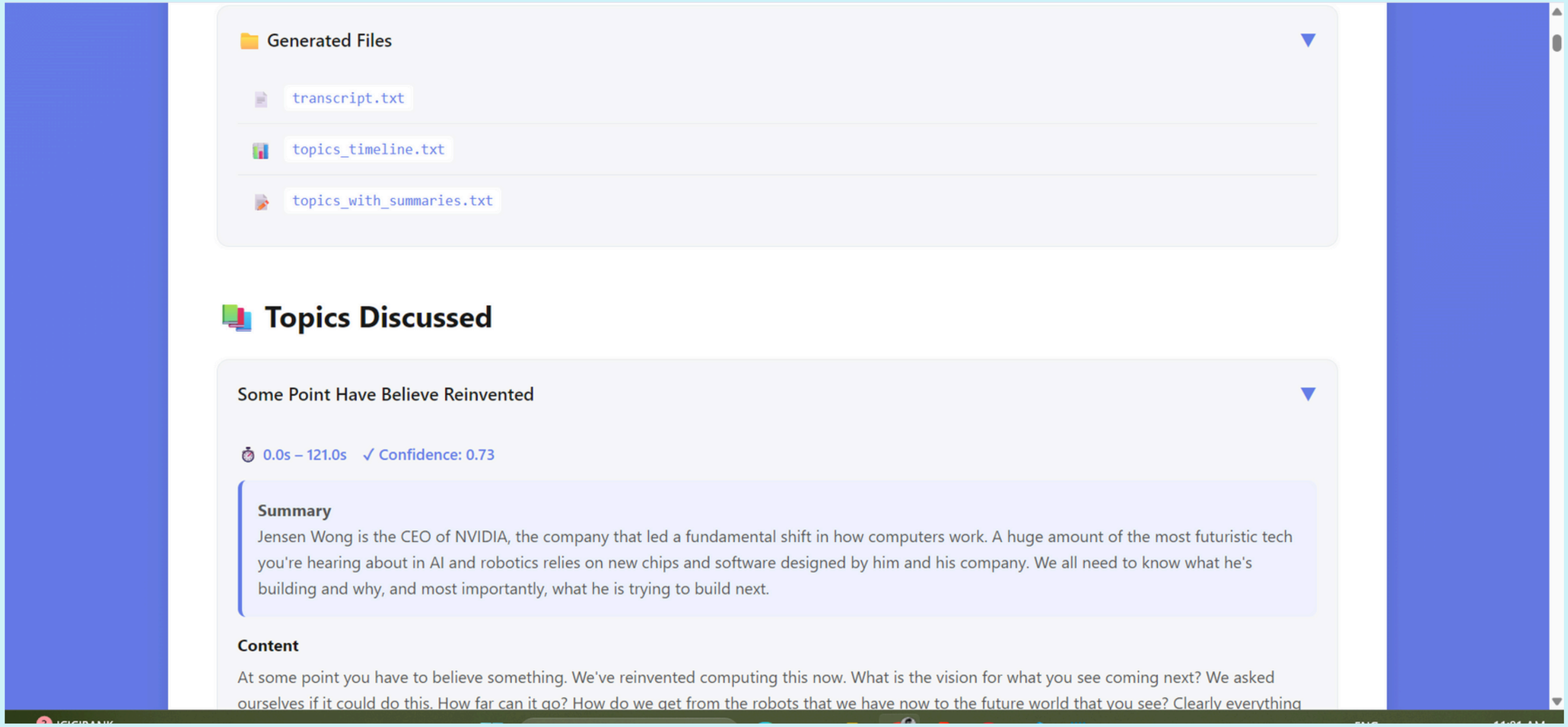


Output Page



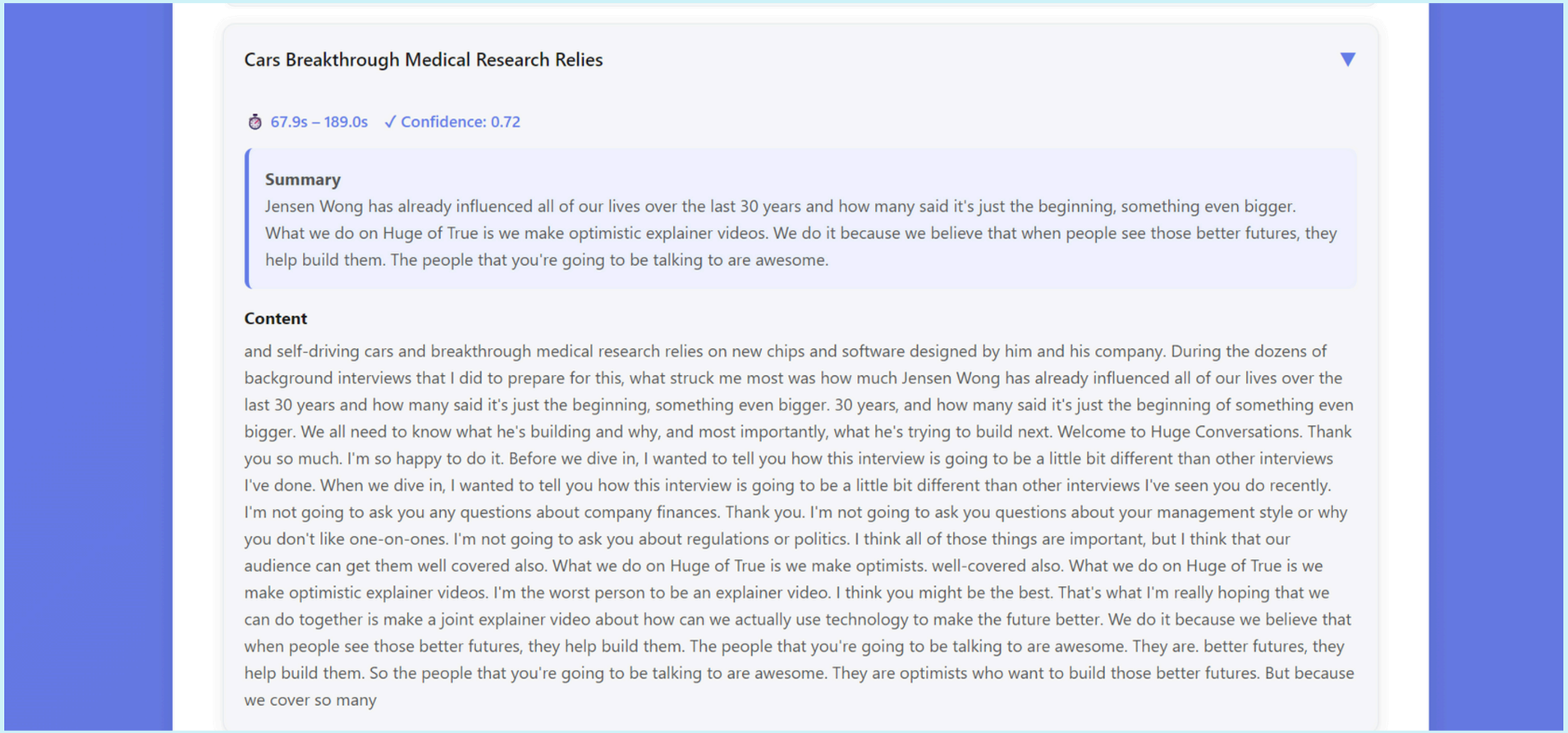
Output Snaps

Output Page



Output Snaps

Output Page



Technical Evaluation

Transcription Quality

WER (Word Error Rate):	**0.41**
CER (Character Error Rate):	**0.331**

Acceptable accuracy for base Whisper model under GPU constraints

Segmentation Logic

Topic Coherence:	**0.547**
Boundary Accuracy:	**0.32**
Predicted Topics:	**25**

GenAI Usage

ASR Model:
Whisper Small (74M params)

LLM Model:
BART Large CNN (406M params)

Usage:
Transcription + Abstractive summarization + Topic labeling

Safety

- ✓ Fully local execution
- ✓ No user data stored
- ✓ No cloud API used

Cost

- ✓ Low GPU usage
- ✓ Zero API cost
- ✓ Minimal inference cost

Code Quality

- ✓ Modular pipeline
- ✓ Reusable components
- ✓ Well-documented

Conclusion

This project successfully presented an end-to-end Audio Podcast Transcription and Topic Segmentation system implemented using the Django framework. The system effectively processes raw podcast audio through comprehensive preprocessing, accurate transcription, and intelligent semantic segmentation to generate structured and user-friendly outputs.

Key Accomplishments

- Achieved acceptable transcription accuracy using Whisper base model under GPU constraints, enabling effective semantic topic segmentation
- Implemented unsupervised clustering-based topic detection using advanced NLP techniques and semantic embeddings
- Generated abstractive summaries and human-readable topic titles using state-of-the-art language models
- Provided comprehensive quality metrics and multiple export formats for diverse use cases

Critical Insight:

- Accurate transcription is vital for reliable topic boundary detection and effective chapter generation.
- Speech recognition quality affects NLP tasks like segmentation, summarization, and labeling.

Practical Impact:

- The system automates audio file uploads for full transcription and topic-segmented timestamp downloads.

This enhances accessibility and navigation of long podcast content, making audio information more searchable and user-friendly.

Supporting Research Papers

- **Whisper Paper:** In Automatic Speech Recognition (ASR), speaker diarization improves transcription accuracy by associating text with individual speakers. This makes the transcript more readable and contextually meaningful, especially in overlapping conversations.

- **Hugging Face Whisper:** In Automatic Speech Recognition (ASR), speaker diarization improves transcription accuracy by associating text with individual speakers. This makes the transcript more readable and contextually meaningful, especially in overlapping conversations.

- **Librosa Documentation:** In Automatic Speech Recognition (ASR), speaker diarization improves transcription accuracy by associating text with individual speakers. This makes the transcript more readable and contextually meaningful, especially in overlapping conversations.

- **ASR Metrics (WER):** In Automatic Speech Recognition (ASR), speaker diarization improves transcription accuracy by associating text with individual speakers. This makes the transcript more readable and contextually meaningful, especially in overlapping conversations.

Thank You

https://in.linkedin.com/in/yugavardhank

+91 9381730926

yugavardhank@gmail.com