A REVIEW ON TECHNIQUES FOR SPEECH TO TEXT CONVERSION AND TEXT SUMMARIZATION

Hemanshu Waghmare, Dhruv Dalvi, Rishabh Jain, Sanket Asole, Yuvraj Chavan

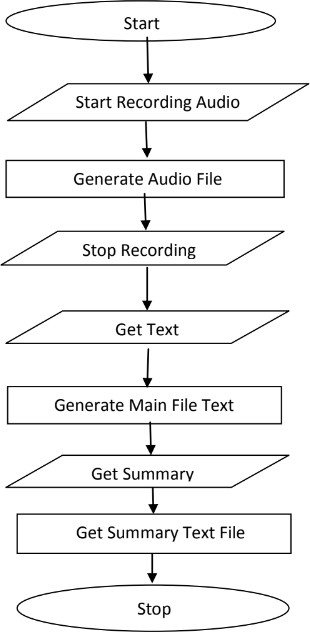
*Department of Computer Science and Engineering, YCCE, Nagpur, India*

Abstract: Speech-to-text and text summarization are two key areas, which are part of automatic natural language processing, which have been very useful in recent times as there has been an upsurge in the amount of digitized information as well as the need for efficient transcription and distilling information into its essence. This review paper evaluates the conventional approaches and structural patterns registrable in the recent innovations across both opuses with a keen interest on their complexities, centricity, and space of application. The first issue addressed is related to the transcription of speech into text. These include deep learning and end to end models like transformer networks for neural network-based speech transcription as well as cloud-based services. Several strengths and weaknesses of these methods are assessed relative to precision, time, and flexibility towards different languages and trends within same languages. Then the paper moves on to examine the text summarization techniques. Two types of text summarization techniques are compared, namely extractive and abstractive text summarization, emphasizing advances by transformer models such as BERT and GPT in generating context sensitive summaries. The last focus of the paper is on the speech recognition and summation technologies in practice as used by virtual assistants, meeting transcription applications, and content generation tools.

***Keywords:*** Machine learning, Speech to Text, Speech recognition, communication, Machine translation, Automatic Speech Recognition (ASR), Transformer.

# Introduction

The introduction of artificial intelligence (AI) and natural language processing (NLP) tools has revolutionized the way content is consumed. Speech-to-text transformation and text summarization stand out as linguistics technologies that are pivotal to effective interaction and access to information in the present global village. With the emergence of various voice- based, automatic transcription and text saturation over usage, the need for efficient speech-to-text translation and text summarization has grown greatly. The speech recognition technology such as ASR is of great importance since it encodes spoken words and phonemes into textual representation, hence supporting the ability to search and utilize voice data on various platforms. Before, systems ASR were based on statistical approaches of the analysis of the speech signal, such as Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM) – which they used together with the language and acoustic models to derive a reasonable accuracy. Nevertheless, with the development of deep learning and end-to-end neural models, ASR has evolved and now these techniques offer higher accuracy rates and an ability to recognize varying accents, languages, and contexts more effectively. Text summarization uses the information available in long texts to produce an overview that is shorter in terms of content and wording. It can also either be extractive which selects certain key sentences in the original text or be abstractive where a new summary is created that states the main point of the original text. Furthermore, in recent years transformer-based models such as BERT and GPT have successfully performed both extractive and abstractive types of text summarization achieving new heights in coherence and target relevance. This review paper aims to provide a comprehensive overview of current speech-to-text and text summarization methods, examining their underlying technologies, strengths, limitations, and real-world applications. By exploring these interrelated fields, we highlight the latest innovations and identify future directions to improve the integration of these technologies in diverse industries.



# Methods:

The methods for speech to text conversion and text summarization can be divided into categories such as cloud based, Neural Network based and Transformer based. Using the combination of these approaches multiple architectures are created. Some of these architectures are PodSumm architecture [11] and BertSumm (transformer) architecture

* 1. Cloud based

The speech is transformed into text by cloud hosted advanced speech-to-text technologies by using sophisticated machine learning models. Alike Google Cloud Speech-to-Text, Amazon Transcribe, and Microsoft Azure Speech Service offer integration and customization for transcription services, automated punctuation, identification of call participants, and support for multiple languages, both real-time and in bulk. They provide accurate services due to the presence of multi deep learning models on these platforms. Additionally, cloud-based services eliminate the need for local infrastructure, offering accessibility from anywhere, easy integration into applications, and the ability to handle large volumes of audio data efficiently.

Though the cloud-based methods are effective over the time for the production use the cost of the services may add up and cost heavily with increase in the number of the consumers. Cloud based methods are best used for a small-scale project. These methods are easy to setup and implement due to them being heavily abstracted.

* 1. Neural Network

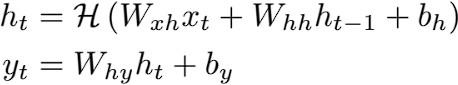
Long Short-Term Memory (LSTMs)

The paper [2] dives into the design and usefulness of LSTM systems, emphasizing their capacity to capture successive conditions in content information Through an examination of context- based text generation techniques, the author showcases the efficacy of LSTM networks in generating coherent and contextually relevant textual content. text generation models, highlighting their significance in tasks such as machine translation, dialogue systems, and summarization. Furthermore, the author presents experimental results and performance evaluations to demonstrate the effectiveness of LSTM networks in context-based text generation, paving the way for advancements in natural language processing and artificial intelligence research." Through a detailed examination of context-based text generation techniques, the author demonstrates the effectiveness of LSTM networks in generating coherent and contextually relevant textual output. The document outlines methods of incorporating contextual information into LSTM-based models, illustrating their use in machine translation, dialogue generation, summarization, and other fields.

Recurrent Neural Networks (RNN)

The paper [1] and [3] present us with structure of RNN network and its various implementations such as RNN Transducer. RNNs are ideal in tasks such as speech to text and text summarization RNNs have a unique perspective on speech because they continuously strive to convert audio features into text sequences. This enables them to transform audio features into textual forms. Such an ability is advantageous as it allows a network to understand context in the spoken language. RNNs are therefore able to compensate for varying patterns of speech and pronunciation accents that may exist across different individuals. These transcribers are more effective for RNNs’ greater transcription success increases their effectiveness in a broad range of applications including voice- command mechanics and automated transcription systems.

Deeps RNNs also consists of various layers such as Connectionist Temporal Classification, RNN transducer which are decoded by a decoding function and its output is regularised using Regularization process. The paper experiment on various combinations of the layer to find the most optimal and low error producing layer.



*Figure 1 Hidden layer function in RNN*

Extractive Summarization

The authors of paper [4] utilizes Extractive summarization [8] to generate a text summary of the input text. The words in the text are usually split and made ready for processing as the first step. The next step is feature engineering, where relevant qualities are chosen, like creating a feature representing a word using word embeddings which encode words semantically in numerical forms. After this, a RNN, an LSTM network, or any other neural network model is implemented and trained on labeled data to understand how sentences correspondence to their context and how important they are. To sum up, the model processes every sentence independently and assigns each a score indicating how important it is in the context of the paragraph. After that, the scores are used to filter the sentences, and the highest scoring sentences are selected and put together into a summary that is not only clear but also comprises significant information from the original text. Finally, the selected sentences are organized and formatted to produce the final summary.

* 1. Transformer based

Generative Adversarial Networks (GANs)

The overview [2] comprehensively explores the application of GANs in creating literary substance, highlighting their potential over different spaces. The GANs are built over the top of transformers [6]. By analyzing the basic standards and designs of GANs in content era, the overview illustrates the points of interest and challenges related with this approach. Assessment strategies particular to GAN-based content era, along with relevant evaluation measurements, are examined to evaluate the quality and coherence of created content. Moreover, the study addresses eminent progressions, rising patterns, and future investigate bearings in leveraging GANs for text generation, underscoring their importance in progressing the field of common language preparing. The overview fastidiously analyzes the utilization of GANs over a range of text generation assignments, counting language modeling, dialogue generation, and story generation. By dissecting the engineering and preparing methods of GANs in text generation. The study explains the challenges and openings characteristic in this approach.

Two-Stage Transformer

In the paper [5], the authors proposed and created a two-stage approach using the transformers [6] for variable-length summary has been proposed. Initially, the text segmentation module makes use of a pre-trained BERT and a bidirectional LSTM network to segment the input text. Next, BERTSUM or the BERT based extractive summarization model is constructed in order to perform the essential task of extracting the most salient sentence from each segment. With respect to training the two-stage summation model, the first step would involve employing the extracted sentences to train or language build the document summarization model depicted in the second stage. Thereafter, the extracted segments are used to train the specific summarisation module depicted in the first stage with the consideration of the outputs of this module and the documents summarisation module from the second stage. The parameters defining the segment summarization module are perturbed based upon the cumulative loss scores of the document summarization and segment summarisation modules. After this stage of training, an alternate training method termed as collaborative training is performed in which the parameters defining the segment summarisation and documents summarisation modules are trained repeatedly until the parameters converge. To test the framework the outputs from the segment summarization module are integrated, to produce variable-length, one-shot, abstractive summary. For evaluation, the BERT-biLSTM- based text segmentation model is evaluated using ChWiki\_181k database and obtains a good effect in capturing the relationship between sentences. Finally, the proposed variable-length abstractive summarization system achieved a maximum of 70.0% accuracy in human subjective evaluation on the LCSTS dataset.

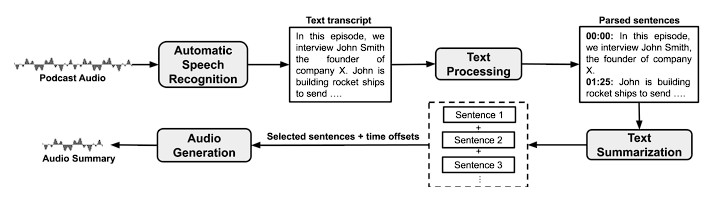
Dual-encoding using transformer

Recurrent neural network-based sequence-to- sequence attentional models have proven effective in abstractive text summarization [8]. In the paper [7], the author has proposed a model for abstractive text summarization using a dual encoding. The proposed method employs a dual encoder including the primary and the secondary encoders. Specifically, the primary encoder conducts coarse encoding in a regular way, while the secondary encoder models the importance of words and generates more fine encoding based on the input raw text and the previously generated output text summarization. The two-level encodings are combined and fed into the decoder to generate more diverse summary that can decrease repetition phenomenon for long sequence generation. The experimental results on two challenging datasets (i.e., CNN/DailyMail and DUC 2004) demonstrate that our dual encoding model performs against existing methods.

* 1. PodSumm Architecture

PodSumm first generates a transcript of the podcast audio using an ASR module and parses the

text transcript into individual sentences [10]. It then uses a text summarization model to select relevant sentences, along with their time offsets in the audio, and generates the final audio summary associated with the text summary. Each stage is discussed in detail below. 1) Automatic Speech Recognition: ASR methods perform the task of automatic speech-to-text transcription. As the purpose of this work is not to develop a new ASR system or improve on an existing one, we choose to use a well- known and publicly-available solution



*Figure 2 PodSumm Architecture*

for this task, namely AWS Transcribe. 2) Text Processing: The transcripts obtained from the ASR module contain the text for the individual words and punctuation marks, their start and end times in the audio, and their confidence scores regarding the prediction. The author chooses to use an open- source library for NLP, namely spaCy3, to parse the text into individual sentences with their corresponding start and end times. Additionally, they force a sentence break when a pause of over two seconds between words occurs. 3) Text Summarization: We generate text summaries by selecting relevant sentences from the transcripts, using automatic extractive summarization. They used the recently proposed Pre- Summ4 model [10], which builds upon BERT [10] to obtain a sentence level encoding, and stacks inter- sentence Transformer layers to capture document- level features for summarization. The final trained model showed the best performance (ROUGE-L F- score of 0.64) in comparison with the baseline method (ROUGE-L F-score of 0.52).

**Proposed Method**

**2.1 Audio Preprocessing**

**2.1.1 Noise Reduction**  
Using pydub and noise reduce the presence of background noise in audio recordings can severely degrade the performance of STT systems. We first preprocess the audio to reduce unwanted noise. Tools like pydub help in loading and handling audio files, while noisereduce employs spectral gating methods to effectively reduce background noise.

This noise reduction step enhances the clarity of speech by reducing the effects of background noises and disturbasnces like wind, traffic, or machine noise.

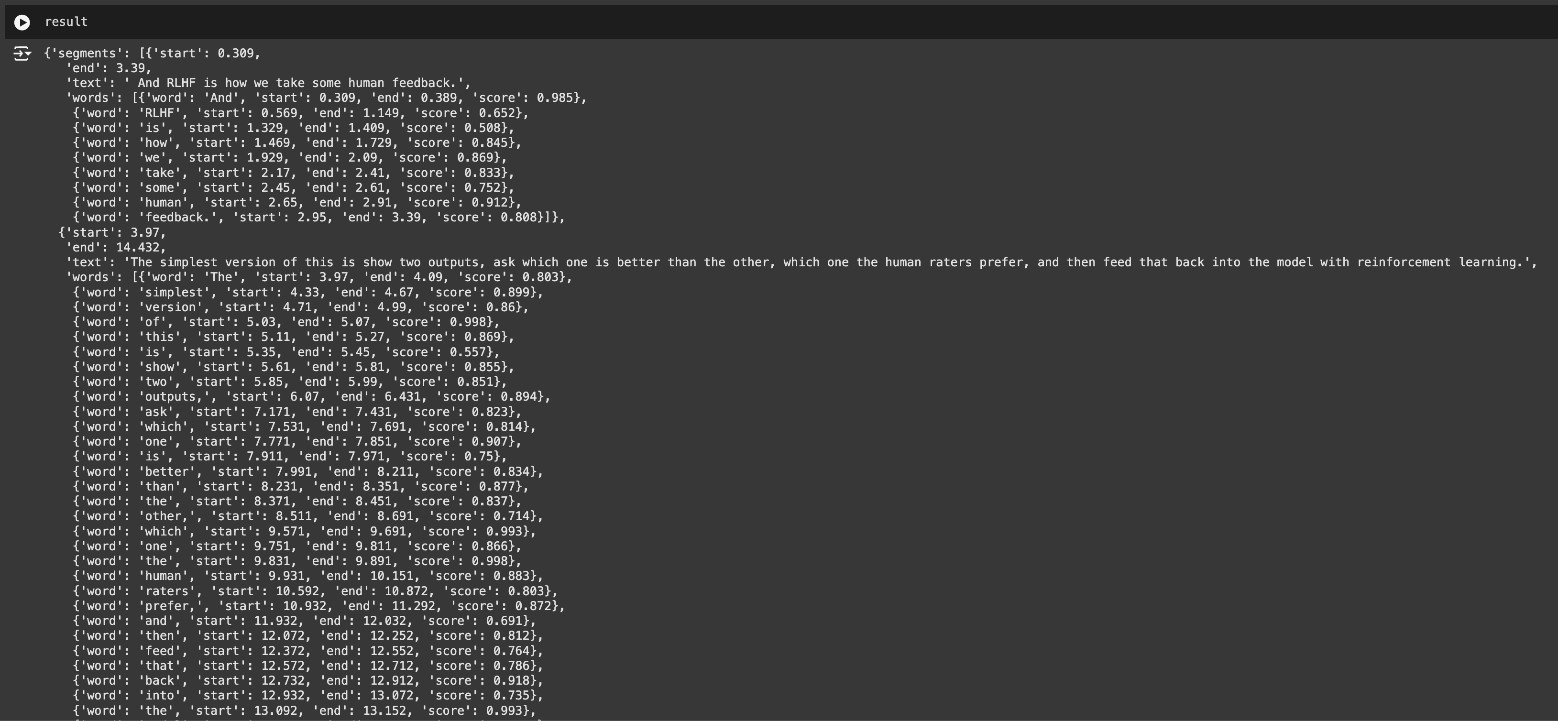
**2.1.2 Justification of Noise Reduction Techniques**  
Noise Reduction techniques are crucial in STT as they improve the signal-to-noise ratio (SNR) and provide cleaner input to the transcription model. Tools like noisereduce utilize spectral gating, where spectral features associated with noise are suppressed. This preprocessing significantly boosts the transcription accuracy.

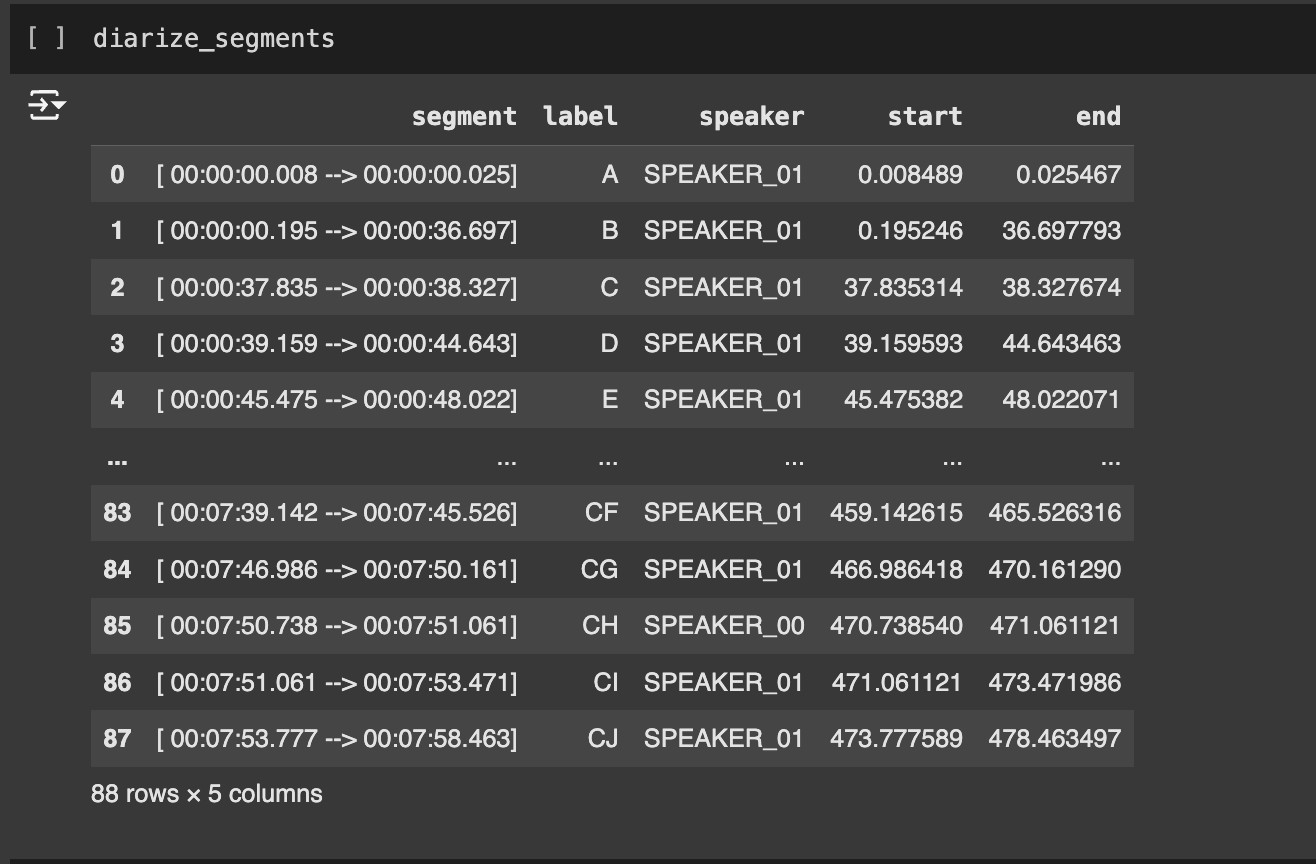
**2.2 Audio Segmentation Using pydub**  
For long audio files, it is essential to segment the audio into smaller, manageable chunks. This helps both in terms of memory efficiency and improving the performance of the transcription model. By segmenting the audio, we reduce the burden on the system during transcription and ensure the STT model processes each chunk independently, resulting in better overall performance.

**2.2.1 Audio Segmentation Rationalization**  
Audio segmentation is critical in handling larger datasets, improving batch processing efficiency, and enabling real-time applications where memory constraints may be present. Segmenting ensures that long silences, overlapping conversations, or high memory usage do not hinder the model's performance. [10]

**2.3 Speech Recognition and Diarization Using WhisperX**  
Once the audio is pre-processed, WhisperX, an advanced model built upon OpenAI's Whisper architecture, is utilized for speech recognition and speaker diarization. WhisperX is particularly effective in handling multi-speaker audio, generating accurate transcripts with identified speaker segments. WhisperX is integrated with a diarization module, which helps differentiate between speakers and provides time-stamped transcriptions.

**WhisperX Speech Recognition and Diarization**  
After the audio is segmented and cleaned, WhisperX processes the speech and performs speaker diarization.





**Table -1: Various methods applied for Speech-To-Text**

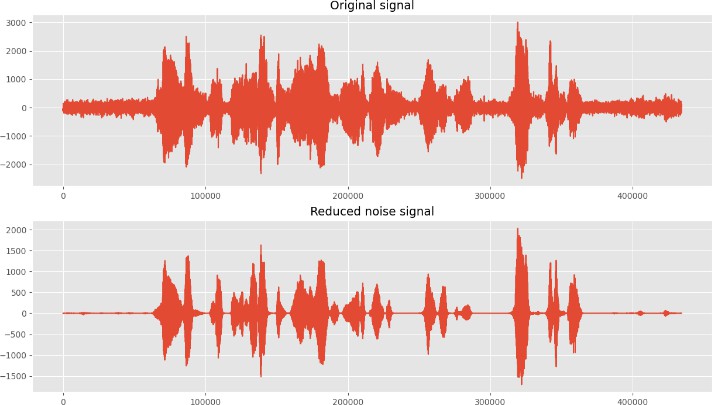
|  |  |  |
| --- | --- | --- |
| **S. No.** | **Technique Used** | **Description** |
| 1 | Automatic Speech Recognition, HMM model and human machine interface | The paper studied the deployment of STT by HMM and suggested to develop a machine interface system that depends on voice. The system could be deployed for helping 2 types  of users:   * People with disability who cannot access their email through use of mouse and keyboard, this category of users will be benefitted by the usage of a Speech-to-Text conversion system. * People who do not understand English or are not efficient in English and feel good to communicate in their native language i.e. English, Punjabi, Hindi. |
| 2 | MFCC and HMM | **[4]** Proposed a STT system replacing traditional MFCC with HMM. The conventional MFCC approach was less efficient in extracting the features from the speech signals hence a new approach was suggested using HMM. The features passed to the HMM network resulted in better feature recognition from the input audio in contrast with the MFCC method. HMM exhibited vast improvement in the quality of feature extraction from the audio resulting in better computational time and accuracy for a Speech-To-Text conversion system. |

|  |  |  |
| --- | --- | --- |
| 3 | Text processing, Text- To- Speech (TTS) synthesizer, Speech Enhancement | **[10]** Suggested that Test-to-Speech synthesizer is developing rapidly from past few years to gain the current shape. The most suitable methods for TTS are Formant, Articulator and Concatenative synthesis. Even in India some research organizations are also working on Text-to-Speech in regional languages like Marathi, Hindi, Telugu, Punjabi, Kannada, so on. A vast scope of improvement can be achieved in TSS synthesis to obtain a good amount of natural and emotion aspect. |
| 4 | S2T summarization | S2T summarization systems help by identifying the most relevant content within human speech and producing a con- densed form text suitable for the need. Extractive summarization selects relevant sentences or paragraphs from transcripts, but this method may sometimes lack cohesion and readability [6].  S2T summarization is usually achieved using a cascade approach, where an automatic speech recognition (ASR) model generates transcripts, followed by a text-to-text (T2T) summarization model that produces summaries [18]. |
| 5 | BERT and Bidirectional LSTM | [5] The variable-length abstractive summarization model is divided into a text segmentation module and a two-stage Transformer-based summarization module. The proposed text segmentation module, which utilizes BERT and Bidirectional LSTM, shows improved performance over existing methods. The two-stage Transformer-based summarization module combines extractive and abstractive methods to produce fluent and variable-length abstractive summaries. |
| 6 | Machine Learning, ANN ASR, Cuck Search Algorithm | [10] The paper summarizes the basic processes involved in a STT system which covers architecture of ASR(Automated Speech Recognition). The main focus for this paper is using Machine Learning in ASR, SVM, ANN with Cuckoo search algorithm along with ANN and back propagation classifier. The basic phases like: pre-processing, extraction of features and classification, of the STT system are studied by using machine learning. According to the generated results Hybridization of an algorithm with an optimization technique is considered better technique, traditional classifier results can be further improved by doing hybridization of it with other algorithms for optimization. |

|  |  |  |
| --- | --- | --- |
| 7 | RNN and Sentiment Analysis | The structure of RNN network and its various implementations such as RNN Transducer. RNNs are ideal in tasks such as speech to text and text summarization RNNs have a unique perspective on speech because they continuously strive to convert audio features into text sequences. This enables them to transform audio features into textual forms. Such an ability is advantageous as it allows a network to understand context in the spoken language [1].  RNNs are therefore able to compensate for varying patterns of speech and pronunciation accents that may exist across different individuals. These transcribers are more effective for RNNs’ greater transcription success increases their effectiveness in a broad range of applications including voice-command mechanics and automated transcription systems [3]. |

**2.1 Audio Preprocessing**

**2.1.1 Noise Reduction**  
Eliminating extraneous sound is very important for improving the quality of speech-to-text systems. This improvement also enhances the clarity of noise, is accomplished using the pydub library for loading and handling audio and noisereduce for appplying spectral gating, which reduces undesired background noise. The audio is first converted into a numpy array, enabling noisereduce to analyze and suppress lower-intensity frequencies associated with noise, while preserving key speech elements. The cleaned audio is then saved in a compatible format for further processing, ensuring clearer transcription results.



**Fig 2.2.1** Original Signal vs Reduced Noise Signal

**2.1.2 Justification of Noise Reduction Techniques**  
Noise reduction is crucial for STT systems, as background noise lowers transcription accuracy. Using spectral gating methods with noisereduce enables targeted suppression of noise frequencies, improving the signal-to-noise ratio. This ensures that speech is clearer and more accurately transcribed, especially in real-world noisy environments.

**2.2 Audio Segmentation**  
Segmenting audio into smaller, manageable chunks optimizes processing efficiency and enhances transcription accuracy. By dividing longer audio files, we can handle memory constraints better, allowing for smoother performance in STT processing.

**2.2.1 Audio Segmentation Rationalization**  
Audio segmentation is particularly beneficial in reducing processing load for long recordings and improving STT performance. Shorter audio segments reduce the risk of memory issues and make it easier to handle turn-taking or silences in multi-speaker scenarios, leading to clearer, more organized transcriptions.

**2.3 Speech Recognition and Diarization**  
Speech recognition and diarization are conducted using WhisperX, which transcribes audio and identifies speaker turns with timestamps. This enables accurate differentiation between speakers and assigns precise start and end times for each segment, making the STT output useful for multi-speaker environments.

**3. Proposed Model**

**3.1 Proposed Model: Wav2Vec 2.0**

* **Input**: Raw audio waveform.
* **Preprocessing**:
  + Basic preprocessing such as audio segmentation (optional, depending on audio length).
* **Architecture**:
  + **Wav2Vec 2.0 Encoder**: This component processes the raw audio and learns latent speech representations using a Convolutional Neural Network (CNN).
  + **Contextualized Representation**: Wav2Vec 2.0 uses Transformer layers on top of these representations to capture context and produce meaningful speech features.
  + **Fine-tuning with a CTC (Connectionist Temporal Classification) Loss**: Fine-tune Wav2Vec 2.0 on labelled speech datasets, using CTC to map learned representations directly to text.
* **Post-Processing**:
  + Language modelling and error correction using a pre-trained language model to improve grammatical correctness and fluency.
* **Output**: Transcribed text.

A diagram of a process flow

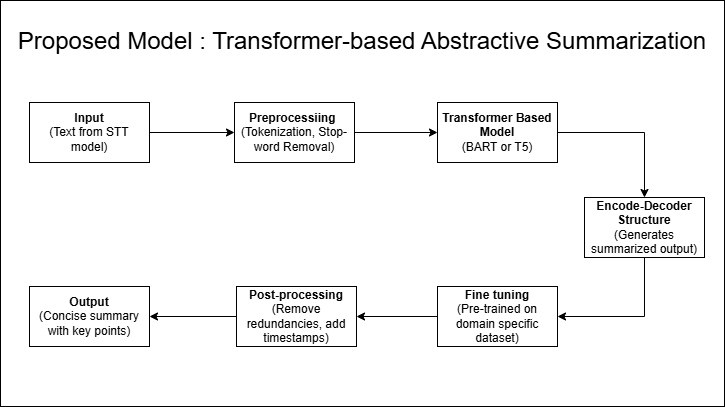
AI-generated content may be incorrect.

Wav2Vec 2.0 is more effective than traditional RNN/LSTM models, as it processes raw audio directly, improving both efficiency and accuracy, especially with fewer labelled data.

**3.2 Text Summarization Model:**

**Proposed Model: Transformer-based Abstractive Summarization (e.g., BART or T5)**

* **Input**: Text from the STT model.
* **Preprocessing**:
  + **Tokenization**: Breaking the text into tokens (words, subwords, or characters).
  + **Stop-word removal**: Eliminate unimportant words (e.g., “the”, “is”) to focus on relevant content.
* **Transformer-based model (BART or T5)**:
  + **Encoder-Decoder structure**: The encoder reads and processes the input text, while the decoder generates the summarized output. BART or T5 models excel at abstractive summarization by rewriting the input text into a more concise form.
  + **Pre-trained model fine-tuning**: Use pre-trained models like BART (Bidirectional and Auto-Regressive Transformers) or T5 (Text-to-Text Transfer Transformer) fine-tuned on your domain-specific dataset.
* **Post-Processing**:
  + Remove any redundant or repetitive content generated during summarization.
  + Add timestamps and format the summary text to reflect meeting structures.
* **Output**: Concise, human-readable summary with key points from the input text.



This model’s abstractive nature allows it to generate new sentences, rather than just extracting phrases, resulting in more natural and concise summaries.

1. Discussion
   1. Speech to Text conversion

In the following table, a comparative discussion is given about various approaches in speech to text and we have discussed major landmark achieved in this field as well as significant advantages and disadvantages to each approach. For example Hidden Markov Models ( HMMs) and Guassian Mixture Models have history with earlier days of speech recognition, where automatic speech recognition was borned outta these statistical models which lead to good useful work done on low resourced data sets etc but NOT in real time The need for more precise and flexible models has led to the increased use of deep learning-based models in projects such as Recurrent Neural Networks(RNNs) and Long Short Memory(LSTM), which are used since they can handle sequences without losing sight of dependencies.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technique** | **Description** | **Advantages** | **Limitations** | **Examples/Applications** |
| Hidden Markov Model (HMM) | Statistical model that uses probabilities to predict sequences of sounds to words. | Well-established, simple, and interpretable. | Struggles with noisy environments and diverse accents. | Early ASR systems, dictation software. |
| Gaussian  Mixture Model (GMM) | Used with HMMs to model  different sound distributions. | Improved acoustic modeling over pure HMMs. | Sensitive to variations in accents, noise, and requires large datasets. | Used in early versions of Siri, Google ASR. |
| Recurrent Neural Networks (RNN) | Neural network that processes sequences, making it  suitable for speech. | Better at capturing sequential dependencies than statistical models. | Struggles with long- range dependencies and complex speech patterns. | Real-time speech  recognition, language translation. |
| Long Short-Term Memory (LSTM) | A type of RNN designed to retain information over long sequences. | Handles long-range dependencies well, robust in noisy environments. | Computationally intensive, difficult to scale to large datasets. | Used in real-time  transcription, virtual assistants. |
| End-to-End Models | Models like transformers that directly convert speech to text without intermediate  steps. | Simplifies pipeline, high accuracy, adaptable to multiple languages. | Requires large datasets for training, may struggle with low- resource languages. | Wav2Vec, Deep Speech, automated transcription services. |
| Transformer- Based Models | Deep learning  models that process entire speech sequences in parallel. | High accuracy, faster processing, adaptable to various accents and dialects. | Complex to train, resource-intensive, large data  requirements. | Wav2Vec 2.0, Google’s Speech-to-Text API. |
| Cloud based Models | Cloud based speech to text model provided by various | Achieves high  accuracy in transcribing audio into text, making it suitable for various applications. | Limited customization options are available; also relies on a stable internet connection for optimal performance. | Google Cloud Speech-to- Text, Amazon Transcribe, Microsoft Azure Speech Service, IBM Watson Speech to Text. |

* 1. Text summarization

The comparative analysis of text summarization technologies presented in the table reveals a diverse array of approaches, each with its unique strengths and limitations. The table below list the most common used algorithms for text summarization.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Year** | **Algorithm** | **Key Developments** | **Pros** | **Cons** |
| 2016 | Rule-based Systems | Utilization of linguistic rules and heuristics for text generation. | Offers transparency and interpretability in the generation process. | Scalability is limited; crafting rules manually can be labor- intensive. |
| 2017 | Hidden Markov Models | Application in both speech recognition and generative tasks. | Established approach with a wide range of applications in text modeling. | Struggles to capture complex dependencies between words or phrases. |
| 2018 | Recurrent Neural Networks (RNN) | Enhanced mechanisms for improved context modeling, specifically for sequential data. | Effectively manages and processes sequential information, making it suitable for text generation. | Prone to issues like gradient vanishing; slow convergence during training can hinder performance. |
| 2020 | Generative Adversarial Networks (GANs) | Introduction of improved training techniques to enhance model stability. | Capable of producing diverse and realistic text samples that mimic human-like generation. | Often faces challenges related to training instability and mode collapse, which can affect quality. |
| 2021 | Long Short- Term Memory Networks  (LSTM) | Advanced architectures focused on capturing sequential dependencies in text data. | Effectively retains context over longer sequences, improving performance in tasks requiring  understanding of continuity. | May still struggle with very long-term dependencies, leading to loss of information over time. |
| 2022 | Facebook’s BART Model | Utilization of a transformer-based architecture specifically designed for summarization tasks. | Produces coherent and concise summaries, enhancing readability and comprehension. | Requires significant computational resources and can be complex to train effectively. |

**Architecture Diagram and System Overview**

A diagram of a computer

AI-generated content may be incorrect.

**System Architecture**

The proposed system integrates speech-to-text transcription and text summarization using an embedded hardware setup combined with software-driven natural language processing (NLP). The system architecture consists of an audio input module, microcontroller processing unit, storage, and power management components. The figure below represents the architecture of the hardware design used in this study.

**Hardware Components and Description**

1. **MAX9814 Electret Microphone Module**
   * This is the primary audio capture device used to record speech input. It features automatic gain control (AGC) to optimize audio quality for speech processing.
2. **Arduino Nano (ATmega328P)**
   * The microcontroller is responsible for managing data acquisition, pre-processing signals from the microphone, and controlling peripheral components like storage and RTC.
3. **DS1302 Real-Time Clock (RTC) Module + CR2032 Battery**
   * The RTC module provides accurate timestamping for recorded speech data, ensuring the logs are synchronized for further processing.
4. **SD Card Reader Module (SPI-based) & Micro SD Card (8GB - 32GB, FAT32)**
   * The system stores captured speech data and processed text outputs on an SD card for later retrieval and analysis.
5. **USB Mini B Cable & USB Power Bank (2000mAh - 10,000mAh)**
   * The microcontroller and peripherals require a steady power source, which is provided via a USB power bank.
6. **Breadboard & Jumper Wires**
   * Used for circuit prototyping and establishing connections between various hardware components.
7. **Tactile Push Buttons (3x)**
   * These buttons are used for user interaction, such as starting and stopping recordings or triggering summarization functions.
8. **LEDs (Red/Green/Blue, 5mm) – 4**
   * Status indicators for system processes like recording, processing, and completion of transcription/summarization.
9. **Resistors (220Ω - 6, 10kΩ - 6)**
   * These resistors are used for current regulation in the circuit, ensuring stable operation of LEDs and buttons.
10. **Capacitors (0.1µF (104) Ceramic Capacitor - 2)**
    * These capacitors help in power stabilization and noise filtering in the circuit.

**Functionality and Workflow**

The MAX9814 microphone picks up audio which is then processed by the Arduino Nano. The audio is stored on the SD card so it can be transcribed at a later date. Once the speech-to-text conversion is achieved with modern NLP techniques, the text goes through a multi-step summarization process which includes both extractive and abstractive techniques. This final processed output can be retrieved and used for practical applications such as transcription of meetings, virtual assistants, and content summarization tools.

This is the initial attempt at incorporating speech recognition with summarisation to embedded systems in order to facilitate automatic transcription processes.

**References-**

[1] M. Mehta, K. Gupta, S. Tiwari and Anamika, "A Review on Sentiment Analysis of Text, Image and Audio Data," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1660-1667, doi: 10.1109/ICCMC51019.2021.9418360.

[2] V. Maheshwar Reddy, K. Deepika, K. Adithya Surya Prakash, and M. Sanathan, "A survey on audio analysis: Text characterization and summarization," World Journal of Advanced Research and Reviews, vol. 21, no. 03, pp. 1596–1601, 2024, doi: 10.30574/wjarr.2024.21.3.0789.

[3] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," arXiv:1303.5778 [cs.NE], 2013, doi: <https://doi.org/10.48550/arXiv.1303.5778>

[4] Ghadekar, Premanand & Anand, Divsehaj & Gupta, Aryan & Oswal, Preeti & Sharma, Dheeraj & Khare, Shreyas. (2023). Audio Based Text Summarization Using Natural Language Processing. 10.1007/978-981-99-3656-4\_17.

[5] M. -H. Su, C. -H. Wu and H. -T. Cheng, "A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 2061-2072, 2020, doi: 10.1109/TASLP.2020.3006731.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., & Gomez, A. N. (2017). Attention is all you need [J]. Advances in neural information processing systems, 30(1), 261-272.

[7] K. Yao, L. Zhang, D. Du, T. Luo, L. Tao and Y. Wu, "Dual Encoding for Abstractive Text Summarization," in IEEE Transactions on Cybernetics, vol. 50, no. 3, pp. 985-996, March 2020, doi: 10.1109/TCYB.2018.2876317.

[8] Dhumal, Priyanka & Sutar, Sudarshan & Surve, Indraneel & Munawwar, Mirza & Nanaware, Vishal. (2024). Text Summarization Using NLP. International Journal of Advanced Research in Science, Communication and Technology. 319-324. 10.48175/IJARSCT-18650.

[9] Monteiro, R., Pernes, D. (2023). Towards End-to-End Speech-to-Text Summarization. In: Ekštein, K., Pártl, F., Konopík, M. (eds) Text, Speech, and Dialogue. TSD 2023. Lecture Notes in Computer Science(), vol 14102. Springer, Cham. <https://doi.org/10.1007/978-3-031-40498-6_27>

[10] A. Vartakavi, A. Garg and Z. Rafii, "Audio Summarization for Podcasts," 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 2021, pp. 431-435, doi: 10.23919/EUSIPCO54536.2021.9615948.

[11] Vinnarasu, A., and Deepa V. Jose. "Speech to text conversion and summarization for effective understanding and documentation." International Journal of Electrical and Computer Engineering (IJECE) 9, no. 5 (2019): 3642-3648.

[12] D. Y. S. R. Ibrahim Patel, “SPEECH RECOGNITION USING HMM WITH MFCC- AN,” 2010.

[13]. S. R. Mache, “Review on Text-To-Speech Synthesizer,” International Journal of Advanced Research in Computer and Communication Engineering, 2015.

[14]. Gupta, S., Gupta, S.K.: Abstractive summarization: An overview of the state of the art. Expert Systems with Applications 121, 49–65 (2019). https://doi.org/https://doi.org/10.1016/j.eswa.2018.12.011

[15]. Rezazadegan, D., Berkovsky, S., Quiroz, J.C., Kocaballi, A.B., Wang, Y., Laranjo, L., Coiera, E.W.: Automatic speech summarisation: A scoping review (2020). <https://doi.org/10.48550/arXiv.2008.11897>

[16] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," arXiv:1303.5778 [cs.NE], 2013, doi: <https://doi.org/10.48550/arXiv.1303.5778>

[17] M. Mehta, K. Gupta, S. Tiwari and Anamika, "A Review on Sentiment Analysis of Text, Image and Audio Data," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1660-1667, doi: 10.1109/ICCMC51019.2021.9418360.

[18] Prakash, N.C., Narasimhaiah, A.P., Nagaraj, J.B., Pareek, P.K., Maruthikumar, N.B. and Manjunath, R.I., 2022. Implementation of NLP based automatic text summarization using spacy. International Journal of Health Sciences, 6, pp.7508-7521

[19] Ming-Hsiang Su, Chung-Hsien Wu, Senior Member IEEE, and Hao-Tse Cheng, “A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization” DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing <http://www.ieee.org/publications_standards/publications/rights/index.html>.

[20]. N. T. D. B. Sunanda Mendiratta, “A Robust Isolated Automatic Speech Recognition System using Machine Learning Techniques,” 2019

[21] Maghilnan, S., and M. Rajesh Kumar. "Sentiment analysis on speaker-specific speech data." 2017 International Conference on Intelligent Computing and Control (I2C2). IEEE, 2017.

[22] Lu, Zhiyun, et al. "Speech sentiment analysis via pre-trained features from end-to-end ASR models." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

[23] Murarka, Aishwarya, et al. "Sentiment Analysis of Speech." International Journal of Advanced Research in Computer and Communication Engineering 6.11 (2017): 240-243.

[24] M.F. Mridha: A Survey of Automatic Text Summarization: Progress, Process and Challenges,2021.

[25] Wang Lin,Tao Jin,Ye Wang: TAVT:Towards Transferable Audio-Visual Text Generation, Zhejiang University,2022.

[26] Pratima Mohan Thorat , Prof. Dr. M. S. Bewoor: A Novel Approach for Voice Based Text Summarizer,2022

[27] Oriol Vinyals, Suman Ravuri, and Daniel Povey, “Revisiting Recurrent Neural Networks for Robust ASR,” in ICASSP, 2012.

[28] A. Maas, Q. Le, T. O’Neil, O. Vinyals, P. Nguyen, and A. Ng, “Recurrent neural networks for noise reduction in robust ASR,” in INTERSPEECH, 2012.

[29] Abdel rahman Mohamed, Dong Yu, and Li Deng, “Investigation of full-sequence training of deep belief networks for speech recognition,” in Interspeech, 2010.

[30] Kam-Chuen Jim, C.L. Giles, and B.G. Horne, “An analysis of noise in recurrent neural networks: convergence and generalization,” Neural Networks, IEEE Transactions on, vol. 7, no. 6, pp. 1424–1438, Nov. 1996.

[31] Sadaoki Furui, Tomonori Kikuchi, Yousuke Shinnaka, and Chiori Hori, Speech-to-Text and Speech-to-Speech Summarization of Spontaneous Speech

[32] Rajesh S. Prasad, Dr. U.V. Kulkarni, Machine Learning in Evolving Connectionist Text Summarizer

[33] M. Saadeq Rafieee, Somayeh Jafari, Considerations to Spoken Language Recognition for Text-to-Speech Applications

[34] F. Violaro, O. Boeffard, A Hybrid Model for Text-to-Speech Synthesis.

[35] K. Y. Chen, S. H. Liu, B. Chen, H. M. Wang, E. E. Jan, W. L. Hsu, and H. H. Chen, “Extractive broadcast news summarization leveraging recurrent neural network language modeling techniques,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 23, no. 8, pp. 1322-1334, May 2015.

[36] K. Al-Sabahi, Z. Zuping, and M. J. I. A. Nadher, "A Hierarchical Structured Self-Attentive Model for Extractive Document Summarization (HSSAS)," IEEE Access, vol. 6, pp. 24205-24212, Apr. 2018.

[37] Y. Liu, "Fine-tune BERT for Extractive Summarization," arXiv preprint [Online], 2019, Available: arXiv:1903.10318.

[38] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in Proc. the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 93-98.

[39] H. Zhang J. Xu, and J. Wang, "Pretraining-Based Natural Language Generation for Text Summarization," arXiv preprint [Online], 2019, Available: arXiv:1902.09243, 2019.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint [Online], 2018, Available: arXiv:1810.04805, 2018.

[41] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In Proc. of NAACL, 2016.

[42] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), Shanghai, China, Mar. 2016, pp. 4945–4949.

[43] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, “An investigation on the accuracy of truncated DKLT representation for speaker identification with short sequences of speech frames,” IEEE Trans. Cybern., vol. 47, no. 12, pp. 4235–4249, Dec. 2017.

[44] A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” in Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP), Lisbon, Portugal, Sep. 2015, pp. 379–389.

[45] S. Chopra, M. Auli, and A. M. Rush, “Abstractive sentence summarization with attentive recurrent neural networks,” in Proc. Conf. North Amer. Assoc. Comput. Linguist. Human Lang. Technol., San Diego, CA, USA, Jun. 2016, pp. 93–98.

[46] R. Nallapati, B. Zhou, C. N. dos Santos, Ç Gülçehre, and B. Xiang, “Abstractive text summarization using sequence-to-sequence RNNs and beyond,” in Proc. 20th SIGNLL Conf. Comput. Nat. Lang. Learn. (CoNLL), Berlin, Germany, Aug. 2016, pp. 280–290.

[47] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, and H. Jiang, “Distraction-based neural networks for document summarization,” CoRR, vol. abs/1610.08462, 2016. [Online]. Available: <http://arxiv.org/abs/1610.08462>.

[48] K. Shetty and J. S. Kallimani, “Automatic extractive text summarization using K-means clustering,” in 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017, pp. 1–9.

[49] R. Boorugu and G. Ramesh, “A survey on NLP-based text summarization for summarizing product reviews,” in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 352–356.

[50] S. Adhikari, “NLP based machine learning approaches for text summarization,” in 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 535–538.

[51] S. Jugran, A. Kumar, B. S. Tyagi, and V. Anand, “Extractive automatic text summarization using SpaCy in Python & NLP,” in 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021, pp. 582–585.

[52] N. C. Prakash, A. P. Narasimhaiah, J. B. Nagaraj, P. K. Pareek, N. B. Maruthikumar, and R. I. Manjunath, “Implementation of NLP based automatic text summarization using spacy,” International Journal of Health Sciences, vol. 6, pp. 7508–7521, 2022.

[53] Zhuang, L., Wayne, L., Ya, S., Jun, Z.: A robustly optimized BERT pre-training approach with post-training. In: Proc. of the 20th Chinese National Conference on Computational Linguistics. pp. 1218–1227. Chinese Information Processing Society of China, Huhhot, China (Aug 2021). <https://doi.org/10.48550/arXiv.1907.11692>

[54] Vartakavi, A., & Garg, A. (2020). PodSumm: Podcast audio summarization. In PodRecs: The Workshop on Podcast Recommendations. September 25, 2020.

[55] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. In 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, November 3–7, 2019.

[56] Furui, S., Kikuichi, T., Shinnaka, Y., & Hori, C. (2003). Speech-to-speech and speech to text summarization. In First International Workshop on Language Understanding and Agents for Real World Interaction, Sapporo, Japan, July 13, 2003.

[57] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, June 2–7, 2019.

[58] K. M. Shivakumar, V. Ravi, “A Survey on Extractive Text Summarization Approaches,” in Proceedings of the 2021 International Conference on Computational Intelligence and Knowledge Economy, pp. 327-336, 2021.