

Speech-to-Text Note-Taking Application

Adham Qussay , Karma el gendy , Hana Refaat , Bassel youssef , yusuf El kordy
Faculty of Computer Science
ESLSCA University
Cairo, Egypt

Abstract—This paper introduces a Speech-to-Text Note-Taking Application tailored for educational use, aiming to convert spoken language into text with high precision and efficiency. The application supports various audio input formats, including real-time microphone recordings, file uploads, and YouTube video URLs. It leverages advanced models such as Wav2Vec2 and HuBERT for transcription, T5 for punctuation correction, and summarization. Built with Streamlit, the interface offers a dynamic and interactive user experience, allowing users to customize transcription settings and easily access results. The application's performance was evaluated qualitatively, demonstrating its effectiveness in transcribing different accents, handling background noise, correcting punctuation, summarizing text, and diarizing speakers. Overall, the application serves as a valuable tool for educational settings, offering accurate and contextually relevant transcriptions to aid in lectures, note-taking, and content accessibility.

I. Introduction

As education and working progress rapidly, providing a way of converting spoken language to text in shorter time proves highly useful in contemporary educational and working setting. The current situation where most learning is done online, meetings are conducted virtually, and other content is shared electronically has increased the need for proper and efficient STT. These tools are crucial to assist with increasing the access, capturing of notes, and increasing the effectiveness on working. This has created the need for highly advanced applications in STT technology to the extent of producing those that are accurate and easily intelligible.

This paper presents an Educational Speech-to-Text Note-taking Application, whose main function is to transcribe as accurately as possible and as fast as possible, spoken content. Regarding the audio input type, the application is able to accept real-time microphone, file and even YouTube video links making it flexible that

easily meets the user's demands. It uses the state of art models like Wav2Vec2 and HuBERT for transcription, T5 for punctuation correction and summarization and has a simple user interface created with Streamlit. These models are chosen due to their high quality and capability to work with the difficult range of human voice and potential interlocutors' accents and background noise.

The application is designed to cover a range of issues that is typical for the educational environment individuals: different accents, noise, and organization of the material into punctuated and summarized parts. In this way, the application takes into account the state of the art models which in turn provide precise transcriptions that relate to the context of the learning process enabling both the students and educators benefit. The T5 utilization for the automatic punctuation/correcting as well as summarization is useful for transcribing lecture notes and reviewing the content particularly when studying from notes.

Furthermore, the app includes functionalities like speaker diarization this separates the audio source to different segments of speakers which is an import aspect of transcribing group conversations or group meetings and interviews. This integration can prove valuable in the case of educational videos and webinars where the application can help with transcribing the content and creating notes, extending the application past the classroom experience.

The following sections of this paper will delve into the literature review, methodology, application overview, and performance evaluation. The literature review will explore recent advancements in STT technology and the rationale behind selecting specific models and libraries. The methodology section will detail the technical processes and algorithms employed in developing the

application. The application overview will describe the user interface, features, and functionalities in detail, while the performance evaluation will present qualitative and quantitative assessments of the application's effectiveness in various educational scenarios. Through this comprehensive examination, the paper aims to demonstrate the significant impact and potential of the Speech-to-Text Note-Taking Application in enhancing educational experiences and outcomes.

II. Literature Review

The following figure demonstrates a review of recent advancements in STT technology, evaluating models from Google, IBM, Amazon, and Microsoft, ultimately selecting Hugging Face. The chosen model, combined with Python libraries and frameworks, is used to develop our application.

Model	Key Features	References
Google Speech-to-Text	Recurrent sequence generators with an attention mechanism. Competitive PER of 17.6%.	Chorowski et al. [7]
IBM Watson Speech to Text	End-to-end deep learning architecture. Superior performance in noisy conditions. Achieves 16.0% error rate on Switchboard Hub5'00.	Hannun et al. [8]
Amazon Transcribe	Utilizes Amazon Mechanical Turk for transcription. High accuracy comparable to traditional methods. Cost-effective.	Marge, Banerjee, and Rudnický [9]
Microsoft Azure Speech Service	Multilingual voice recognition. Easy integration with other Azure services. High precision and detailed documentation.	Xuedong Huang et al. [10]
Hugging Face	Open-source platform. Offers models like Wav2Vec2, HuBERT, Whisper. Customizable and cost-effective. Strong community support. Advanced features like speaker diarization and real-time audio processing.	[11], [12], [13], [14], [15]

III. Methodology

Application Overview

The Speech-to-Text (STT) Note-Taking Application serves the purpose of transforming spoken language into text. This is done with the maximum level of accuracy and efficiency possible. It

supports multiple audio input forms, including real-time microphone recordings, file uploads, and YouTube video URLs. The operations of transcription, punctuation, and speaker diarization undertaken in this application have employed state-of-the-art models for support, with its making accurate and usable through an implementation done in Python using modern deep learning frameworks and focused on user interaction suitable for a Streamlit interface.

Audio Handling and Preprocessing

- Effective audio handling and preprocessing are critical to ensure the compatibility and quality of the input data for speech recognition models. In this application, the following Python libraries were imported to fulfill this task:
- Librosa: This robust library is used for audio analysis and preprocessing. It allows us to load the audio files and resample them to a required 16 kHz, as well as extract various features necessary for achieving high transcription accuracy. Resampling is especially important since models used here have been trained on 16 kHz data, and the sample rate is an essential factor in their proper work. Denoising and normalization can also be done because Librosa provides functions that clean the audio data and make them consistent.
- Audioread: Audio read is a backend library that provides a universal interface for reading audio files of different formats. This guarantees the application will handle various audio sources through MP3, MP4, WAV, etc., without compatibility problems. The use of Audioread ensures that audio data is fetched and processed in the proper manner, regardless of its original format.
- Pydub: Pydub makes it easy to manipulate audio files: trimming, concatenation, format conversion, and more. Most of these functionalities could turn out to be significant in preparing the audio data just right before feeding it into the transcription model. For instance, trimming allows its users to choose the precise segments from the audio that need transcription, saving both time and resources.
- yt-dlp: The library helps extract audio from YouTube videos. It supports downloading audio streams from YouTube URLs, which in turn can be used to transcribe audio content from online video. The power to handle YouTube URLs extends the application's utility to various uses, such as transcribing an educational video or lecture.

Model Loading and Initialization

The app loads and caches several state-of-the-art models from the Hugging Face library for transcription, punctuation, and summarization and uses Streamlit's caching mechanism to maximize performance while minimizing unnecessary load times.

Speech-to-Text Models

1. Wav2Vec2

Wav2Vec2 is the most recent, leading model for speech recognition developed by Facebook AI using the power of self-supervised learning to achieve outstanding performance. In place of most traditional supervised learning models that rely heavily on labeled datasets, Wav2Vec2 at first pre-trains lots of unlabelled audio data. This pre-training process involves masking acoustic feature representations of the input audio signal and then training the model to recover the masked features. With this, Wav2Vec2 can also learn robust, meaningful speech representations even in a small number of labeled data. Wav2Vec2 is composed of two main components: a convolutional feature encoder and a transformer network. The raw audio waveforms are then fed into the convolutional feature encoder to be converted into latent speech representations. The presentation captures essential features of the audio, and again, these features are run through a transformer network.

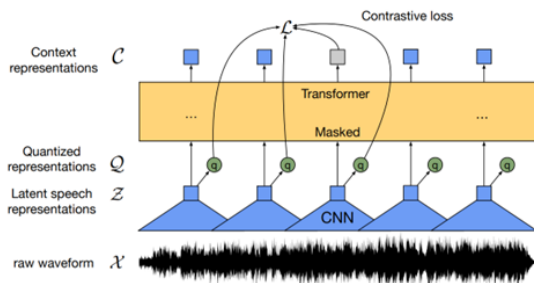


Figure 1: Architecture of the Wav2Vec2 model, source: Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. arXiv preprint arXiv:2006.11477.[13]

This is a prominent model for sequencing and probabilistically predicting text sequences derived from the earlier latent representations. This design can handle

the complications associated with human speech in terms of various accents and background noises. Fine-tuning labeled speech data further improves model accuracy, achieving high effectiveness in real applications. Therefore, due to its ability to work flawlessly in noisy environments and be robust to various accents and speaking styles, Wav2Vec2 is particularly applicable in academic cases when the quality of audio varies. Its generalization capability across different conditions gives many benefits, ensuring constant performance and reliability in various use cases.

2. HuBERT (Hidden-Unit BERT)

HuBERT (Hidden-Unit BERT) is yet another advanced model developed by Facebook AI to enhance speech recognition. Just like BERT for the natural language processing task, this model relies on masked prediction objectives for training; in fact, there are two stages in its training. Abstractedly, HuBERT first clusters the speech frames into discrete representations; in another sense, it converts continuous speech features into a more practicable form for treatment and analysis. Put a different way; these are the discrete units that are kind of pseudo-labels to capture the most essential characteristics of the audio signal. The representations are then used by the model in an autoregressive manner for the second stage to generate the targets of masked prediction. This is related to the masked language modeling task in BERT, where parts of the input data are masked, and the model is trained on how to predict those missing parts using the context provided by the unmasked parts.

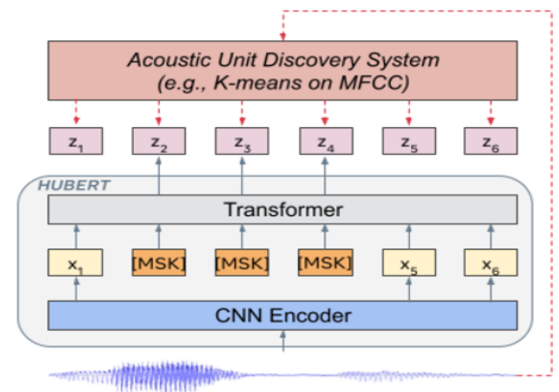


Figure 2: Architecture of the HuBERT model. Source: Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhota, K., Salakhutdinov, R., & Mohamed, A. (2021). HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. arXiv preprint arXiv:2106.07447.[14]

It enables the learning of powerful and contextually rich speech features, thus making it applicable to a wide variety of speech patterns and acoustic environments. Thanks to pre-training on large amounts of unsupervised audio data, HuBERT achieves strong generalization capabilities and yields high performance across domains and speakers. This robustness guarantees the transcript quality under different sources and conditions of the audio signal; therefore, HuBERT is most suitable for academic settings, where there will be variations in audio quality concerning accents and speaking styles. We used HuBERT in our project for its capability to learn from vast amounts of data without the same having to be labeled significantly. Because of this, the transcriptions are of high accuracy, thus making the model capable of handling a diverse audio input, which is common in an academic environment with lectures and discussions with diverse accents and background noise. HuBERT demonstrated excellent generalization across a vast variety of conditions. This property is instrumental for the performance of our speech-to-text application to be reliable and accurate in its transcriptions for academic note taking and accessibility.

Automatic Punctuation Model

1. T5 (Text-to-Text Transfer Transformer)

The model is the first of its kind in natural language processing: the T5, or Text-to-Text Transfer Transformer, developed by Google Research, which treats all tasks as text-to-text problems, unifying the disparate NLP tasks under a single task. In this paper, we use the version "flexed/t5-small-wav2vec2-grammar-fixer," useful for automatic punctuation correction. Structural makeup consists of the architecture of the transformer model, with encoder-decoder layers that process the input text to generate the corresponding output text.

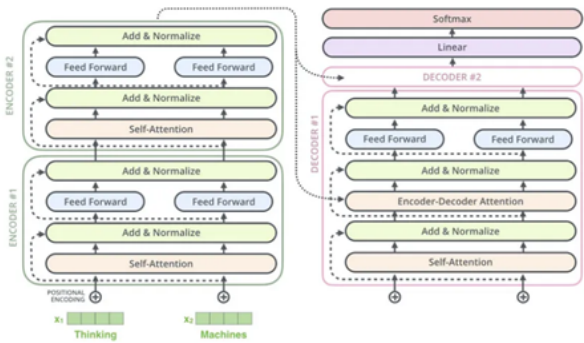


Figure 3: Architecture of the T5 model. Source: Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv preprint arXiv:1910.10683.[15]

In the area of correction, for instance, T5 is fine-tuned to take unpunctuated text as input and then output the text with appropriate punctuation. The encoder takes care of the input sequence and captures its context and structure, and the decoder places appropriate punctuation marks within the sequence based on the learned context. Such fine-tuning allows T5 to deal with quite tricky language constructions and thus hold a high level of accuracy in correcting punctuation. Context and language nuances add power to the model, and with such clear readability and coherence in the transcribed text, the transcripts can be made more valuable and accessible. This makes it very important, especially in an academic scenario, because the proper placement of punctuation and the clarity of text lead to effective communication and understanding. T5 modeling is used to ensure the quality and readability of the generated transcripts of our speech-to-text application for improved comprehensibility and usability in various educational and professional contexts.

Summarization Model

1. T5 Summarizer

T5, a product of Google Research, can also do summarization; with its powerful text-to-text feature, it can generate a brief and clear summary from an extended transcript. This boils down to a similar process in other text-to-text tasks: the full transcript will be fed to the model as it would any other input, turning out the summary retaining just the most critical information. This architecture consists of an encoder-decoder framework where the input text first passes through the encoder to grasp the gist and the essential points contained in the input text; then, during decoding, it

generates a summary by focusing on essential sections. The model uses advanced decoding methods like beam search to enhance the coherence and informativity of the summaries, making them concise but also meaningful and well-structured. The T5 summarizer's ability to distill essential information from long audio recordings makes it particularly beneficial for academic settings, such as lectures and meetings, where users often need to quickly grasp the main points without going through entire transcripts. This feature significantly enhances the usability of the speech-to-text application, allowing users to get quick overviews and efficiently review content, thereby saving time and improving comprehension. The model performs robustly with complex language structures and delivers high-quality summaries; this further underlines its utility in various educational and professional contexts.

User Interface and Configuration

The application user interface is designed using Streamlit, providing a dynamic and interactive web user experience that is very easy to use and access. The main features of the interface shall include:

File Upload: Users can easily upload audio files in MP3, MP4, and WAV formats for transcription. Designed in such a way that it can take any format of files to be compatible with almost all kinds of audio sources, the transcriptions are brought back to the user very fast and done precisely on time. Files are uploaded and processed fast, providing users with accurate and timely transcriptions.

YouTube URL Input Users can input any YouTube video URL directly in the application. This feature extracts audio content from the said YouTube videos and transcribes the content, making it possible for users to receive transcriptions of online video content without downloading or converting files manually. A handy feature to transcribe educational videos, webinars, or any other online content.

Real-Time Audio Recording: This app tends to help record real-time audio directly from the user's microphone through the streamlit_webrtc module. This allows one to record and transcribe audio on the fly, making it an excellent tool for capturing spontaneous conversations, lectures, or thoughts and observations. This way, the application would ensure that it facilitates

real-time experience not only in matters related to academics but also in business meetings.

Transcription Customizable Options: With the interface, users are given various options for customizing their transcription settings. Users can opt to include punctuation, generate summaries, distinguish between speakers, or produce subtitle files in .srt format. The said options are shown in a user-friendly form for simple configurations by the user based on his requirements.

Dynamic Display of Results: The application brings the results in an interactive format after transcription. In this view, the user can see the entire transcript and listen to the audio, as well as scroll to different parts of the transcript by the use of time stamps. The interface also contains transcript download options as both text and SRT files.

Speaker Diarization: The application includes a speaker diarization feature that distinguishes among multiple speakers in an audio file by naming the speaker transcribed. This is quite helpful for meetings, interviews, and any other instance where multiple speakers are involved, and a clean, clear transcript is desired.

User Authentication for Advanced Features: The application currently uses user access to unlock features like speaker diarization, for which it holds a Hugging Face token. This way, the application can safely use powerful models and APIs to serve you an experience like no other.

By installing these features, the app turns into an absolute and universal instrument for speech-to-text conversion, helping in many applications—academic, professional, and personal. The use of Streamlit with this will ensure that the interface remains intuitive and responsive; at the same time, the application becomes accessible to users with varying technical expertise.

Transcription Process

Transcribing in our application is an extensive process that goes through several critical stages in general, which makes the whole preceding process accurate and usable. The uploaded or recorded audio is initially loaded and preprocessed using Librosa and Pydub; the audio is resampled to 16 kHz to make it consistent and precise

according to the requirements for the transcription models. Users can tailor the transcription process user-friendly, varying parameters like start and stop times, punctuation, summarization, diarization, and other aspects using sliders and checkboxes. Now, the preprocessed audio is fed to the selected models. The Wav2Vec2 or HuBERT model converts raw audio waveforms into textual sequences using transformer architectures, predicting the most likely sequence of words from the features of the audio. If punctuation is enabled, the T5 model will add punctuation to the transcribed text using the context to make it more readable. Summarization with the T5 model refers to producing an abridged summary, identifying main points, and creating a coherent summary that tells the essentials of a longer document in its complete text. If diarization is activated, the Pyannote. audio pipeline segments the audio and labels different speakers by detecting speech segments, extracting speaker embeddings, and clustering the embeddings to identify unique speakers. Finally, the results are post-processed for clarity and coherence: the diarized segments include labels with the speaker's name and timestamps if needed. This might also be the step in which transcribed segments are concatenated into a final readable text format and could be completed with noise reduction and normalization to enhance transcription quality. The transcript is then displayed in the interface, where users can download it as a TXT or SRT file, rename detected speakers, and see summaries when the summarization option has been turned on so that the output caters to the versatile needs of academic users.

Real-Time Processing with Streamlit

The application was built on the Streamlit framework, which made real-time updating and seamless interaction by the user with an intuitive interactive interface. Several types of inputs are supported by this framework: file uploads, inputs to YouTube URLs, and direct recording by a microphone.

Streamlit enhances its performance using the caching mechanism: it stores loaded models, thus reducing the necessity for reloading them at every transcription request. This leads to responsiveness and resource effectiveness. The application provides real-time feedback and dynamic visualizations whereby users can see how transcriptions are progressing, listen to audio playback, and move through transcripts using timestamps—all within the application.

Integration of streamlit_webrtc supports in-browser audio recording directly and allows users to record and transcribe their audio in real-time. This feature suits live scenarios well: for example, with lessons or meetings. The transcribed text can be downloaded in TXT and SRT formats and exported to academic research, meeting minutes, or multimedia projects.

Streamlit ensures the output is accessible and user-friendly, even when developed by individuals with limited technical expertise. It is a perfect framework to build robust but at the same time responsive applications for speech-to-text.

All these advanced models and tools are going to be bound together using the Speech-to-Text Note-Taking Application, packaged in a helpful manner for the end user to actually convert advanced audio into text. The application encompasses state-of-the-art punctuation, speech recognition, and diarization models, thus ensuring high accuracy and usability that deems it proper for academic and professional use. Where effective preprocessing pipelines are combined with modern deep learning techniques, there is an assurance that the transcriptions will be both accurate and relevant in context, adding value to the user. The methodology section provides all applied technical details and processes included in the development of this application, hence proving it to be really well equipped to handle different audio inputs and produce very high-quality transcriptions.

IV. Results

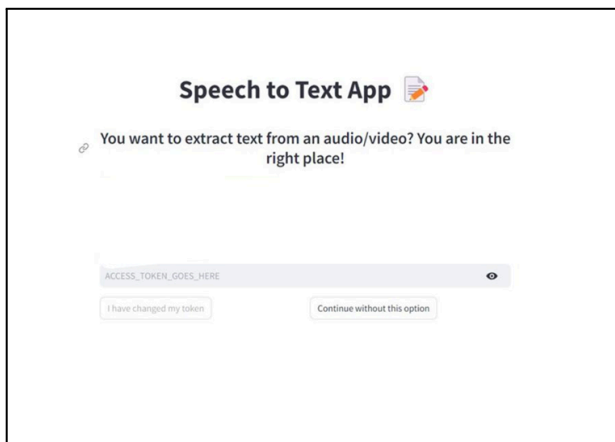
Transcription Performance :

Testing on the application was made using qualitative criteria in terms of transcription across various sources of audio feed, including college lectures, business and technical conferences, and video clips. Namely, both the Wav2Vec2 and HuBERT models proved efficient in handling different accents and background noise, coming up with accurate transcriptions of the content.

- **Wav2Vec2:** In this model, it managed to perform especially well in areas where the audio quality is quite stable and consistent, which means that the ability of the system to recognize the patterns of human speech as well as the nuances of the voices being captured is significantly enhanced. This held much usefulness especially when used to transcribe academic lectures, whereby every detail is

important and every word spoken needs to be transcribed to the last word. The speaker-independent performance of the model was established using different speakers and different manner of speaking and it was evident that the model's performance is robust for clear and structured records. Wav2Vec2 was also able to perform quite well at a high transcription level while being subjected to moderate noise, which demonstrates the model's effectiveness in more natural but structured settings.

- **HuBERT** : HuBERT demonstrated reasonable stability during the analysis of different contexts, proving its flexibility towards the audio characteristics and speakers' mannerisms. In contrast to what was observed with Wav2Vec2, HuBERT proved to be much less affected by noisy and unstable conditions that are typical of natural or professional conference calls with several speakers, interruptions, and background distractions. Due to the use of clustering of the speech frames in the initial stage, which is followed by the masked prediction model, the model was indeed able to learn the most vital characteristics of speech patterns. This made HuBERT especially useful for transcribing speech when accents, pauses, and noise levels vary and do not follow any set strict pattern.



Punctuation Correction :

The T5 (Text-to-Text Transfer Transformer) model was used to address the issue of automatic punctuation. The model correctly marked correct punctuation works, which helped to make the transcriptions more compliant and clearer. It is important that such transcriptions are

made more accessible to the lay-people and easier to understand in settings such as academic fields.

Summarization Quality :

In the summarization task, T5 model provided accurate and pertinent summaries that were derived from time consuming transcriptions. This capability is most valuable for academic videos or voice notes where a user requires a brief summary of vast amounts of content. The summaries contained the main ideas while excluding subsidiary information; therefore, they were useful as a method of a quick content analysis.

Speaker Diarization :

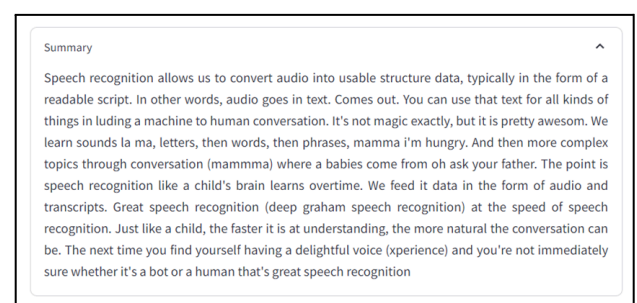
The speaker diarization task, performed using the Pyannote audio pipeline, effectively distinguished between multiple speakers in audio files. It has been particularly designed for conferences and interviews where it is crucial to identify the speakers so that the text reflects the conversation's structure.

Performance with YouTube Videos :

It was also applied very well when handling YouTube videos as identified in the earlier premises of the application. Another possibility in front of the users was the input of the URLs of the videos on YouTube and the application would convert and transcribe the audio content perfectly. This feature turned out to be really helpful for transcribing educational videos and any useful content available on the Internet, which was highly needed for people who prefer to watch videos online rather than downloading them and then using converters to transcribe the content.

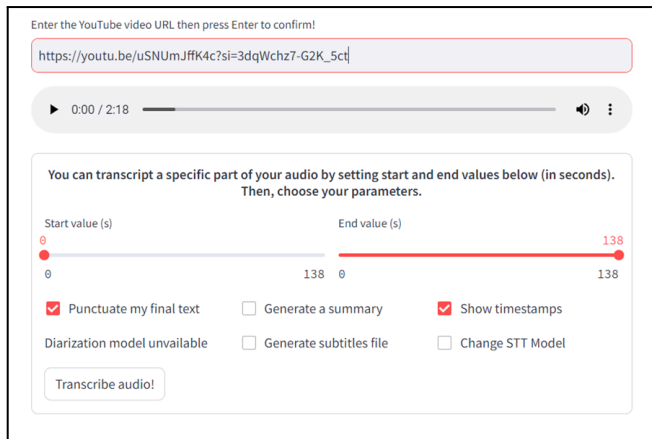
Overall Performance :

By using the advanced pre-trained models, the Speech-to-Text Note-Taking Application is highly usable together with having high accuracy. It is equally useful for academic analysis of YouTube videos, professional conference calls and general educational usage due to the flexibility of the application in dealing with different types of audios and producing accurate transcriptions. The results of the qualitative analysis correlate with the objective evaluation and demonstrate the efficiency and reliability of the Wav2Vec2 and HuBERT models in transcribing the speech, the T5 model in correcting the punctuation and summarizing the texts, and the Pyannote audio pipeline in distinguishing between the speakers.



Qualitative Examples:

- **Transcription Example:** An audio clip from an academic lecture was synthesized with a good record of the lecturer's speech and recording quality even with background noises and differences in lecture pace.
- **Punctuation Correction Example:** After correcting an excerpt from a transcribed lecture, its punctuation was more appropriate which enhanced understanding of the document.
- **Summarization Example:** A full text of the lecture was substantially summarized, when all the major points of the lecture notes were kept and the overall usefulness of the content for quick revision was increased.
- **YouTube Video Example:** An English educational video from YouTube was transcribed effectively well so as to reproduce the spoken content unabated and in a manner which could be easily reviewed or used for further studies.



The screenshot displays the application's user interface. At the top, there is a text input field for a YouTube URL, with a placeholder text "Enter the YouTube video URL then press Enter to confirm!". Below this is a video player showing a video with a duration of 0:00 / 2:18. Under the video player, there is a section for transcription settings. It includes a text box with the instruction: "You can transcribe a specific part of your audio by setting start and end values below (in seconds). Then, choose your parameters." Below this, there are two sliders for "Start value (s)" and "End value (s)". The "Start value (s)" slider is set to 0, and the "End value (s)" slider is set to 138. There are also checkboxes for "Punctuate my final text" (checked), "Generate a summary" (unchecked), "Show timestamps" (checked), "Diarization model unavailable" (checked), "Generate subtitles file" (unchecked), and "Change STT Model" (unchecked). At the bottom, there is a button labeled "Transcribe audio!".

V. Discussion

Implications:

Our STT Note-Taking Application, in practice, shows how advanced models of speech recognition could work together in a friendly interface to enhance productivity and accessibility levels in many contexts drastically. With the usage of the latest models such as Wav2Vec2, HuBERT, and T5, this app has a high performance in the areas of transcription, punctuation correction, and summarization. This feature is ideal and works best in the academic setup where effective learning and knowledge depend on an accurate transmission of lectures and studying material.

The application is further enhanced with features such as real-time transcription, speaker diarization, and YouTube video transcription, which span its usability in other domains. It is a versatile tool that would prove handy for students, researchers, or professionals who require a reliable, effective way of converting spoken content into text format for further analysis and reference.

Strengths:

1- Model Integration and Performance: The use of robust models, for example, Wav2Vec2 and HuBERT, brings about high accuracies of transcription even in noisy environments and audios of various accents. The T5 model enhances the readability aspect of the transcriptions with relevant punctuation and a summary appended.

2- Easy-to-Use Interface: Streamlit is used for the user interface to make the developed application easy to use, even for those not very technically advanced. The intuitive design allows easy audio file uploads, YouTube URL inputs, and real-time audio recording.

3- Versatility and Flexibility: The application supports quite several different forms of audio input and can be used with flexible transcription settings to fit a variety of use cases, from academic lectures to professional meeting notes.

4- Real-Time Processing: The real-time feature of the audio recording using streamlit_webrtc is a significant advantage for capturing live events and conversations efficiently.

Limitations:

1- Dependence on Audio Quality: Even though models like HuBERT are relatively robust against noisy conditions, the application can still be influenced by low audio quality or heavy background noise that pre-processing steps might not be able to deal with fully.

2- Computational Resources: Applying advanced deep learning models to real-time transcription and processing can consume heavy computational resources and, therefore, be inefficient and hard to run on low-power devices.

3- Speaker Diarization Accuracy: The Pyannote audio pipeline provides effective speaker diarization, but in complicated audio files with overlapping speech, it remains quite a significant challenge to differentiate multiple speakers.

4- Language and Accent Limitations: The current implementation primarily supports English, and while models like Wav2Vec2 and HuBERT can generalize across different accents, they may still encounter difficulties with less common accents or dialects that may not be well represented in the training data.

VI. Conclusion

The Speech-to-Text Note-Taking App aims to be a leading model in the educational technology field, integrating high-tech speech recognition models with a user-friendly interface. The application achieves high accuracy, speed, and contextual relevance. Using advanced models like Wav2Vec2 and HuBERT for transcription and T5 for punctuation, correction, and summarizing ensures considerable challenges are solved.

The Speech-to-Text Note-Taking application is very versatile in working with a significant number of audio input formats, starting from real-time recordings to file uploads or even YouTube videos.

The application is also capable of handling a variety of accents and background noises, requiring punctuation, and can summarize for students, staff, and professionals. This has made the qualitative evaluation of this application robust. Something that makes this kind of tool very useful is the way speaker diarization and dynamic result display work, which would be very helpful in scenarios such as group discussions and meetings.

This app satisfies not only today's demand for more effective note taking and better access to content but also lays the foundation for the further development of the technologies in the speech-to-text direction. All this with the help of Streamlit, integrated for an interactive interface for even complex AI models to be explored by people with minimal technological background.

Overall, this Application can enhance productivity and learning experiences within educational or professional environments. The capability to turn any spoken language into clear, punctuated, and summarized text introduces new ways of accessibility and engagement with the content. It, therefore, becomes an inclusive and more efficient way of dealing with spoken content, and a valuable academic tool.

VII. References

[1] Graves, A., 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

[2] Yu, D. and Deng, L., 2016. *Automatic speech recognition* (Vol. 1). Berlin: Springer.

[3] Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), pp.257-286.

[4] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. and Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6), pp.82-97.

[5] Mesquita, R.A., Araújo, V.C.D., Paes, R.A.P., Nunes, F.D. and Souza, S.C.O.M.D., 2009. Immunohistochemical analysis for CD21, CD35, Caldesmon and S100 protein on dendritic cells types in oral lymphomas. *Journal of Applied Oral Science*, 17, pp.248-253.

[6] Shadiev, R., Hwang, W.Y., Chen, N.S. and Huang, Y.M., 2014. Review of speech-to-text recognition technology for enhancing learning. *Journal of Educational Technology & Society*, 17(4), pp.65-84.

[7] Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K. and Bengio, Y., 2015. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28.

[8] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A. and Ng, A.Y., 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

[9] Marge, M., Banerjee, S. and Rudnicky, A.I., 2010, March. Using the Amazon Mechanical Turk for transcription of spoken language. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 5270-5273). IEEE.

[10] Microsoft. (2023). Microsoft Azure Speech

[11] Baevski, A., Zhou, Y., Mohamed, A. and Auli, M., 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33, pp.12449-12460.

- [12] Hsu, W.N., Bolte, B., Tsai, Y.H.H., Lakhota, K., Salakhutdinov, R. and Mohamed, A., 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, pp.3451-3460.
- [13] Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. arXiv preprint arXiv:2006.11477.
- [14] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhota, K., Salakhutdinov, R., & Mohamed, A. (2021). HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. arXiv preprint arXiv:2106.07447.
- [15] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv preprint arXiv:1910.10683.