23011101119 – S K KARISMA

23011101128 – SAMYUKTHA R G

DATA MINING

CIA - I

SPEECH PROCESSING

1. DOCUMENTATION ON SPEECH

Nature of Continuous Speech

Continuous speech is a fluid stream of sounds, with each segment corresponding to a phone—an acoustic feature influenced by factors such as the speaker's voice and contextual variations. Phones do not exist in isolation but blend seamlessly:

- **Diphones** are formed by the transition between two consecutive phones.
- **Triphones** and **quinphones** offer a more detailed context by considering the influence of neighboring phones.

In spontaneous speech, syllables remain stable, though the number of phones can vary due to natural speech patterns. Words limit the possible combinations of phones, aiding in the accurate interpretation of spoken language.

Additionally, **fillers** like "um," "uh," or breathing sounds are naturally occurring, reflecting hesitation or the speaker's thought process.

Types of Speech Sounds

- **Voiced Sounds**: Produced by the periodic vibration of the vocal cords, as in vowels.
    - Male pitch range: 50–250 Hz
    - Female pitch range: 120–500 Hz
- **Unvoiced Sounds**: Created by turbulent airflow through a constriction, like the "s" sound in "six."
- **Plosives**: Produced when air pressure is built up and suddenly released, as in "t" or "p."
- **Mixed Sounds**: Contain both voiced and unvoiced components, like the "z" sound in "is."

Speech Signal Representation

- **Time-domain representation**: The raw speech signal as an audio waveform, where the signal amplitude varies over time.

- **Frequency-domain representation**: The signal is decomposed into frequency components using the Fast Fourier Transform (FFT).
- **Spectrograms**: A 2D representation of frequency content changing over time, created using the Short-Time Fourier Transform (STFT).

Speech Features

- **Mel-frequency Cepstral Coefficients (MFCC)**: Represent the short-term power spectrum of sound, emphasizing perceptually relevant frequencies.
- **Formants**: Resonant frequencies important for vowel recognition and speaker identification.
- **Pitch**: The perceived frequency of speech, crucial for distinguishing speech from non-speech sounds.
- **Energy**: Reflects the loudness of the speech signal, useful for detecting speech pauses or background noise.

Phonemics and Phonetics

Phonemics is the study of phonemes, the smallest units of speech that carry linguistic meaning. Phonemes are essential for distinguishing words, like the difference between "pet" and "bet," where /p/ and /b/ alter the meaning. English typically has 40–45 phonemes, with each linked to specific articulatory movements. Minimal pairs, like “pet” vs. “bet,” help identify phonemes.

- **Allophones** are variations of a phoneme that do not change the meaning, such as the difference between the aspirated [ph] in "pit" and unaspirated [p] in "spit."

MPEG-7 Audio Features

The **MPEG-7** standard classifies audio and speech signals, including features for analyzing and categorizing sounds.

$$\boldsymbol{Centroid} = \sum_{n=0}^{N-1} \boldsymbol{f(n)x(n)} \,/ \sum_{n=0}^{N-1} \boldsymbol{x(n)}$$

- **Audio Spectrum Centroid (ASC)**: Measures the logarithmic frequency scale centered around 1 kHz, representing the frequency distribution in the power spectrum. It helps determine whether the sound is perceived as bright or dull.
- **Audio Spectrum Spread (ASS)**: Measures the spectral distribution around the ASC, capturing how frequencies spread relative to the centroid. It helps differentiate between noise and speech by analyzing spectral content variance.

2. FUNDAMENTAL TECHNICAL COMPONENTS

2.1. Audio Acquisition

The first step in working with speech data is acquiring the audio signal. This can be done using microphones or other recording devices. The quality of the recording device and the environment in which the recording is made play a crucial role in the quality of the speech data.

2.2. Digitization

Once the speech is recorded, it needs to be converted into a digital format for processing. This involves:

- **Sampling:** The continuous audio signal is sampled at regular intervals to create a discrete signal. The sampling rate (measured in Hertz) determines the number of samples per second. A common sampling rate for speech is 16 kHz.
- **Quantization:** Each sampled value is then quantized into a finite number of levels, which is typically represented by a certain number of bits (e.g., 16-bit quantization).

2.3. Preprocessing

Before analyzing the speech data, some preprocessing steps are often necessary:

- **Normalization:** Adjusting the amplitude of the audio signal to a standard level.
- **Noise Reduction:** Filtering out background noise to improve the clarity of the speech signal.

2.4. Feature Extraction

To convert the speech signal into a numerical representation, we extract features that capture the essential characteristics of the audio. Common features include:

- **Time-Domain Features:**
  - **Zero-Crossing Rate (ZCR):** The rate at which the signal changes sign.
  - **Energy:** The sum of the squares of the signal amplitude.
- **Frequency-Domain Features:**
  - **Spectral Centroid:** The center of mass of the spectrum.
  - **Mel-Frequency Cepstral Coefficients (MFCCs):** A representation of the short-term power spectrum of sound.

2.5. Windowing and Framing

Speech signals are non-stationary, meaning their statistical properties change over time. To analyze them effectively, the signal is divided into small overlapping segments called frames (e.g., 20-40 milliseconds). Each frame is multiplied by a window function (e.g., Hamming window) to reduce edge effects.

2.6. Numerical Representation

Once the features are extracted, they are represented as numerical vectors. Each frame of the speech signal is converted into a feature vector, resulting in a sequence of vectors that represent the entire audio signal.

2.7. Analysis and Processing

With the numerical representation of the speech data, various analyses can be performed:

- **Statistical Analysis:** Calculating mean, variance, and other statistical properties of the features.
- **Pattern Recognition:** Identifying patterns or anomalies in the speech signal.

APPLICATIONS OF SPEECH PROCESSING:

1. Speech Recognition (ASR - Automatic Speech Recognition)

2. Speech Synthesis (Text-to-Speech - TTS)

3. Speaker Recognition & Verification

4. Speech Enhancement & Noise Reduction

5. Speech Emotion Recognition

6. Speech-to-Music & Voice Manipulation

7. Language Translation

8. Keyword Spotting & Censorship

APPLICATION CHOSEN:

Keyword Spotting :

Keyword spotting/keyword recognition detects a word or short phrase within a stream of audio.

The most common use case of keyword recognition is voice activation of virtual assistants.

CHALLENGES:

1. Background noise:

Environmental sounds like traffic, music, or other people talking can significantly degrade speech quality, making it difficult for a system to accurately transcribe speech.

2. Homophones:

Words that sound the same but have different meanings (e.g., "to" and "too") can lead to recognition errors.

3. Speaker variations:

Different accents, dialects, and speaking styles from various individuals can confuse speech recognition systems.

4. Speech impairments:

People with speech disorders like stuttering or slurred speech can pose challenges for speech processing systems.

SOLUTIONS:

1. Background noise:

Step 1: Convert Speech to Frequency Domain (Fourier Transform)

- Convert raw audio waveform into frequency components using Fast Fourier Transform (FFT).
- Noise is usually concentrated in specific frequency bands.

Step 2: Estimate Noise Profile

- Assume the first few milliseconds are silence or background noise.
- Compute an average noise spectrum from these initial frames.

Step 3: Subtract Noise from Speech

- Apply Spectral Subtraction:

  $$S\_clean(f)=S\_noisy(f)-N(f)$$

  If the result is negative, set it to zero to avoid distortions.

Step 4: Reconstruct the Speech Signal

- Convert the cleaned frequency components back to the time domain using Inverse FFT (IFFT).

2. Homophones:

Step 1: Preprocess Keyword and Homophones

1. Convert speech to sequence of words
2. Build a homophone dictionary H:
   - Example: {"flower": ["flour"], "two": ["to", "too"], "right": ["write"]}
3. Expand the keyword set:
   - Include homophones → K_extended = {K} ∪ H(K).
   - Example: If K = {"flour"}, then K_extended = {"flour", "flower"}.

Step 2: Generate N-grams for Context

1. Extract word sequences from S.
2. Generate N-gram windows (bigrams, trigrams, etc.)

Step 3: Beam Search for Contextual Validation

1. Initialize a beam search queue with B (beam width) candidate keyword sequences.
2. Expand each candidate using the next N-gram words.
3. Score candidates based on context:
   - If a homophone appears, analyze surrounding words.
   - Use language rules or a context dictionary to check meaning.
4. Prune low-probability candidates and keep the top B valid matches.

Step 4: Validate and Output Results

1. For each detected keyword in K_extended, check if it matches K in context.
2. Return all validated occurrences of K.

3. Speaker variations:

Step 1: Convert Speech to Phonemes

- Instead of using raw text, convert words into phonetic units.
- Example:
  - "water" (American) → "wɔtɚ"
  - "water" (British) → "wɔtə"
  - "water" (Indian) → "waːtər"

Step 2: Apply Phoneme Transformation Rules (Accent Normalization)

- Define rules to normalize phoneme variations across accents.
- Example:
  - "r-dropping" (British English) → "caʀ" → "caʰ" (car)
  - "th-fronting" (Cockney) → "th" → "f" (think → fink)

    - "t-flapping" (American) → "t" → "ɾ" (butter → budder)

Step 3: Compute Phonetic Distance (Levenshtein + Soundex/Metaphone)

- If a word sounds similar to a known word but is slightly different, replace it.
- Use Levenshtein Distance (edit distance ≤ 2 means it's likely the same word).
- Use Soundex/Metaphone for phonetic similarity.

Step 4: Adaptive Thresholding for Mispronunciations

- If confidence is low due to an accent, dynamically adjust the threshold.
- If the difference between phonemes is only vowel-based, lower the threshold.

4. Speech impairments:

Step 1: Define a fixed window size (e.g., 3 previous phonemes).

Step 2: Move the window across the speech transcript and check for repeated sequences.

Step 3: If a phoneme appears more than a threshold (e.g., 2 times) in the window, replace it with a single occurrence.

Step 4: Continue sliding until all repetitions are smoothed.