PODCAST TRANSCRIPTION AND SUMMARIZATION WITH SPEECH SYNTHESIS

Dr. P.V. Kashid[1], Ajit Chourashiya[2], Darshan Ugalmugale [3], Harshvardhan Dalvi [4],Pallavi Dumbare [5]

*1 Assistant Professor, Department of Information Technology, Sir Visvesvaraya Institute of Technology, Nashik, Maharashtra, India
*2,3,4,5 Department of Information Technology, Sir Visvesvaraya Institute of Technology, Nashik, Maharashtra, India

---

Abstract - We propose a system which automatically generate audio summaries for podcasts, allowing listeners to quickly preview podcast episodes. The proposed system first transcribes the audio from a podcast using automatic speech recognition (ASR), then summarizes the transcript using extractive text summarization, and finally returns the audio associated with the text summary. Motivated by a lack of relevant datasets for this task, we created our own by transcribing the audio from various podcasts and generating summaries for these transcripts using a manual annotation tool. After transcription, the text is processed through natural language processing (NLP) algorithms to extract the most important points, key topics, and relevant information from the podcast.

The first component of the system involves automatic transcription, where speech-to-text algorithms convert audio podcast episodes into textual format. This transcription process employs state-of-the-art machine learning models trained on large datasets to achieve high accuracy and reliability.

The system employs natural language processing (NLP) techniques to generate concise summaries of podcast episodes. These summaries aim to capture the essence of the content, providing users with a quick overview of key topics, insights, and discussions covered in the episode.

***Key Words*:** Podcasts, audio summarization, automatic speech recognition, automatic summarization.

1.INTRODUCTION

Podcasts have soared in popularity, offering a treasure trove of knowledge, entertainment, and inspiration. Yet, accessing this wealth of information can sometimes be daunting, especially for those with time constraints or hearing impairments. That's where our project comes in – bridging the gap between spoken word and accessibility through innovative transcription, summarization, and speech synthesis techniques.[1] To find appropriate information, a user needs to search through the entire documents this causes information overload problem which leads to wastage of time and efforts. To deal with this dilemma, automatic text summarization plays a vital role. Simultaneously, summarization condenses the key points and main ideas of a podcast episode into a concise format, providing a quick overview for busy individuals or those seeking a rapid grasp of the episode's highlights. The integration of speech synthesis technology further augments this process by utilizing artificial intelligence to convert the written text back into natural-sounding speech, replicating the original podcast experience for users who prefer auditory consumption. This amalgamation of transcription, summarization, and speech synthesis not only broadens the accessibility of podcasts but also

facilitates content repurposing and amplifies search engine optimization efforts. It empowers content creators and listeners alike by fostering a more comprehensive and inclusive podcast experience, catering to diverse preferences and accessibility needs.[2]. Through meticulous transcription, we unlock the spoken gems within each podcast episode, converting dialogue into text with remarkable accuracy. But we don't stop there. Our sophisticated summarization algorithms distill hours of audio content into concise, informative summaries, providing listeners with a quick overview of key points and highlights.

But perhaps the most transformative aspect of our project lies in speech synthesis – the ability to recreate human-like speech from text. With state-of-the-art synthesis technology, we breathe life into the written word, transforming transcripts and summaries into engaging audio experiences. Whether it's generating lifelike narrations or enabling seamless accessibility for the visually impaired, speech synthesis adds a new dimension to podcast consumption.

Join us on this journey as we explore the endless possibilities of podcast transcription and summarization with speech synthesis. From enhancing accessibility to revolutionizing content consumption, our project is at the forefront of innovation, shaping the future of audio communication one transcript at a time.

2. PRIOR WORK

Podcast summarization has only recently gained interest as a research problem. As far as we know, our work is the first to tackle the problem of audio summarization for podcasts. Most prior methods for podcast summarization focus on generating text summaries using abstractive summarization, where the original content has been rephrased. A

previous work also made use of ASR and text summarization but to generate short one-sentence summaries in the context of document retrieval for podcast search, and not for podcast audio summarization. [3]

A dataset has also been recently introduced for podcast search and abstractive text summarization. While text summaries can inform listeners about the topics in a podcast episode, they cannot convey subjective attributes that are typically found in the audio itself. Audio summaries would not only allow that, but they would also let users more conveniently listen to a summary rather than read one, for example, while cooking at home or driving. the basic assumption is that the vocal tract transfer function can be satisfactorily modeled by simulating formant frequencies and formant amplitudes. The synthesis thus consists of the artificial reconstruction of the formant characteristics to be produced. This is done by exciting a set of resonators by a voicing source or noise generator to achieve the desired speech spectrum, and by controlling the excitation source to simulate either voicing or voicelessness.[4]

The addition of a set of anti-resonators furthermore allows the simulation of nasal tract effects, fricatives and plosives. The specification of about 20 or more such parameters can lead to a satisfactory restitution of the speech signal. The advantage of this technique is that its parameters are highly correlated with the production and propagation of sound in the oral tract. The main current drawback of this approach is that automatic techniques of specifying formant parameters are still largely unsatisfactory, and that consequently, the majority of parameters must still be manually optimized[5].

3. METHODOLOGY

It involves several key steps to ensure accuracy, efficiency, and quality.[6]

Define Objectives and Scope:

- Clearly define the goals of the project: transcribing podcasts, summarizing content, and generating synthesized speech.
- Specify the scope of podcasts to be transcribed (e.g., specific topics, languages, durations).

Data Collection:

- Gather a diverse podcasts covering various topics and speakers.
- Ensure legality of data collection and usage, respecting copyright laws.

Preprocessing:

- Clean audio files to remove background noise, distortions, and irrelevant segments.
- Convert audio files to suitable formats for transcription and analysis.

Automatic Speech Recognition (ASR):

- Utilize ASR software to convert audio files into text.
- Fine-tune ASR models for improved accuracy, especially for domain-specific terminology or accents.
- Implement quality checks to identify and correct transcription errors.

Summarization:

- Develop summarization algorithms to condense transcribed text into concise summaries.
- Consider extractive or abstractive summarization techniques based on project requirements.
- Ensure summaries capture the essence of the podcast while maintaining coherence and relevance.

Natural Language Processing (NLP):

- Apply NLP techniques for enhancing the quality of summaries, such as entity recognition, sentiment analysis, and keyphrase extraction.
- Implement algorithms to identify important topics and themes within podcasts.

Speech Synthesis:

- Use text-to-speech (TTS) technology to generate synthetic voices from transcribed text.
- Choose appropriate TTS models that offer natural-sounding speech and support multiple languages.
- Fine-tune synthesized voices for clarity, tone, and pacing.

Evaluation:

- Establish evaluation metrics to assess the accuracy of transcriptions, quality of summaries, and naturalness of synthesized speech.
- Conduct human evaluations to validate the effectiveness of the system.
- Iterate on the methodology based on feedback and evaluation results.

Integration:

- Integrate transcription, summarization, and speech synthesis components into a cohesive system.
- Develop user interfaces or APIs for easy access and utilization of the system.

Deployment and Maintenance:

- Deploy the system on appropriate platforms, such as web applications or cloud services.
- Monitor system performance and address any issues or improvements through regular maintenance.
- Stay updated with advancements in ASR, NLP, and TTS technologies for continuous enhancement of the system.

Ethical Considerations:

- Ensure privacy and data security measures are in place, especially when dealing with sensitive content.
- Respect copyright laws and obtain necessary permissions for podcast usage and distribution.
- Provide transparent information to users regarding data usage and privacy policies.

User Feedback and Iteration:

- Encourage user feedback to improve the system's usability and effectiveness.

  Incorporate user suggestions and iterate on the methodology to address shortcomings and enhance user experience.

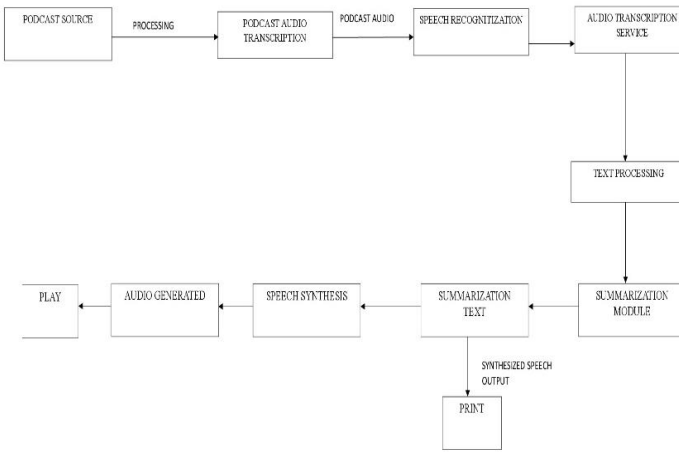


Fig. 1:- System Architecture

4.PROCESS OF PROJECT

Requirements Gathering:

- Define the scope of the project: What features do you want to include? What platforms will it run on?
- Understand the target audience and their needs.
- Decide on the technologies and tools you'll be using.

Transcription:

- Choose a suitable speech-to-text (STT) API or library. Some popular choices include Google Cloud Speech-to-Text, IBM Watson Speech to Text, or Mozilla DeepSpeech.
- Write code to feed the audio files (podcasts) to the STT system and obtain the transcribed text.

Summarization:

- Decide on the approach for summarization. You could use extractive summarization techniques, which involve selecting important sentences or phrases from the transcript.
- Implement an algorithm for extractive summarization. You can use libraries like Gensim or NLTK for this purpose.
- Experiment with different summarization techniques to find the one that works best for your use case.

Speech Synthesis:

- Choose a text-to-speech (TTS) API or library. Popular choices include Google Text-to-Speech, Amazon Polly, or Mozilla TTS.
- Write code to convert the summarized text into speech using the chosen TTS system.

- Experiment with different voices and parameters to make the synthesized speech sound natural and engaging.[7]

Integration:

- Integrate the transcription, summarization, and speech synthesis components into a cohesive system.
- Write code to handle the flow of data between these components and ensure smooth operation.
- Implement error handling and logging to facilitate debugging and maintenance.

User Interface:

- Design a user-friendly interface for interacting with the system. This could be a web application, a mobile app, or a command-line interface, depending on your target platform.
- Implement features such as uploading audio files, displaying transcriptions and summaries, and playing synthesized speech.

Testing and Refinement:

- Test the system thoroughly to identify and fix any bugs or issues.
- Gather feedback from users and stakeholders and make necessary refinements to improve the system's performance and usability.

Deployment:

- Deploy the system to a suitable environment, such as a web server or a cloud platform.
- Ensure that the deployment environment is properly configured and meets the system's requirements.
- Monitor the system's performance in production and make adjustments as needed.

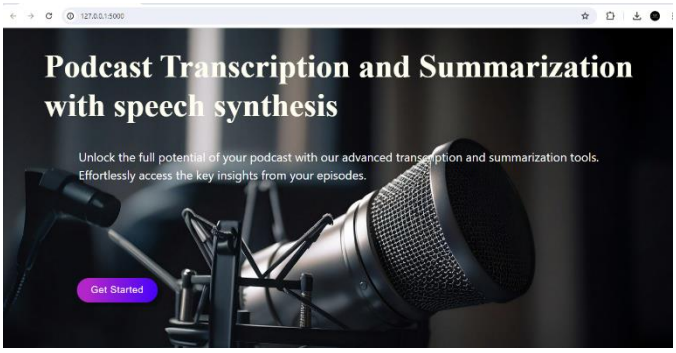
Maintenance and Updates:

- Regularly update the system to incorporate new features, fix bugs, and address security vulnerabilities.
- Monitor user feedback and usage patterns to guide future development efforts.
- Keep abreast of advances in speech recognition, summarization, and synthesis technologies, and incorporate relevant improvements into the system.

5. IMPLEMENTATION

1. **First Screen:**

After hosting on local its show this screen.



2. Get Started:-

After that click on get started it will redirect another page which is main Page or transcripted Page.

3. Main Screen

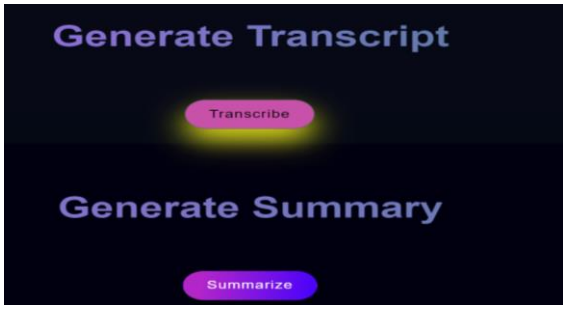
After Clicking on get started button its redirect to this screen which is main screen/ transcript page.

4. Show Video:-

After the main screen paste the podcast link in linkbox and click on show video it will show the video thumbnail/ video frame.

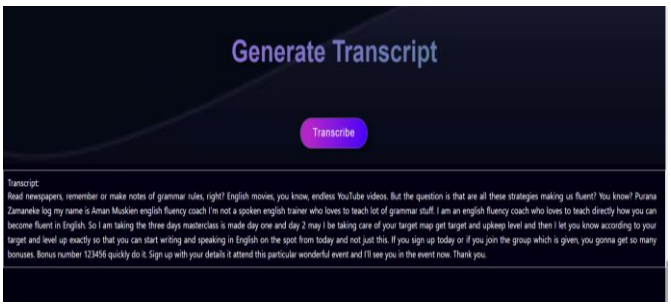
4. Generate Transcript:-

After show video click on transcribe button it will buffer.



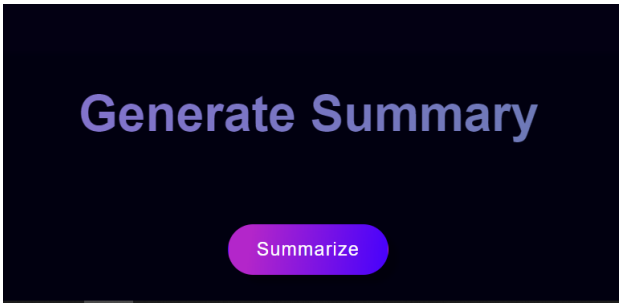
6. Generated Transcript Text

After show video click on transcribe button it will buffer. and show the transcribe text.



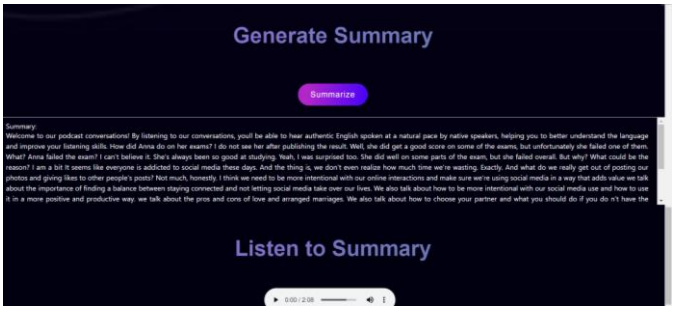
7. Generate Summarize:-

After show video click on summarizr button.



8. Generated Summarized Text:-

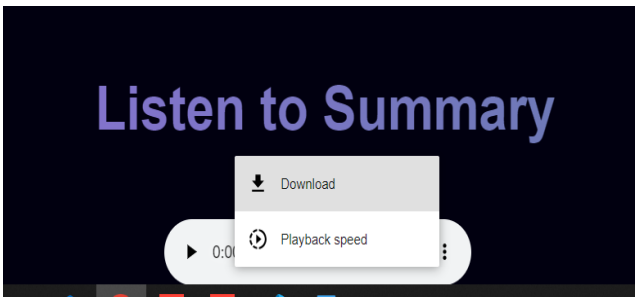
After show video click on transcribe button it will buffer. and show the summarized text.



9. Download Audio:-

After show video click on Summarize text shown it will activate audio option. click on play audio button it will play audio.

After playing audio click on download option it will download the audio in local folder user can listen later and set playback speed as per need.



CONCLUSIONS

Through meticulous transcription, we've transformed spoken words into accessible text, enabling broader audiences to engage with podcast content. Summarization techniques have efficiently distilled lengthy episodes into concise yet informative snippets, facilitating quick comprehension and discovery. Furthermore, the integration of speech synthesis technology has provided a novel avenue for delivering podcast content, offering an immersive and personalized listening experience.

This project has not only showcased the capabilities of modern natural language processing and machine learning techniques but also demonstrated their practical applications in enhancing accessibility and user experience in the realm of audio content consumption. Moving forward, the insights gained from this endeavor can serve as a foundation for further research and development, contributing to the ongoing evolution of audio content processing technologies.

Ultimately, the success of this project lies in its ability to bridge the gap between spoken content and its consumption, empowering individuals with diverse needs and preferences to engage with podcast content in a more meaningful and efficient manner.

ACKNOWLEDGEMENT

The implementation and documentation of this project would not be succeeded without the kind support from individuals. First of all, we would like to express our special gratitude to **Dr. P.V. Kashid** who always gives us valuable advice and kind assistance to complete this project. Last but not least, we would like to thank the Faculty of Information Technology, Savitribai Phule Pune University for giving us the great knowledge. Finally, we would like to give our appreciation to our parents who support us since the beginning till the end of this project.

REFERENCES

[1] A. Vartakavi, A. Garg and Z. Rafii, "AudioSummarization for Podcasts," 2021 29th European Signal Processing Conference (EUSIPCO), 2021.

[2] U. Hahn and I. Mani, “of Automatic Researchers are investigating summarization tools and methods that,” in IEEE Computer 33.11, no. November, pp. 29–36, IEEE, 2000.

[3] D.H. Klatt “Review of text-to-speech conversion for English,” Journal of the Acoustical Society of America, vol. 82(3), 1987.

[4] A. Khan and N. Salim, “A review on abstractive summarization methods,” Journal of Theoretical and Applied Information Technology, vol. 59, no. 1, pp. 64–72, 2014.

[5] R. Ferreira, L. De Souza Cabral, R. D. Lins, G. Pereira E Silva, F. Freitas, G. D. C. Cavalcanti, R. Lima, S. J. Simske, and L. Favaro, “Assessing sentence scoring techniques for extractive text summarization,” Expert Systems with Applications, vol. 40, no. 14, pp. 5755–5764, 2013.

[6] R. A. Garc´ıa-Hernandez and Y. Ledeneva, “Word sequence models for ´ single text summarization,” in Proceedings of the 2nd International Conferences on Advances in Computer-Human Interactions, ACHI 2009, pp. 44–48, IEEE, 2009.

[7] K. Sarkar, "An Approach to Summarizing Bengali News Documents," in International Conference on Advances in Computing, Communications and Informatics, pp. 857–862, ACM, 2012.

[8] Ms. Pratibha V. Waje, Dr. R. Jain, A Recommendation System for Execution Plans, Journal of Shanghai Jiaotong University, Volume 16, Issue 11, November - 2020,107-113.