Localized Speech-to-Text for Inclusive Learning

Pranav H
Dept. of Computer Science
Amrita School of Computing
Amrita Vishwa Vidhyapeetham,
Bengaluru, India

Mayank Pandey
Dept. of Computer Science
Amrita School of Computing
Amrita Vishwa Vidhyapeetham,
Bengaluru, India

I. INTRODUCTION

The most natural way for communication as humans is speech, and there have been constant efforts to make speech a viable and effect way of communication with computing devices, this is achieved with the ever-evolving toolset known as speech to text. These tools constantly evolving as they are , have failed to account for variations in dialects [3] and pronunciation of words along with being available for regional and vernacular languages which varies the impact and effectiveness of these tools from region to region [4]. In this context, it is empirical that we strive to improve these tools to overcome this disparity, making them available to all irrespective of region or language.

In light of the pressing need for these technological tools to be not just widely accessible, but also effectively utilized across diverse regions and languages, this paper puts forth a novel proposition. It suggests the deployment of an on-device, self-supervised [5], speech-to-text module that has been localized specifically for the multitude of regional languages spoken across India. The primary objective of this module is not merely to transcribe speech to text, but to serve a greater purpose - to act as a facilitator in the learning process. By catering to the unique linguistic nuances of regional Indian languages[6,7], this module aims to bridge the gap between technology and effective learning, thereby making education more inclusive and comprehensive.

Here we strive to implement the following tools to facilitate our goal towards a smart and inclusive classroom :

- Discussion Logs : Leveraging Audio Fingerprinting and text independent speaker recognition systems [8] to keep an individualised, speaker separated log of the class room discussions, which will enable a quick and easy review of the classroom discussions.
- Summarization : Using the discussion logs made, prepare a short summary to assist in quick review.
- Individualised Remainders and Summary : Leveraging the discussion logs to make a personalised summary tailored to the individual comprising of individual assigned tasks and conversations.
- Live Captioning and Translation : Enable Seamless Real Time Captioning [9] and Translation of the classroom to the user's chosen language.
- Evaluation Aid : Identify the key words used in response to a question and cross referencing [10] it with the expected key words and helping the evaluator during a Viva.

The Scope for the use of this tool is vast and the speech to text module proposed in this paper can be leveraged to assist beyond these tools implemented herein.

II. LITERATURE REVIEW

This collection of research papers provides a comprehensive overview of advancements in the field of Speech-to-Text (S2T) technologies. The authors explore various aspects of S2T, including neural basis of speech production and comprehension, personalized federated learning, voice synthesis, text-to-speech conversion, and keyword extraction techniques. These studies highlight the ongoing efforts to improve the efficiency, accuracy, and personalization of S2T systems, while also addressing challenges such as communication overhead, performance degradation, aperiodic distortion, and the impact of punctuation on keyword extraction. The research underscores the potential of S2T technologies in facilitating seamless human-computer interaction and advancing the field of natural language processing.

Ariel Goldstein et al. [11] investigated how the brain processes real-world speech by utilizing a powerful speech-to-text model known as Whisper. By analyzing the model's predictions about brain activity in response to different aspects of speech, they discovered a widespread network across the brain's outer layer (cortex) dedicated to speech and language. This network involves sensory and motor areas responding to the sound features of speech, while higher-level language regions activate in response to grammatical structure and meaning. This research unveils a complex and distributed brain system responsible for understanding and producing spoken language in everyday situations..

Yichao Du et al. [12]proposed a new approach that tackles two problems commonly faced in training Speech-to-Text systems: high communication costs and inaccurate performance due to diverse data across participants. To address the first issue, a lightweight module called FedLoRA is used by individual devices (clients) to tune their models and interact with a central server, significantly reducing communication needs. Additionally, a technique called FedMem personalizes the central model by incorporating a special classifier that learns from each client's unique data patterns. This combined approach effectively personalizes the model while minimizing communication overhead across various S2T tasks.

Praveena Nuthakki et al. [13] discuss the challenges of traditional concatenation speech synthesis technologies and propose a new approach for voice synthesis. They suggest that their model’s minimal aperiodic distortion makes it an excellent candidate for a communication recognition model. The authors also highlight the need for more robust network foundations and optimization methods for their proposed algorithm to perform at its best. They believe that their approach is as close to human speech as possible, despite the fact that speech synthesis has a number of audible flaws.

Swaroopa Shastri et al. [14] explore the conversion of English to Hindi, first to text, and subsequently to speech. They use two approaches for text character recognition from images: a maximally stable extensible region (MSER) and grayscale conversion. The paper also deals with geometric filtering in combination with stroke width transform (SWT). After detecting text sequences and fragmenting them into words, a 96 percent accurate spell check is performed using naive Bayes and decision tree algorithms. Finally, they use optical character recognition (OCR) to digitize the text and a text-to-speech synthesizer (TTS) to convert it to Hindi.

Nuhu Kontagora et al. [15] This study explored the effectiveness of common keyword extraction methods on speech-to-text data. They developed a new audio dataset and compared two popular methods, RAKE and TextRank, with different settings on both the original written text and its corresponding speech-to-text version. They measured performance using precision, recall, and F-score. TextRank with a specific list of stop words (FOX) performed best on both text and audio, achieving F-scores of 16.59% and 14.22%, respectively. While the audio F-score was lower, it was still considered suitable for applications involving spoken conversations. Notably, the lack of punctuation in the speech-to-text data negatively impacted all methods tested.

Adjila et al. [16] provides a novel method for locating and removing silent portions from voice signals. In applications like automatic voice segmentation and speech recognition, this strategy greatly improves system performance and accuracy by utilizing the continuous average energy of the signal. Notably, it maintains lower computing complexity and outperforms contemporary approaches based on spectral centroid and multi-scale product. The MATLAB evaluation of the study shows that the suggested strategy is resilient when dealing with speech signals in Arabic, French, and English.

III. METHODOLOGY

The Below experiments were conducted :

A1. Spectral Transformation:
Utilize `numpy.fft.fft()` to transform the speech signal into its spectral domain.
Plot the amplitude part of the spectral components to visualize the frequency distribution.
Observe the spectral characteristics of the speech signal.

A2. Inverse Transformation:
Use `numpy.fft.ifft()` to inverse transform the frequency spectrum back to the time domain.
Compare the generated time domain signal with the original signal to assess the fidelity of the transformation.
Evaluate how accurately the spectral information is preserved during the transformation process.

A3. Spectral Analysis of a Word:
Isolate a specific word present in the recorded speech.
Perform spectral analysis on this word using `numpy.fft.fft()`.
Compare the spectrum of the isolated word with the spectrum of the full signal to understand its contribution to the overall spectral composition.

A4. Rectangular Window Analysis:
Define a rectangular window of 20 milliseconds sampled at 22.5 KHz.
Analyze the spectral components within this window using FFT.
Gain insights into the spectral characteristics of the speech signal within a specific time frame.

A5. Windowed Analysis with Heatmap:
Break down the speech signal into window lengths of 20 milliseconds intervals.
Utilize `numpy.fft.rfft()` to evaluate the frequency components within each window.
Stack these frequency components as columns in a matrix.
Visualize the matrix using a heatmap plot to understand the distribution of spectral components across different time intervals.

A6. Spectrogram Plotting:
Utilize `scipy.signal.spectrogram()` to generate the spectrogram of the speech signal.
Plot the spectrogram over the same duration as the previous analyses.
Compare the spectrogram plot with the previous plots to assess similarities or differences in spectral representation and gain a comprehensive understanding of the speech signal's spectral characteristics.

IV. Results

Fig.1 Show’s the amplitude, of the spectral components, of the speech signal converted to the spectral domain.

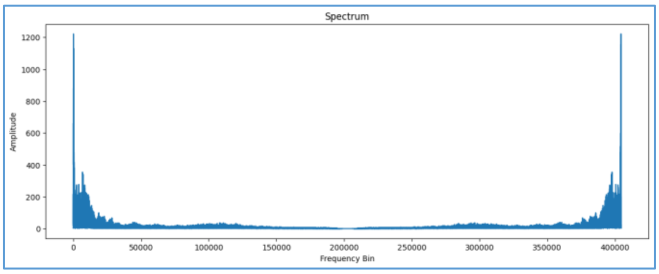

Fig. 1 Amplitude of the Spectral Components

Fig.2 Shows that the IFFT signal and the original signal are similar and can be audibly identical

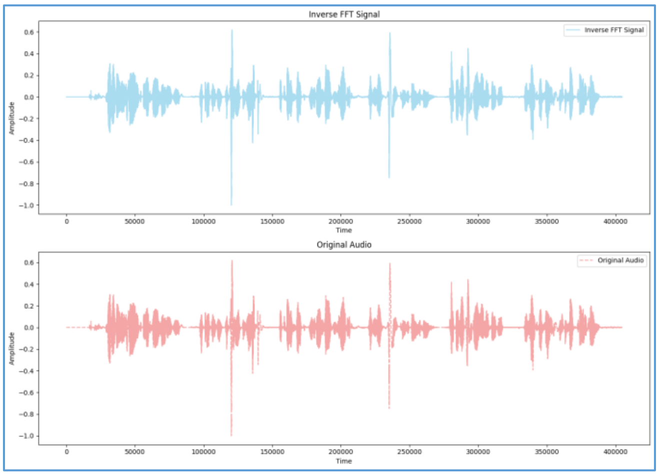

Fig2. Inverse FFT Signal and Original Signal Comparison

Fig.3 shows the spectral analysis of a rectangular window of 20 milli-second sampled at 22.5 KHz.

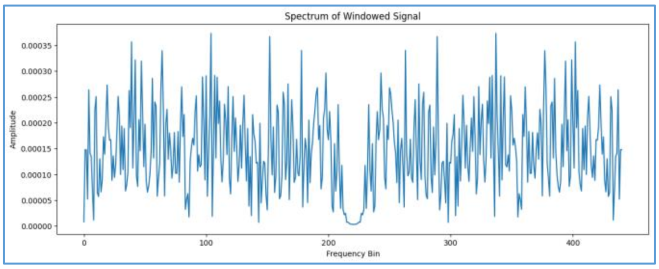

Fig3. Windowed Signal Spectrum

The speech signal is then segmented into 20 mSec intervals, evaluated using numpy.fft.rfft(), and represented as a heatmap matrix, as illustrated in Fig.4, Fig 5 shows the scipy Spectrogram.

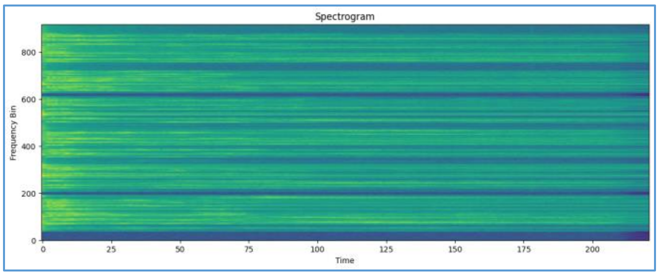

Fig 4. Heatmap Matrix Spectrogram

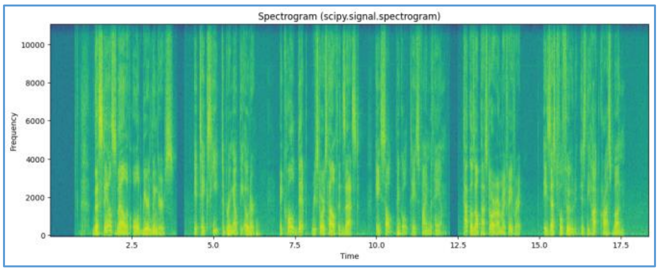

Fig5. Scipy Spectrogram

V. Conclusion

In conclusion, the series of experiments conducted on the speech signal provided valuable insights into its spectral characteristics and transformations. Through spectral transformation using `numpy.fft.fft()`, we visualized the frequency distribution of the speech signal, gaining an understanding of its spectral composition. The inverse transformation with `numpy.fft.ifft()` allowed for the reconstruction of the time-domain signal, facilitating comparison with the original signal to assess transformation fidelity.

Further analysis focused on specific segments, such as isolating a word for spectral analysis. This provided an understanding of how individual components contribute to the overall spectral composition of the speech signal. Additionally, rectangular window analysis shed light on spectral characteristics within specific time frames, offering insights into localized features of the signal.

The windowed analysis with heatmap visualization proved effective in capturing the distribution of spectral components across different time intervals. By stacking frequency components as columns in a matrix and visualizing it as a heatmap, we gained a comprehensive view of how spectral features evolve over time. Moreover, spectrogram plotting using `scipy.signal.spectrogram()` provided a holistic representation of the speech signal's spectral characteristics, allowing for comparison with previous analyses.

Overall, these experiments underscored the importance of spectral analysis in understanding the structure and content of speech signals. The combination of transformation techniques, windowed analysis, and spectrogram plotting facilitated a thorough exploration of the speech signal's spectral properties, enhancing our comprehension of its acoustic features and aiding in various speech processing applications.

References

[1] Y. Wei, J. Xiong, H. Liu, Y. Yu, J. Pan, and J. Du, “AdaStreamLite: Environment-adaptive Streaming Speech Recognition on Mobile Devices,” in Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 7, no. 4, Association for Computing Machinery, New York, NY, USA, December 2023, Art. No. 187, pp. 1-29. doi: 10.1145/3631460

[2] J. Laures-Gore, C. R. Rogers, H. Griffey, K. G. Rice, S. Russell, M. Frankel, and R. Patel, “Dialect identification, intelligibility ratings, and acceptability ratings of dysarthric speech in two American English dialects,” in Clinical Linguistics & Phonetics, Taylor & Francis, pp. 1-12. doi: 10.1080/02699206.2023.2301337.

[3] S. Feng, B. M. Halpern, O. Kudina, and O. Scharenborg, “Towards inclusive automatic speech recognition,” in Computer Speech & Language, vol. 84, 2024, 101567, ISSN 0885-2308. doi: 10.1016/j.csl.2023.101567.

[4] V. Karthikeyan and S. Suja Priyadharsini, “Modified layer deep convolution neural network for text-independent speaker recognition,” in Journal of Experimental & Theoretical Artificial Intelligence, vol. 36, no. 2, Taylor & Francis, 2024, pp. 273-285. doi: 10.1080/0952813X.2022.2092560.

[5] P. Gambhir, A. Dev, P. Bansal, and D. K. Sharma, “End-to-end Multi-modal Low-resourced Speech Keywords Recognition Using Sequential Conv2D Nets,” in ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 23, no. 1, Association for Computing Machinery, New York, NY, USA, January 2024, Art. No. 7, pp. 1-21. doi: 10.1145/3606019.

[6] Devare, M. ., & Thakral, M. . (2023). Enhancing Automatic Speech Recognition System Performance for Punjabi Language through Feature Extraction and Model Optimization. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(8s), 307–313.

[7] F. Wu et al., “Wav2Seq: Pre-Training Speech-to-Text Encoder-Decoder Models Using Pseudo Languages,” in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5. doi: 10.1109/ICASSP49357.2023.10096988.

[8] Y. Wei, J. Xiong, H. Liu, Y. Yu, J. Pan, and J. Du, “AdaStreamLite: Environment-adaptive Streaming Speech Recognition on Mobile Devices,” in Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 7, no. 4, Association for Computing Machinery, New York, NY, USA, December 2023, Art. No. 187, pp. 1-29. doi: 10.1145/3631460.

[9] L. Liu, L. Liu, and H. Li, “Computation and Parameter Efficient Multi-Modal Fusion Transformer for Cued Speech Recognition,” arXiv preprint arXiv:2401.17604, 2024. Primary Class: cs.CV.\

[10] R. Shukla, “Keywords Extraction and Sentiment Analysis using Automatic Speech Recognition,” arXiv preprint arXiv:2004.04099, 2020. Primary Class: eess.AS.

[11] Goldstein, Ariel, et al. "Deep speech-to-text models capture the neural basis of spontaneous speech in everyday conversations." *bioRxiv* (2023): 2023-06.

[12] Du, Yichao, et al. "Communication-Efficient Personalized Federated Learning for Speech-to-Text Tasks." *arXiv preprint arXiv:2401.10070* (2024).

[13] Nuthakki, Praveena, et al. "Deep Learning based Multilingual Speech Synthesis using Multi Feature Fusion Methods." *ACM Transactions on Asian and Low-Resource Language Information Processing* (2023).

[14] Shastri, Swaroopa, and Shashank Vishwakarma. "An Efficient Approach for Text-to-Speech Conversion Using Machine Learning and Image Processing Technique."

[15] Nuhu Kontagora, Bello, et al. "Performance Evaluation of Keyword Extraction Techniques and Stop Word Lists on Speech-To-Text Cor." (2023).

[16] Adjila, A., Ahfir, M. and Ziadi, D., 2021, December. Silence Detection and Removal Method Based on the Continuous Average Energy of Speech Signal. In *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)* (pp. 1-5). IEEE.

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line