Speech Processing Assignment - 3

GitHub Link : https://github.com/Toji339/SpeechProcessing

Dataset Description :

- This LJ Speech dataset has been taken from the Kaggle.
- It contains 13,100 short audio clips of a single speaker
- These clips vary in length from 1 to 10 sec and have a total length of around 24 hours.

Objective :

- - Compute and visualize the Fourier Transform of a speech signal .
  - Compute and visualize the STFT of a speech signal to analyze its time-varying frequency content .
  - analyze and compare the energy distribution of vowels and consonants in speech signals .

Experiment 1

Code :

Fourier Transform for Speech Signal Analysis

```python
import librosa
import librosa.display
import numpy as np
import matplotlib.pyplot as plt

file_path = "/content/LJ001-0031.wav"
signal, sr = librosa.load(file_path, sr=None)

# Experiment 1(A): Fourier Transform


fft_spectrum = np.fft.fft(signal) # computing the fast fourier transform
freqs = np.fft.fftfreq(len(fft_spectrum), 1/sr)

magnitude = np.abs(fft_spectrum) # computing the magnitude

plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)
```

```python
plt.plot(np.linspace(0, len(signal) / sr, len(signal)), signal)
plt.title('Time-Domain Signal')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')

plt.subplot(2, 1, 2)
plt.plot(freqs[:len(freqs)//2], magnitude[:len(magnitude)//2]) # positive frequencies
plt.title('Frequency-Domain Representation (FFT)')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')

plt.tight_layout()
plt.show()
```

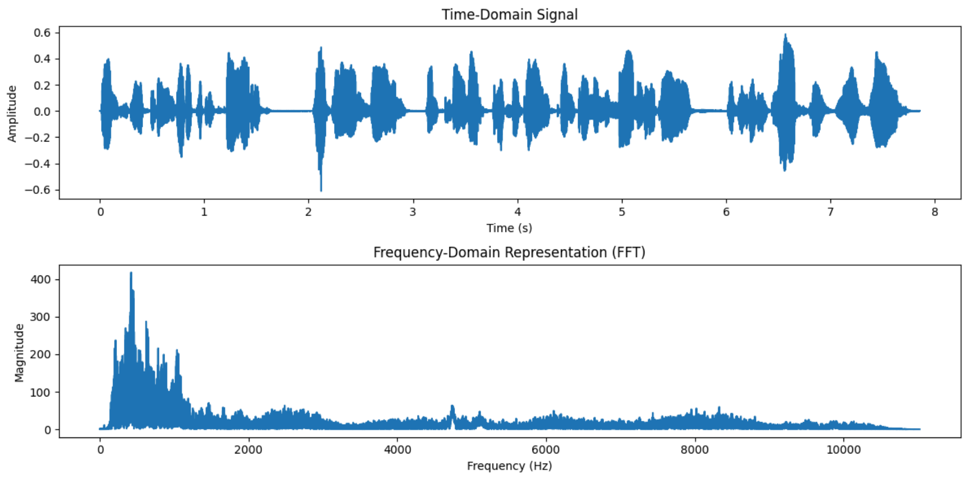

Figure 1: FFT.

Observations on Fast Fourier Transform

Key Observations:

-
  - There is a high concentration of frequency components in the low-frequency range (0-2000 Hz).
  - Peaks in the lower frequencies indicate that most of the speech signal energy is concentrated in these regions.
  - The presence of smaller peaks in higher frequency ranges (above 3000 Hz) suggests the presence of fricatives(e.g., /s/, /sh/ sounds).
  - Fricative lies in the range of 2000Hz to 8000Hz.

Short-Time Fourier Transform (STFT)

Python

```python
#  Experiment 1(B): Short-Time Fourier Transform (STFT)

# Computing the  STFT
stft_result = librosa.stft(signal, n_fft=1024, hop_length=512)
stft_magnitude = np.abs(stft_result)

# Converting into decibels
stft_db = librosa.amplitude_to_db(stft_magnitude, ref=np.max)

# Plot the spectrogram
plt.figure(figsize=(10, 5))
librosa.display.specshow(stft_db, sr=sr, hop_length=512, x_axis='time',
y_axis='log')
plt.colorbar(label='Amplitude (dB)')
plt.title('Spectrogram (STFT)')
plt.show()
```

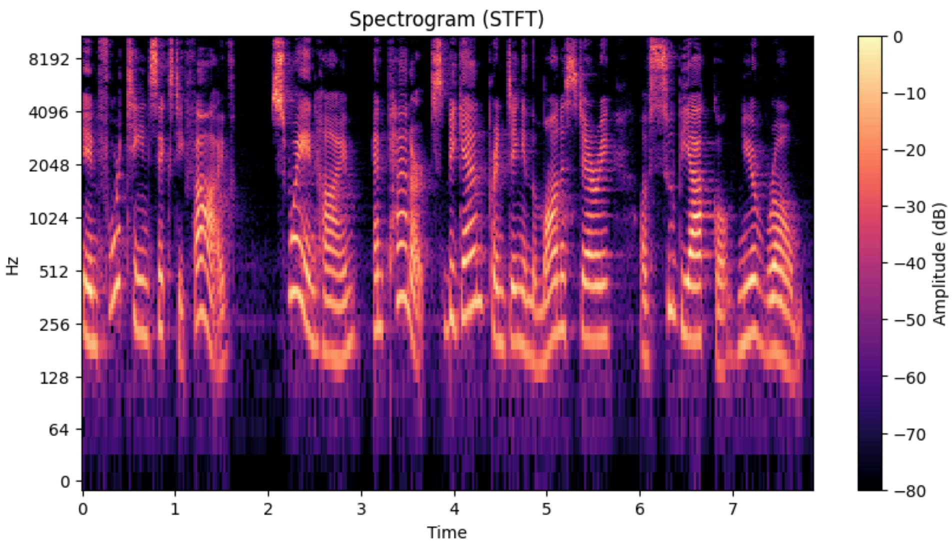

Figure 2: Short Time Fourier Transform .

Observations on Short Time Fourier Transform

Key Observations:

- - The x-axis represents time, the y-axis represents frequency, and the color intensity represents amplitude in decibels (dB).
  - Bright regions indicate high-energy frequency components at specific time intervals.
  - Lower frequencies remain active throughout the speech, confirming that speech primarily consists of lower-frequency components.
  - The high-frequency regions (above 4000 Hz) are less prominent but still present, which corresponds to unvoiced consonants and fricatives.

- Unlike the FFT, the STFT provides both frequency and time information, making it ideal for analyzing speech signals.

Experiment 2

Energy Distribution of Vowels and Consonants

```python


# Extracting a phonemene first 0.5 seconds)
start_sample = 0
end_sample = int(0.5 * sr)
phoneme_segment = signal[start_sample:end_sample]


plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)
time_axis = np.linspace(0, len(phoneme_segment) / sr, len(phoneme_segment))
plt.plot(time_axis, phoneme_segment)
plt.title('Phoneme Segment Waveform')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')


stft_result = librosa.stft(phoneme_segment, n_fft=1024, hop_length=512)
stft_magnitude = np.abs(stft_result)
stft_db = librosa.amplitude_to_db(stft_magnitude, ref=np.max)

plt.subplot(2, 1, 2)
librosa.display.specshow(stft_db, sr=sr, hop_length=512, x_axis='time', y_axis='log')
plt.colorbar(label='Amplitude (dB)')
plt.title('Spectrogram of Phoneme Segment')
plt.show()

# Computing Energy in Different Frequency Bands

# Computing STFT for full signal
stft_result_full = librosa.stft(signal, n_fft=1024, hop_length=512)
stft_magnitude_full = np.abs(stft_result_full)


```

```python
freq_bins = librosa.fft_frequencies(sr=sr, n_fft=1024)


low_freq_indices = np.where((freq_bins >= 300) & (freq_bins <= 3000))[0]
high_freq_indices = np.where((freq_bins >= 4000) & (freq_bins <= 8000))[0]


low_freq_energy = np.sum(stft_magnitude_full[low_freq_indices, :]**2)
high_freq_energy = np.sum(stft_magnitude_full[high_freq_indices, :]**2)

# Computing energy ratio between vowels (low-freq) and fricatives (high-freq)
energy_ratio = low_freq_energy / high_freq_energy if high_freq_energy != 0 else np.inf

print(f"Energy in low-frequency (vowels): {low_freq_energy:.2f}")
print(f"Energy in high-frequency (fricatives): {high_freq_energy:.2f}")
print(f"Energy ratio (vowels to consonants): {energy_ratio:.2f}")
```

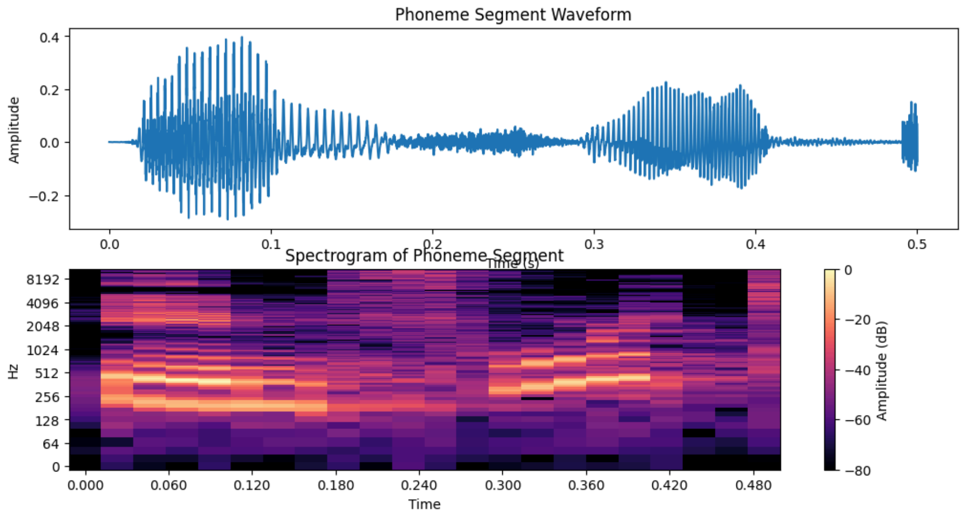

Figure 3: Experiment 2 Visualization .

Key Observations on Experiment 2:

- Vowel sounds (e.g., /a/, /e/, /i/, /o/, /u/) have high energy in low-frequency bands (300–3000 Hz).This is because vowels are produced with an open vocal tract, allowing more resonance in the lower frequency range.
- Consonants, especially fricatives (/s/, /sh/, /f/), show higher energy in the 4000–8000 Hz range.This is due to turbulent airflow in the vocal tract, which generates high-frequency noise.
- The energy in vowels (300–3000 Hz) is significantly higher and more sustained than in consonants..
- The energy in consonants (4000–8000 Hz) is more spread out and discontinuous.
- The vowel-to-consonant energy ratio is typically greater than 1, meaning vowels dominate speech energy