



# CONVERTING PODCAST EPISODES INTO TEXT FORMAT AND SUMMARIZING THEM

<sup>1</sup>Dr. Sandeep Kumar, <sup>2</sup>Mr. Arun S Adiga, <sup>3</sup>Mr. R N Ravi, <sup>4</sup>Mr. M Hitesh, <sup>5</sup>Mr. Suhas B T

<sup>1</sup>Professor, <sup>2</sup>Student, <sup>3</sup>Student, <sup>4</sup>Student, <sup>5</sup>Student

<sup>1</sup>Computer Science and Engineering Department,

<sup>1</sup>Cambridge Institute of Technology, Bangalore, India

**Abstract:** The "Converting podcast episodes into text format and summarizing them" project aims to automate the process of summarizing podcasts, making it convenient for users to quickly grasp the key points of lengthy audio content. The process begins with data collection, where podcast episodes are gathered either as transcripts or audio files. For transcripts, preprocessing techniques are applied to clean the text data, removing unnecessary characters and tags. For audio files, speech recognition tools are employed to convert spoken words into text. The summarization techniques primarily include both extractive and abstractive methods.

## I. INTRODUCTION

In today's fast-paced digital age, the consumption of audio content, particularly podcasts, has become increasingly popular. Podcasts, with their diverse range of topics and voices, provide a wealth of information and entertainment. However, the inherent challenge lies in the audio format itself; the content is not readily accessible for those with hearing impairments, and for others, finding specific information within lengthy audio recordings can be time-consuming and daunting. Recognizing this gap, our project delves into the realm of converting podcasts to text and subsequently summarizing them.

## II. MOTIVATION

There is an overwhelming amount of audio content available, leading to information overload. In the face of information overload, users can struggle to stay informed, entertained, or educated without feeling overwhelmed. There is also the language barrier which refers to the challenges people face when trying to access or understand content that is presented in a language they are not proficient in. Language barriers can be a significant hindrance to accessing the rich and diverse world of podcasts. Time efficiency is a crucial aspect of modern life, especially in a fast-paced digital age where people are constantly bombarded with information and have limited time to consume it. With the rapid advancements in AI and machine learning, there is a tremendous potential to obtain summaries of various podcasts which lasts for long periods in a short and efficient manner. Moreover, it will also help the popularity of the various podcasts which are good but too lengthy for the average users to sit and listen to for extended periods of time.

## III. APPLICATIONS

**Time-Saving for Users:** Summarized podcasts save users time by providing the most important content in a concise format. Users can quickly grasp the essence of a podcast without investing the time required for full-length listening.

**Accessibility:** Summaries make podcasts more accessible to individuals with hearing impairments or those who may prefer reading over listening. This inclusivity can broaden the audience for podcasts.

**Content Creation:** Content creators can use summarized podcasts as references for their work, helping them stay updated on industry trends and gaining inspiration from thought leaders in various fields.

**Learning and Education:** Summaries can serve as supplementary educational resources, providing concise information on specific topics covered in podcasts. This can be particularly useful for students and lifelong learners.

**Media Monitoring:** Podcast summarization can be valuable for media monitoring agencies and PR professionals who need to stay informed about the latest developments in their industries. Summaries of relevant podcasts can provide quick insights into industry news and trends

#### IV. PROBLEM STATEMENT

Podcasts are primarily auditory content, which poses challenges for individuals with hearing impairments or limited access to audio content due to language barriers.

The vast number of podcasts and their diverse topics create information overload, making it challenging for users to identify relevant content efficiently.

Some podcasts discuss time-sensitive topics or current events, making it essential for users to access summarized information quickly and efficiently.

Educators and researchers face challenges in efficiently incorporating podcasts into educational curricula or research analyses due to the time-intensive nature of audio content.

#### V. OBJECTIVES

Develop a podcast summarization system that converts audio content into text-based summaries, making podcasts accessible to individuals with hearing impairments and non-native speakers, thus promoting inclusivity

Create an automated podcast summarization algorithm that distills key points, main ideas, and significant discussions, helping users manage information overload by providing concise and informative summaries of podcast episodes.

Develop a podcast summarization tool specifically designed for educational and research purposes, providing detailed yet concise summaries that aid in classroom instruction, student learning, and academic research.

Develop a real-time podcast summarization system that processes new episodes as they are released, providing timely and relevant summaries to users, ensuring they stay updated on the latest information and discussions.

#### VI. SYSTEM ARCHITECTURE

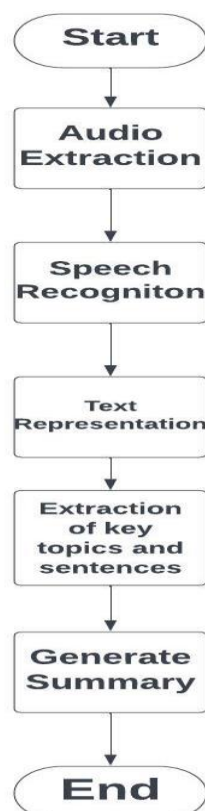


Figure 1: System Architecture

The architectural diagram is broken down into 5 main crucial steps which represent the flow of execution of the system and the relationship between the individual components.

**1. Start:** The "Start" step marks the initiation of the podcast summarization process, triggered when a user selects a specific podcast for analysis and summary generation. This step involves the user's decision to engage with a particular podcast, initiating the subsequent sequence of actions aimed at extracting, analyzing, and summarizing the podcast content. The user's choice sets the project into motion, guiding the system to retrieve the relevant audio data.

**2. Audio Extraction:** In the "Audio Extraction" step, the podcast summarization process advances by retrieving the audio content from the chosen podcast source. This can involve accessing a podcast server or obtaining the audio from a local storage location. The extraction phase is pivotal as it provides the raw material for subsequent analysis. By capturing the podcast's auditory elements, this step ensures that the spoken content becomes available for further processing, including speech recognition. Successful audio extraction is fundamental to the accurate representation of the podcast's verbal content and sets the stage for subsequent steps in the summarization workflow.

**3. Speech Recognition:** In the "Speech Recognition" step, the extracted audio content undergoes transformation from spoken words to textual data. This crucial phase employs advanced speech recognition technology to accurately transcribe the verbal elements of the podcast into a format that can be further analyzed. The primary goal is to convert the audio signals into a textual representation, laying the foundation for subsequent natural language processing (NLP) and summarization processes. This step is instrumental in enhancing accessibility to the podcast content, making it feasible for the system to analyze and distill meaningful information from the spoken words, ultimately facilitating the creation of a concise and informative podcast summary.

**4. Text Representation:** In the "Text Representation" step, the transcribed spoken words from the speech recognition phase are formatted and structured into a textual representation suitable for subsequent analysis. This phase involves organizing the recognized text in a way that preserves its coherence and meaning, making it amenable to natural language processing (NLP) techniques. The formatted text serves as the basis for further linguistic analysis and information extraction. Ensuring a well-organized and intelligible textual representation is pivotal for the subsequent stages of the podcast summarization process, allowing for effective application of NLP algorithms to identify key topics and sentences for summarization.

**5. Extraction of Key Topics and Sentences:** In the "Extraction of Key Topics and Sentences" step, the podcast's textual representation is subjected to natural language processing (NLP) techniques to identify and extract the most relevant and crucial information. This phase involves analyzing the structured text to discern key topics, themes, and significant sentences within the podcast content. NLP algorithms may consider factors such as frequency, importance, and context to prioritize the extraction of essential elements. By distilling the podcast's information into its most pertinent components, this step lays the groundwork for the subsequent application of summarization algorithms, ensuring that the resulting summary captures the essential essence of the original podcast in a concise form.

**6. Generate Summary:** In the "Generate Summary" step, the distilled and prioritized information from the previous stages is synthesized into a concise and coherent summary. This phase employs a summarization algorithm, which may use various techniques such as extraction of key sentences or abstraction to distill the essential elements of the podcast content. The algorithm aims to condense the wealth of information obtained from the podcast into a shorter, easily digestible form while retaining the core meaning. The generated summary serves as the final output of the podcast summarization process, providing users with an efficient way to grasp the main points and insights of the original podcast without the need for a comprehensive listening experience.

**7. End:** The "End" step signifies the culmination of the podcast summarization process, indicating the successful completion of the workflow. At this stage, the summarization algorithm has processed the podcast content, and a concise summary has been generated. The summarization system has fulfilled its objective of distilling key information from the podcast, allowing users to efficiently grasp the core content without the need to listen to the entire episode. This endpoint marks the conclusion of the computational processes and signifies that the summarized content is ready for presentation, distribution, or further user interaction. The summarization project concludes here, having transformed the initial podcast selection into a condensed and informative summary which is then displayed to the user via the front end implementation.

## VII. SEQUENCE DIAGRAM

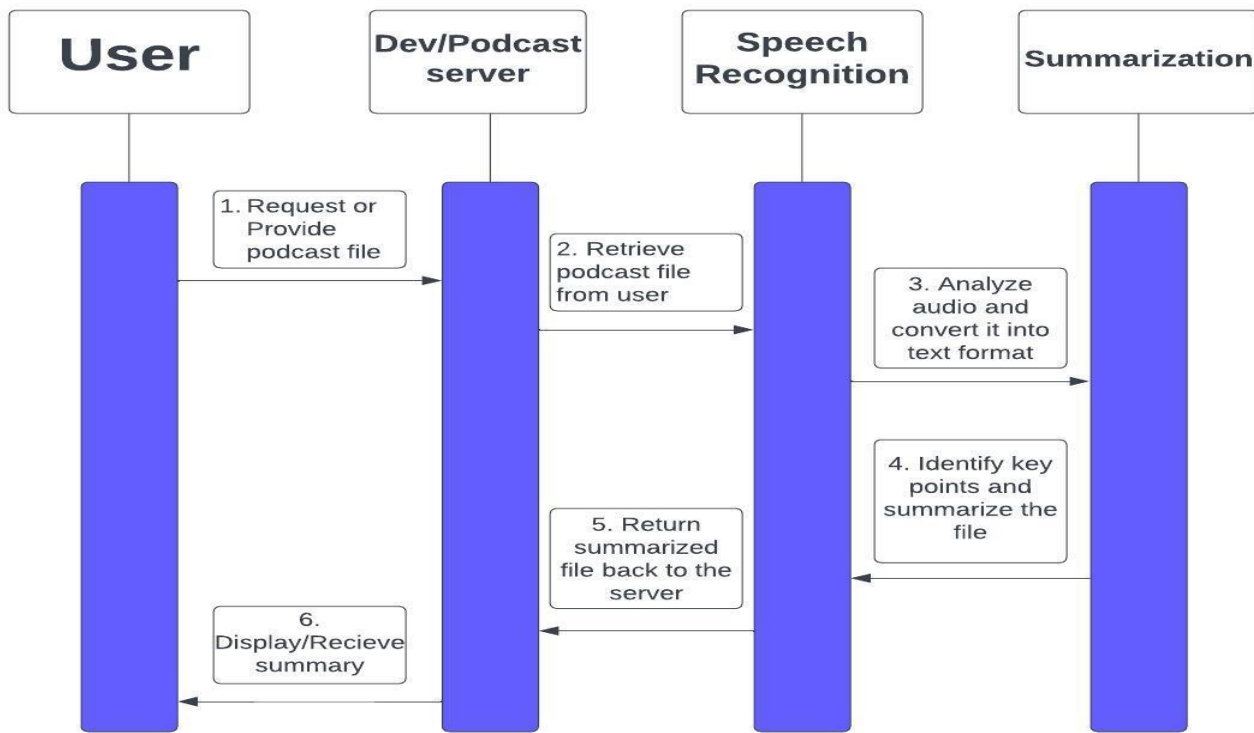


Figure 2: Sequence Diagram

**1. Request or Provide Podcast file:** The "Request Podcast" step in the podcast summarization project involves the user initiating the process by expressing the desire to listen to a particular podcast episode. This interaction is typically initiated through a user interface or application, where the user selects a specific podcast or provides relevant search criteria. The "Request Podcast" step marks the beginning of the podcast summarization workflow. Upon receiving the user's request, the system proceeds to retrieve the selected podcast episode, accessing the audio data associated with that episode. This can involve fetching the podcast feed, downloading the audio file, or streaming it in real-time, depending on the system's architecture. The user's choice may be based on various factors such as the podcast title, episode description, or recommendations. This step sets the stage for subsequent processes, including the playback of the podcast episode and the application of the summarization algorithm. It is a crucial initial interaction point where user preferences and system functionality converge, leading to the seamless integration of podcast content into the summarization pipeline. The system's ability to accurately interpret and respond to the user's podcast request significantly influences the overall user experience in the podcast summarization project.

**2. Retrieve podcast file from user:** The step "Retrieve podcast file from user" represents the process by which the podcast summarization system obtains the audio data associated with the user's requested podcast episode. This interaction is crucial for subsequent stages of the summarization process. Initiated by the user's podcast request, the system enters into the "Retrieve podcast file from user" step. In practical terms, this involves the podcast player component interfacing with external sources such as podcast hosting platforms or databases to fetch the audio file corresponding to the requested episode. The sequence diagram illustrates a flow of messages and interactions between the user, the podcast player, and possibly external data sources. Depending on the system's design, this retrieval process may encompass tasks such as downloading the audio file, streaming it in real-time, or accessing a pre-cached version. The podcast player component acts as the intermediary, managing the acquisition of the podcast content in preparation for subsequent stages, such as audio processing and summarization. Efficient execution of this step ensures timely access to the podcast content, contributing to a smooth and responsive user experience in the podcast summarization project.

**3. Analyze audio and convert it into text format:** The step "Analyze audio and convert it into text format" represents the pivotal process where the podcast summarization system transforms the audio data retrieved from the podcast episode into a textual format. This step is crucial for subsequent text-based analysis and summarization. Following the retrieval of the podcast file from the user, the audio processing component



takes center stage in the sequence diagram. This component is responsible for employing speech-to-text algorithms or services that analyze the audio content, transcribing it into a readable text format. The resulting text representation serves as the foundation for further summarization tasks. The sequence diagram outlines the flow of messages and interactions between the audio processing component and any relevant external services or APIs responsible for converting audio to text. This step is integral in bridging the gap between auditory information and machine-readable text, enabling subsequent natural language processing and summarization algorithms to operate on the textual representation of the podcast content. The effectiveness and accuracy of this conversion process significantly impact the quality of the summarization output, as the accuracy of the transcribed text directly influences the comprehensiveness and coherence of the generated podcast summary.

**4. Identify key points and summarize the file:** The step "Identify key points and summarize the file" signifies the core process where the podcast summarization system leverages the transcribed text to distill the essential information and generate a concise summary of the podcast episode. Following the successful conversion of audio to text, the sequence diagram illustrates the transition to the summarization component. This component employs natural language processing and machine learning algorithms to analyze the transcribed text, identifying key points, themes, and important information within the podcast episode. The diagram outlines the interactions and flow of messages involved in this crucial step. During this phase, the summarization algorithm extracts salient details, filters out redundancies, and captures the essence of the podcast content. The resulting summary encapsulates the most relevant aspects of the episode, providing users with a condensed yet informative overview. The sequence diagram highlights the collaboration between the components responsible for audio processing, text analysis, and summarization, emphasizing the interconnected nature of these processes. The successful execution of this step is pivotal in delivering a coherent and meaningful podcast summary, enriching the user experience by presenting distilled content aligned with their preferences and interests.

**5. Return summarized file back to the server:** In the sequence diagram, the step "Return summarized file back to the server" is the final phase in the podcast summarization process before being displayed to the user, where the generated summary is communicated back to the server for eventual user access. Following the identification of key points and the generation of the podcast summary, the sequence diagram illustrates the flow of information as the summarization component communicates with the server. This step involves the transmission of the condensed textual summary, typically in the form of a file or data structure, back to the server that initially handled the user's podcast request. The interaction signifies the completion of the summarization process, with the server now holding the summarized content. This summary can be stored for future reference, analysis, or caching, depending on the system's design. Additionally, the server may facilitate user access to the summary, enabling seamless retrieval and presentation through the user interface. The successful execution of this step ensures that the podcast summary is effectively integrated into the overall system architecture, making it accessible to the user upon request. The sequence diagram emphasizes the bidirectional nature of communication between the summarization component and the server, illustrating the cohesive workflow that transforms user-initiated podcast requests into concise and meaningful summaries.

**6. Display/Receive Summary:** The "Display/Receive Summary" step in the sequence diagram represents the final stage of the podcast summarization process, where the system communicates the generated summary to the user, completing the user experience loop. Following the transmission of the summarized file back to the server, the sequence diagram illustrates the user interface component interacting with the server to either receive or request the generated summary. This interaction involves the server delivering the condensed podcast summary to the user interface, which can be a web application, mobile app, or any other user-facing platform. The sequence diagram outlines the flow of information as the server transmits the summary to the user interface, emphasizing the bidirectional communication between these components. The user interface then takes responsibility for displaying the summary in a user-friendly format, making the distilled content easily accessible and comprehensible to the end user. This step is crucial for ensuring that the summarization efforts result in a meaningful outcome for the user, providing them with a concise and relevant overview of the podcast episode. The sequence diagram encapsulates the seamless integration of backend processes with the user interface, depicting the cohesive flow of information that transforms raw podcast data into a user-friendly and informative experience.

## VIII. CONCLUSION

Our project delivers a multifaceted solution tailored to enhance the accessibility and comprehension of podcast content for users. Through the provision of concise and coherent written summaries, listeners can swiftly grasp the essential information, key points, and topics discussed in each episode, eliminating the need to listen to the entire recording. By identifying and extracting key topics, themes, and subject matters, users gain a comprehensive overview of the podcast's main focus areas. Moreover, the recognition and highlighting of important entities mentioned, coupled with sentiment analysis, provide valuable context and insights into the emotional undertones of the conversation. Additionally, the generation of metadata, including keywords, tags, and categories, optimizes content discovery for users based on their interests. As a potential bonus, our system offers transcripts, promoting accessibility for individuals with hearing impairments and serving as an additional resource for those who prefer reading. Together, these features empower users with a holistic and enriching podcast experience, aligning with the evolving needs of diverse audiences in the digital landscape.

## REFERENCES

- [1] Kyon eth, 2023, podcasts Summary from Youtube or Spotify using the new Whisper and GPT3 chat APIs from OpenAI, <https://github.com/kyon-eth/podcast-summarizer>.
- [2] Bauer Griffin, 2022, podcast-summarizer using ytld sources, <https://github.com/FanaHOVA/podcast-summarizer>.
- [3] Kevin Shakes, 2022, podcast summarizer using Python libraries, <https://github.com/kevshakes/podcast-summarizer-frontend>
- [4] Mark Evans, 2022, podcast summarization using dataset, <https://github.com/TheOnesThatWereAbroad/PodcastSummarization>
- [5] Tony Leidolf, 2022, Podcast Summarization using BART Model, <https://github.com/tleidolf/BuildAI>
- [6] Burak Teke, 2022, Podcast Summarization using RSS fields, <https://github.com/tekeburak/podcast-summarizer>
- [7] Patrik Nanyn, 2023, Podcast Summarization using S3 Bucket, <https://github.com/patrick-nanys/podcast-summarization-app>
- [8] Anmol Gulati, 2023, Podcast Summarization using AssemblyAI, <https://github.com/AnmolGulati6/Podcast-Summarizer>
- [9] Gigi Kenneth, 2023, Podcast Summarization using OpenAI, <https://github.com/gigikenneth/podcast-summarizer>
- [10] MorenoLaQuatra, 2022, Obtaining Summaries of Large Texts using custom text summarizing Model, <https://github.com/MorenoLaQuatra/MATeR>