Speech Understanding, Assignment 1

Akansha Gautam
M23CSA506
IIT Jodhpur
m23csa506@iitj.ac.in

1. Speech-to-Text Conversion task

1.1. Explain the task and its importance in the real world

Speech-to-Text Conversion, also called Automatic Speech Recognition (ASR), is the process of processing the speech signals and converting the spoken language into text format using computational models. It focuses on maintaining the linguistic context in the audio signal while transcribing it to the textual data.

The Speech-to-Text Conversion plays a critical role in solving the following real-world problems signifies its importance.

- It provides real-team captions for videos and meetings, allowing people with hearing impairments to access spoken content.
- This technique enables people to understand the audio content better if there's some foreign accent present in the audio.
- It is heavily used by customer care agents in automating call transcriptions. These transcriptions can then later be used for providing better care support to the customers.
- Speech-to-text conversion with machine translation facilitates people with different geographies to communicate better.

1.2. Analyze the strengths and limitations of state-of-the-art models or tools in terms of the methods or models available

I and Anchit Mulye (M23CSA507) have worked together on analyzing the strengths and limitations of the state-of-the-art models.

1.2.1. Nova-2

Nova-2 is the powerful speech-to-text (STT) model powered by Deepgram available in the English language. Nova-2 uses a transformer-based architecture which helps boost accuracy for both pre-recorded and streaming transcription of entities, punctuation, and capitalization. Nova-2 is trained on a dataset that is curated from nearly 6 million resources and incorporates an extensive library of high-quality human transcriptions. The nova-2 model can be used for other tasks as well such as speaker diarization, smart formatting, filter words support, and summarization capability.

Strengths of Nova-2:

- Nova-2 provides an average 30% reduction in word error rate over competitors for both pre-recorded and real-time transcription.
- It is trained on a heavily curated dataset with nearly 6 million resources and incorporates an extensive library of high-quality human transcriptions.
- It responds faster.
- This model can be used for other tasks as well such as speaker diarization, smart formatting, filter words support, and summarization capability.

Limitations of Nova-2:

- Nova-2 provides speech-to-text conversion only for the English language (high-resource language).
- Nova-2 is a commercial platform and doesn't provide open-source access.

1.2.2. Whisper

Whisper is an open-source speech-to-text model, powered by OpenAI, which is trained on multilingual and multitask supervised data collected from the web. The Whisper architecture is implemented as an encoder-decoder Transformer. Given input file is split into 30-second chunks, converted into a log-Mel spectrogram, and then passed into an encoder. A decoder is then used to predict the corresponding text out of the encoded chunks of audio signal.

Strengths of Whisper:

- Whisper provides speech-to-conversion for many other regional languages.
- Whisper is an open-source model, readily available to use.
- Whisper can also perform tasks like live-streaming transcription and speaker diarization.

Limitations of Whisper:

- Upload file size in Whisper is limited to 25MB and 30 seconds in duration.

- This model cannot process URLs and callbacks.
- It is infamously prone to hallucinations, resulting in errors in the transcript.

1.3. Discuss the results in terms of the metrics used to evaluate the task, including their strengths and limitations

We used the Word Error Rate (WER) metric to evaluate the performance of Nova-2 and Whisper (?) models on the speech-to-text dataset, The LJ Speech Dataset, available on Kaggle. WER is a common metric used to evaluate the accuracy of speech-to-text (STT) models. A lower WER indicates better accuracy whereas a higher WER suggests more errors in the generated transcription. We can calculate WER using the formula:

$$WER = \frac{S + D + I}{N}$$

where,

- S denotes the number of incorrect words placed in a position of correct ones
- D denotes the number of missing words
- I denotes the number of inserted words
- N denotes the total number of words in the generated transcript

We calculate WER for both the models, Nova-2 and Whisper on the same dataset, The LJ Speech Dataset available on Kaggle. This dataset comprises 1000 short audio clips (ranging from 1 to 10 seconds) of a single speaker reading passages from 7 non-fiction books.

	Nova-2	Whisper
Word Error Rate (WER)	15.48%	13.61%

Table 1. Word Error Rate for both Nova-2 and Whisper model on The LJ Speech Dataset available on Kaggle

Table 1 shows the Word Error Rate (WER) metric results for Whisper and Nova-2 models when given the LJ Speech Dataset. From this table, we can see that Whisper has performed better than Nova. The reason for that could be because Whisper is trained on multilingual datasets with many resources which made it robust in performance as compared to nova-2.

1.4. Suggest what are the open problems and opportunities corresponding to the problem statement

The open problems related to the Speech-to-Text(STT) Conversion task are as follows:

- **Lack of Low-Resource Languages Support**: Most of the models are trained in high-resource languages like English but they struggle with regional languages.
- **Poor transcription quality in real-world environment**: It is difficult for the STT models to extract the linguistic context present in the audio signal in noisy environments.
- **Speaker Variability & Accents**: STT Models struggle due to different accents and pitch variations.

The opportunities corresponding to the Speech-to-Text (STT) Conversion task are as follows:

- STT models can be trained on multilingual datasets so that they become aware of the other regional languages as well.
- We can optimize the current models to reduce latency by processing the audio file in smaller chunks parallelly.
- STT models lack context awareness, leading to misinterpretation of homophones. We can combine speech + video (lip reading) + text for improved accuracy.

2. Experiment with Spectrograms and Windowing Techniques

2.1. Task A

2.1.1. Implement Rectangular Windowing Technique

There is no audio-signal tapering in the Rectangular Windowing Technique which is equivalent to no windowing at all. In figure 1, we can see that the amplitude of the original audio signal and the rectangular windowed audio signal are the same.

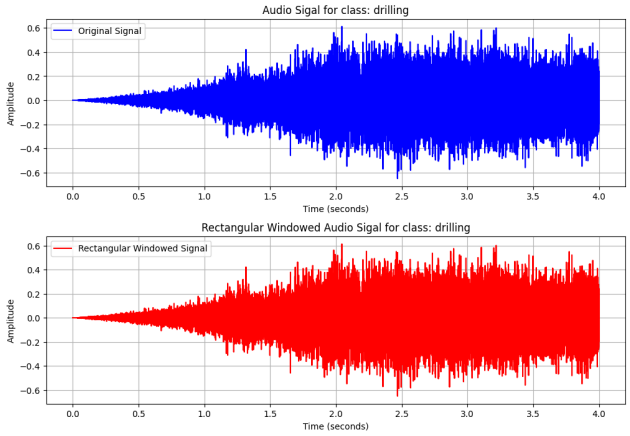

Figure 1. Comparison of Original Audio Signal and Rectangular Windowed Audio Signal for class drilling

2.1.2. Implement Hamming Windowing Technique

The Hamming Windowing Technique tapers off smoothly towards the ends, providing better frequency resolution than a rectangular window. The Hamming Window has a comparatively higher value at the edges than the Hann Window as it doesn't quite reach zero, leading to a slight presence of discontinuity in the signal. We can use the following for-

mula to calculate the Hamming window where N is the full window size:

$$w[n] = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right)$$

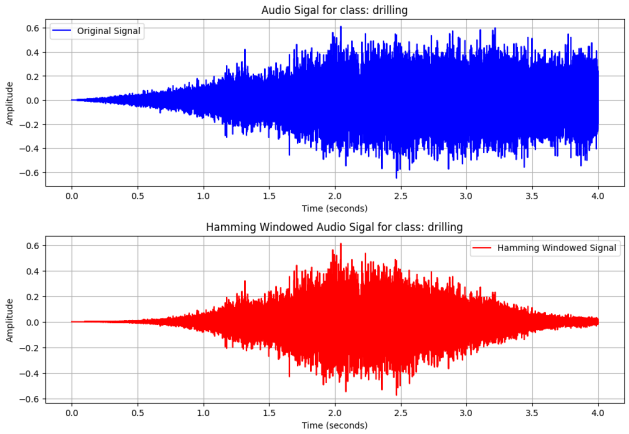

Figure 2. Comparison of Original Audio Signal and Hamming Windowed Audio Signal for class drilling

Figure 2 compares the original audio signal and the Hamming windowed audio signal when choosing a drilling class from the Urban Sound 8K dataset.

2.1.3. Implement Hann Windowing Technique

Hann Windowing Technique is a popular windowing technique used to perform Hann Smoothing. This technique reduces spectral leakage and is often more suitable for smooth transition signals. Hann Windowing touches zero at both ends, thus helping to eliminate all the discontinuity present in the audio signal. We can use the following formula to calculate the Hann window where N is the full window size:

$$w[n] = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{N-1}\right)\right] = \sin^2\left(\frac{\pi n}{N-1}\right)$$

Figure 3 compares the original audio signal and the Hann windowed audio signal when choosing a drilling class from the Urban Sound 8K dataset. In this figure, we can see that, in the Hann windowed technique, zero touches both ends.

2.1.4. Generate and Compare Spectrograms

Figure 4 compares the spectrogram generated using the Short-Time Fourier Transform (STFT) for Rectangular, Hann, and Hamming Window techniques when choosing the class *gun shot*.

- **Rectangular Window Technique**: The spectrogram created with a rectangular window has sharp frequency transitions with many vertical lines. But it has high spectral leakage as it doesn't taper the signal at the edges so energy spreads across frequencies.

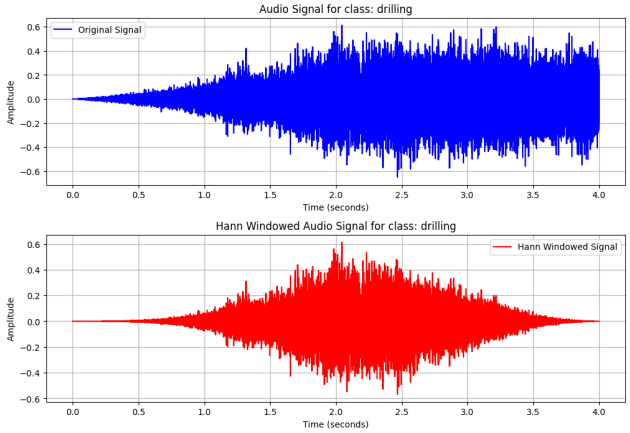

Figure 3. Comparison of Original Audio Signal and Hann Windowed Audio Signal for class drilling

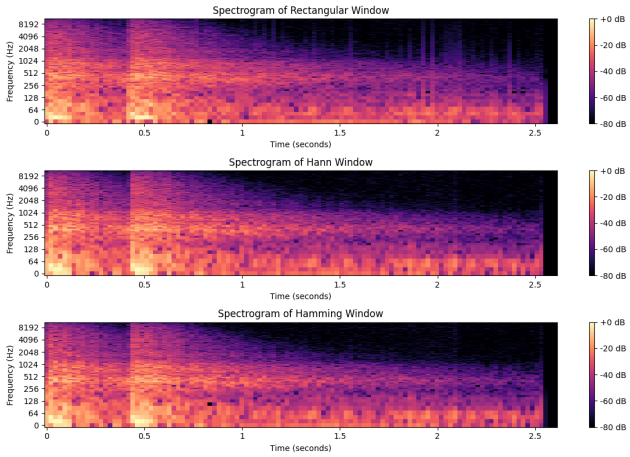

Figure 4. Comparison of Original Audio Signal and Hann Windowed Audio Signal for class drilling

- **Hann Window Technique**: On the other hand, the Hann window has much smoother transitions as it tapers the signal edges and concentrates energy in the main lobe.
- **Hamming Window Technique**: Similarly the Hamming window reduces spectral leakage and maintains amplitude consistency by providing a balance between tapering and energy concentration.

In terms of correctness, the rectangular window is not ideal for frequency analysis due to its inability to reduce spectral leakage. In contrast, the Hann and Hamming windows correctly perform the tapering, thereby improving frequency resolution. Among these, the Hann window is often recommended for audio analysis due to its superior balance between resolution and leakage reduction.

2.1.5. Train a simple Classifier and Evaluate the performance results

I have trained a simple classifier (SVM) using features extracted from the Short-Time Fourier Transform (STFT) Spectrograms. Here are the steps performed for the same:

- Extract features from Short-Time Fourier Transform for each windowing technique
- Encode the label class using LabelEncoder
- Split the train and test set with an 80/20 split
- Normalise the extracted features using StandardScaler
- Use Principal Component Analysis (PCA) to perform the dimensionality reduction on all the normalized features set
- Train SVM classifier on the reduced features set and encoded label
- Make predictions on the trained model using the test features set
- Evaluate model prediction using metrics such as *precision*, *recall*, *f1-score* and *accuracy*

	Rectangular Window	Hann Window	Hamming Window
Accuracy Score	0.58	0.60	0.61

Table 2. Accuracy score using SVM for Rectangular, Hann, and Hamming Window Technique

Table 2 shows results from the SVM classification task revealing the impact of different window functions on the model's performance.

- The Rectangular Window had the lowest accuracy score of 0.58. This is because it doesn't taper the signal at the edges, which leads to spectral leakage and less effective feature extraction.
- The Hann Window performed a bit better, achieving an accuracy of 0.60. This improvement is due to its ability to smoothly taper the signal at the edges, reducing spectral leakage and providing cleaner features.
- The Hamming Window came out on top with an accuracy of 0.61. It strikes a good balance between tapering the signal and retaining its energy, making it the most effective choice for this classification task.

2.2. Task B

Figure 5 shows the Spectrogram of English EDM song where darker regions represent higher energy and lighter regions represent lower energy. In this spectrogram, we can the presence of consistent and broad dark horizontal line near the frequency range of 16384 hertz with two broad vertical lines at the start and end of the music which denotes the presence of high-frequency and high-amplitude signal waves. There is a similar pattern present across this music where there's a high-amplitude signal wave surrounded by multiple low-amplitude signal waves that resonate with the EDM music. Over here, the Hann windowing technique is used to reduce spectral leakage by providing a clear representation of the complex audio signal.

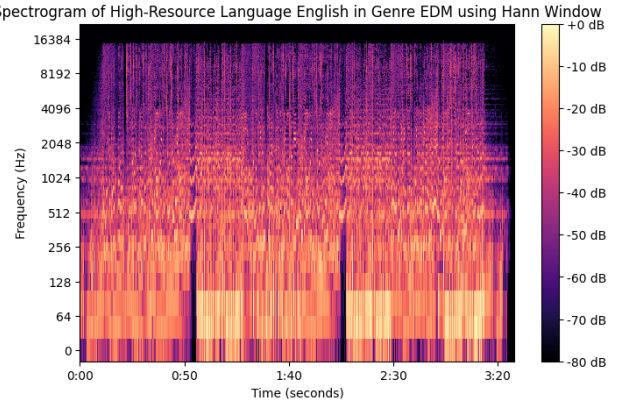

Figure 5. Spectrogram of High-Resource Language English in Genre EDM using Hann Window

Figure 6 depicts the Spectrogram of English POP music. The presence of low-amplitude low-frequency signal waves is much higher in this music as compared to the English EDM which makes sense as in EDM music, a lot of heavy musical instruments are used.

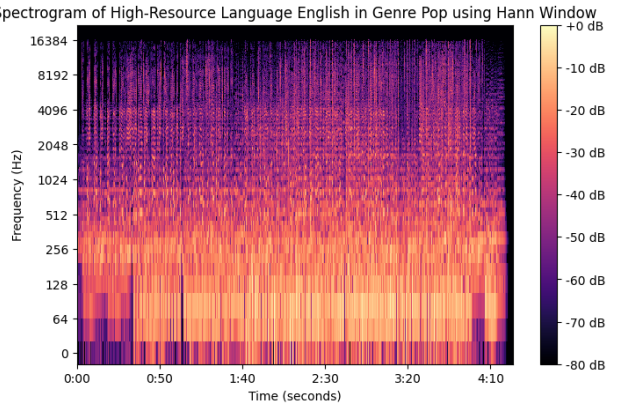

Figure 6. Spectrogram of High-Resource Language English in Genre POP using Hann Window

Figure 7 shows the spectrogram of unique characteristics of Rajasthani folk music. In this spectrogram, the darker region highlights the louder sounds while the lighter region highlights the softer sounds. The x-axis focuses on the time in seconds while the y-axis focuses on the frequency in hertz and color intensity focuses on the sound intensity. There are multiple vertical lines present with darker gradients indicating the presence of repeated, low-frequency beats with high sound intensity. We can also see that a darker horizontal line present near frequency 8192 hertz shows a repeated higher frequency being consistent throughout the music.

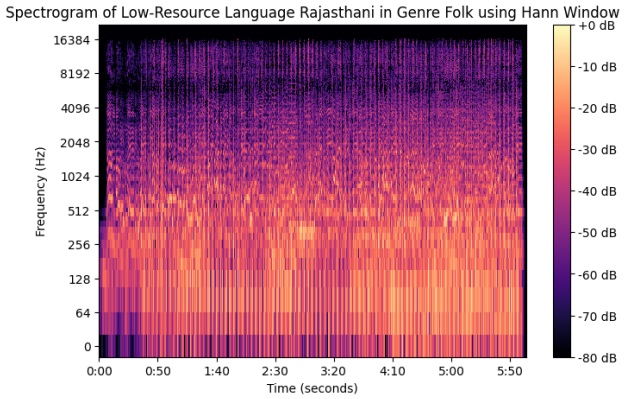

Figure 7. Spectrogram of Low-Resource Language Rajasthani in Genre FOLK using Hann Window

Figure 8 shows a spectrogram of a Punjabi rap song in the rap genre using a Hann window. From the spectrogram it is clear that the Hann window effectively reduces spectral leakage, resulting in clear frequency bands. In the spectrogram, the lower frequencies dominate, showing the bass-heavy nature typical of rap music. The mid-range frequencies show moderate activity, probably representing the singer's sound and rhythm. The high frequency with vertical lines shows sharp and short sounds. This distribution of amplitude and frequency in this Spectrogram demonstrates that this song is base-heavy with sharp notes.

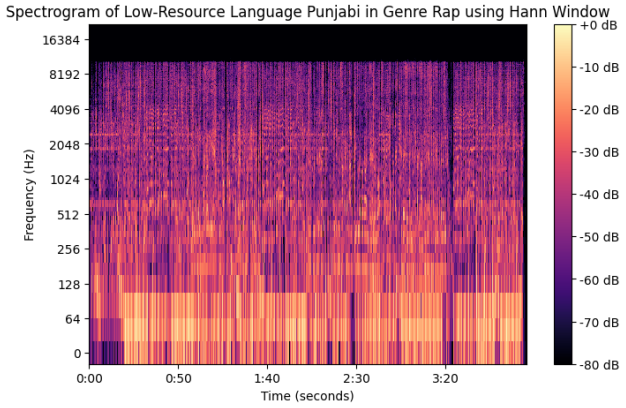

Figure 8. Spectrogram of Low-Resource Language Punjabi in Genre RAP using Hann Window

3. Additional Details

- Github Repository Link
- Question 1 was done in a group with Anchit Mulye (M23CSA507)
- Question 2 was done independently