1. Title

Automated Medical Podcast Transcription and Topic Segmentation

---

2. Overview

In recent years, medical podcasts have become an important source of information for medical students, healthcare professionals, researchers, and educators. These podcasts cover a wide range of topics such as diseases, symptoms, diagnostic methods, treatment strategies, clinical experiences, and public health awareness. Due to their conversational nature, podcasts provide real-world insights that are often not available in traditional textbooks.

However, most medical podcasts are lengthy and unstructured. Identifying specific information from an audio-only format requires listening to the entire episode, which is time-consuming and inefficient. This makes it difficult for users to quickly access relevant medical information or reuse podcast content for academic and reference purposes.

To address this challenge, this project proposes an AI-based system that automatically converts medical podcast audio into text, segments the transcript into meaningful topic-based sections, extracts important keywords, performs sentiment analysis, and presents the results through an interactive interface. The system transforms raw medical audio into structured, searchable, and user-friendly content.

---

3. Approach

The system follows a pipeline-based approach to efficiently process medical podcast audio. Initially, the uploaded audio file is preprocessed to ensure compatibility with speech recognition models. This includes format conversion, resampling, and normalization.

The preprocessed audio is then converted into text using an Automatic Speech Recognition (ASR) model. The generated transcript is divided into sentences and analyzed using Natural Language Processing techniques. Topic segmentation is performed by grouping semantically related sentences into coherent sections.

For each topic segment, important medical keywords are extracted to summarize the discussion. Sentiment analysis is applied to understand the emotional tone of each segment. Transcription quality metrics are computed to evaluate the accuracy of the generated text. Finally, all outputs are displayed through an interactive web-based interface that allows users to explore transcripts, topics, keywords, and sentiment insights.

---

4. Tech Stack (Implemented Architecture)

The project is implemented using a layered technology stack to support efficient audio processing, speech recognition, natural language analysis, and interactive visualization.

4.1 Programming Language

- **Python 3.8+**
  Used as the core language for backend processing, machine learning integration, and application logic.

4.2 Audio Processing Layer

- **Librosa** – Audio loading, resampling, and preprocessing
- **SoundFile** – Reading and writing audio files
- **FFmpeg** – Audio format conversion (MP3 ↔ WAV)

This layer ensures that uploaded medical podcast audio files are converted into a standardized format suitable for speech recognition.

4.3 Speech-to-Text (ASR) Layer

- **OpenAI Whisper (small model)**

Whisper is used for Automatic Speech Recognition to convert medical podcast audio into text. It supports long-duration audio and handles complex medical terminology effectively.

4.4 Natural Language Processing Layer

- **Sentence Transformers** – Sentence embeddings for semantic similarity
- **NLTK** – Tokenization, stopword removal, sentence processing
- **KeyBERT** – Context-aware keyword extraction

This layer performs topic segmentation and identifies key medical terms from each topic segment.

4.5 Sentiment Analysis Layer

- **TextBlob**

Used to analyze the emotional tone of each topic segment and classify sentiment polarity.

4.6 Visualization Layer

- **Matplotlib** – Keyword frequency plots and sentiment graphs
- **WordCloud** – Visual representation of extracted medical keywords

4.7 Frontend / User Interface

- **Gradio**

Provides an interactive web-based interface for uploading audio, viewing transcripts, topic segments, keywords, and sentiment analysis results.

4.8 Execution Environment

- **Local Machine / Google Colab**
- **Python Script / Jupyter Notebook**

---

5. Setup

To set up the project environment, Python 3.8 or above is required. All required dependencies must be installed using the Python package manager. FFmpeg must also be installed to support audio format conversion.

After installing the required libraries, essential NLP resources such as tokenizers and stopword datasets are downloaded using NLTK. Once setup is complete, the application can be executed locally or in cloud-based environments such as Google Colab.

---

6. Usage

The user launches the application and uploads a medical podcast audio file in WAV or MP3 format. After uploading, the system automatically begins processing the audio.

The application generates a complete transcript, divides it into topic-based segments, extracts important keywords, and performs sentiment analysis. Results are displayed in different sections of the interface, enabling users to easily navigate between transcripts, topic segments, keyword analytics, and sentiment information.

---

7. Troubleshooting

Low transcription accuracy can often be improved by using high-quality audio with minimal background noise. Unsupported or corrupted audio files may cause processing failures; therefore, only standard WAV or MP3 formats should be used.

Slow model loading may occur during initial execution due to model downloads or limited internet connectivity. If visual outputs such as keyword clouds do not appear, it may indicate insufficient text data. Logging mechanisms help track errors and assist in debugging.

8. Limitations and Challenges

The system supports only recorded audio files and does not handle real-time transcription. Transcription accuracy depends on audio quality, speaker clarity, and pronunciation of medical terminology.

Topic segmentation may not always perfectly detect topic boundaries in highly conversational podcasts. Sentiment analysis may not fully capture the contextual sensitivity of medical discussions. The system is strictly intended for educational and informational purposes and does not provide medical diagnosis or treatment recommendations.

9. Future Scope

Future enhancements include the integration of Medical Named Entity Recognition to automatically identify diseases, drugs, and symptoms. Speaker diarization can be added to distinguish between multiple speakers.

The system can be extended to support multilingual medical podcasts and semantic search across multiple podcast episodes. Automatic topic-wise summarization and cloud-based deployment can further improve scalability, accessibility, and usability.

10. References

1. OpenAI Whisper Documentation
2. HuggingFace Transformers Documentation
3. Sentence Transformers Research Papers
4. KeyBERT Documentation
5. Librosa Audio Processing Library Documentation
6. Research studies on Automatic Speech Recognition and Natural Language Processing