UNIT V: Speech Processing and Advanced NLP Models

Speech Fundamentals: Phonetics and Acoustic Phonetics, Digital Signal Processing in Speech Analysis, Feature Extraction in Speech: Short-Time Fourier Transform (STFT), Mel-Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction (PLP), Hidden Markov Models (HMMs) in Speech Recognition.

SPEECH PROCESSING AND ADVANCED NLP MODELS

1. Introduction

- **Speech Processing** bridges human spoken language and computational systems.
- It covers **speech recognition, speech synthesis, speaker identification, and spoken dialogue systems**.
- Advanced NLP models (deep learning, transformers, etc.) are now integrated with speech to power **voice assistants, MT, and multimodal AI**.

2. Speech Processing

a) Speech Recognition (ASR – Automatic Speech Recognition)

- Converts spoken input into text.
- Pipeline:
  - **Acoustic Modeling** → maps audio signals to phonemes.
  - **Language Modeling** → predicts word sequences.
  - **Decoding** → selects the most likely sentence.
- **Example:** Google Speech API, Alexa.

b) Speech Synthesis (TTS – Text-to-Speech)

- Converts text into natural-sounding speech.
- Methods:
  - Concatenative synthesis (unit selection).
  - Parametric synthesis (HMM-based).
  - Neural TTS (WaveNet, Tacotron).

c) Speaker Recognition

- **Speaker Identification** → "Who is speaking?"
- **Speaker Verification** → "Is this person who they claim to be?"
- Used in security, banking apps.

d) Challenges in Speech Processing

- Noise, accents, dialects.
- Low-resource languages.
- Code-switching (mix of languages).

3. Advanced NLP Models

a) Deep Learning in NLP

- RNNs, LSTMs, and GRUs → capture sequential dependencies.
- Used in **speech recognition, MT, sentiment analysis**.

b) Transformer Models

- Replace recurrence with **self-attention mechanism**.
- Examples: **BERT, GPT, T5, BART**.
- Advantages: parallelization, better handling of context.

c) Large Language Models (LLMs)

- Pre-trained on massive corpora.
- Capabilities: text generation, translation, reasoning, speech-to-text integration.
- Examples: **GPT-4, PaLM, LLaMA**.

d) Speech + NLP Integration

- End-to-end **speech-to-text translation** (e.g., Meta's SeamlessM4T, OpenAI Whisper).
- Multimodal models: handle **speech + text + image** together.

4. Applications

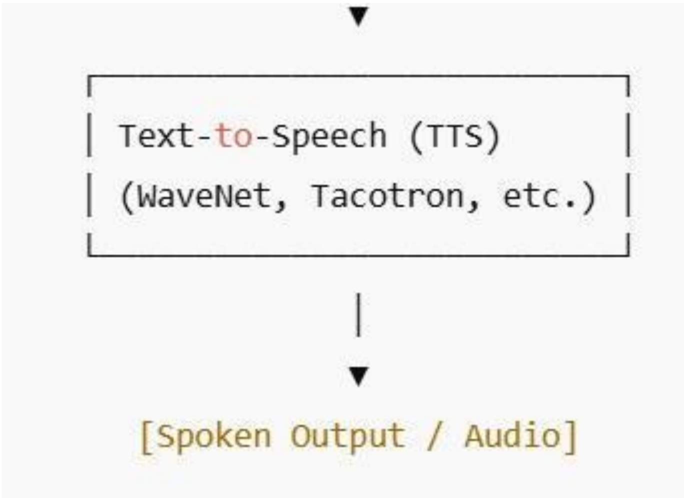
- Voice assistants (Siri, Alexa, Google Assistant).

- Real-time speech translation (Skype Translator, Google Translate).
- Healthcare dictation systems.
- Accessibility tools (screen readers, speech therapy).

5. Conclusion

- Speech Processing enables **human-computer interaction through spoken language**.
- Advanced NLP models (Transformers, LLMs) push speech technologies to new levels.
- Future: **multilingual, multimodal, and context-aware speech systems**.

```
    [Speech Input / Audio]
               |
               ▼
 ┌────────────────────────────┐
 │   Speech Processing        │
 │ (Signal Processing + ASR)  │
 └────────────────────────────┘
               |
               ▼
    [Text Representation]
               |
               ▼
 ┌────────────────────────────┐
 │   Advanced NLP Models      │
 │  (BERT, GPT, T5, etc.)     │
 │ - Translation              │
 │ - Summarization            │
 │ - Q&A / Reasoning          │
 └────────────────────────────┘
               |
               ▼
    [Generated Text Output]
               |
               ▼
```



SPEECH FUNDAMENTALS

Speech fundamentals form the base for all **speech and language technologies**. Understanding **production, signal properties, and types of sounds** helps in building efficient speech-based NLP systems.

1. Introduction

- Speech is the **primary mode of human communication**.
- In **Speech Processing**, understanding speech fundamentals is essential for tasks like **ASR (Automatic Speech Recognition), TTS (Text-to-Speech), and Speaker Recognition**.

2. Speech Production Process

- Human speech is produced through the **vocal tract system**:
    1. **Lungs** → provide airflow.
    2. **Vocal cords (glottis)** → vibrate to produce voiced sounds.
    3. **Articulators** (tongue, lips, teeth, palate) → shape sounds into phonemes.

3. Key Properties of Speech Signal

- **Time-domain features:** waveform, amplitude.
- **Frequency-domain features:** spectrum, formants.
- **Pitch (F0):** perceived as tone of voice, related to vocal cord vibration.

- **Formants:** resonant frequencies of the vocal tract.
- **Phonemes:** smallest speech units (e.g., /p/, /a/, /t/).

4. Speech Signal Characteristics

- **Analog in nature**, continuous wave.
- **Digitization:** speech is sampled & quantized to store in digital form.
- **Sampling rate:** commonly 8 kHz (telephony) or 16 kHz (ASR).
- **Spectrogram:** visual representation (time vs. frequency vs. energy).

5. Types of Speech Sounds

- **Voiced sounds** → vocal cords vibrate (e.g., /a/, /b/).
- **Unvoiced sounds** → no vibration (e.g., /s/, /f/).
- **Vowels** → open vocal tract, periodic.
- **Consonants** → constricted vocal tract, can be noisy.

6. Challenges in Speech Fundamentals

- **Coarticulation** → overlap of sounds.
- **Accents & dialects** → variations in pronunciation.
- **Background noise** → distorts signal.
- **Speaker variability** → pitch, speed, style differences.

7. Applications

- **Speech recognition (ASR).**
- **Voice synthesis (TTS).**
- **Forensic analysis & speaker identification.**
- **Language learning tools.**

PHONETICS AND ACOUSTIC PHONETICS

Phonetics provides the foundation for studying speech sounds, while **acoustic phonetics bridges linguistics and signal processing** by analyzing sound wave properties. This is essential for **ASR, TTS, and other speech-based NLP systems**.

1. Introduction

- **Phonetics** is the study of **speech sounds** – how they are produced, transmitted, and perceived.
- **Acoustic phonetics** is a branch of phonetics focusing on the **physical properties of speech sounds** as sound waves.

2. Branches of Phonetics

1. **Articulatory Phonetics**
   - Studies how speech sounds are produced by the **vocal organs** (tongue, lips, vocal cords).
   - Example: how /p/ differs from /s/.
2. **Acoustic Phonetics**
   - Examines the **physical characteristics of sound waves**.
   - Properties studied: **frequency, amplitude, duration, formants**.
3. **Auditory Phonetics**
   - Studies how humans **perceive and process speech sounds** using the auditory system.

3. Acoustic Properties of Speech Sounds

- **Frequency (Pitch):** rate of vocal cord vibration (measured in Hertz).
- **Intensity (Loudness):** energy of the sound wave.
- **Duration:** length of the sound.
- **Formants:** resonant frequencies that distinguish vowels.
- **Spectrogram:** visual tool showing time, frequency, and intensity.

4. Phonetic Units

- **Phoneme:** smallest unit of sound that changes meaning (*pat* vs. *bat*).

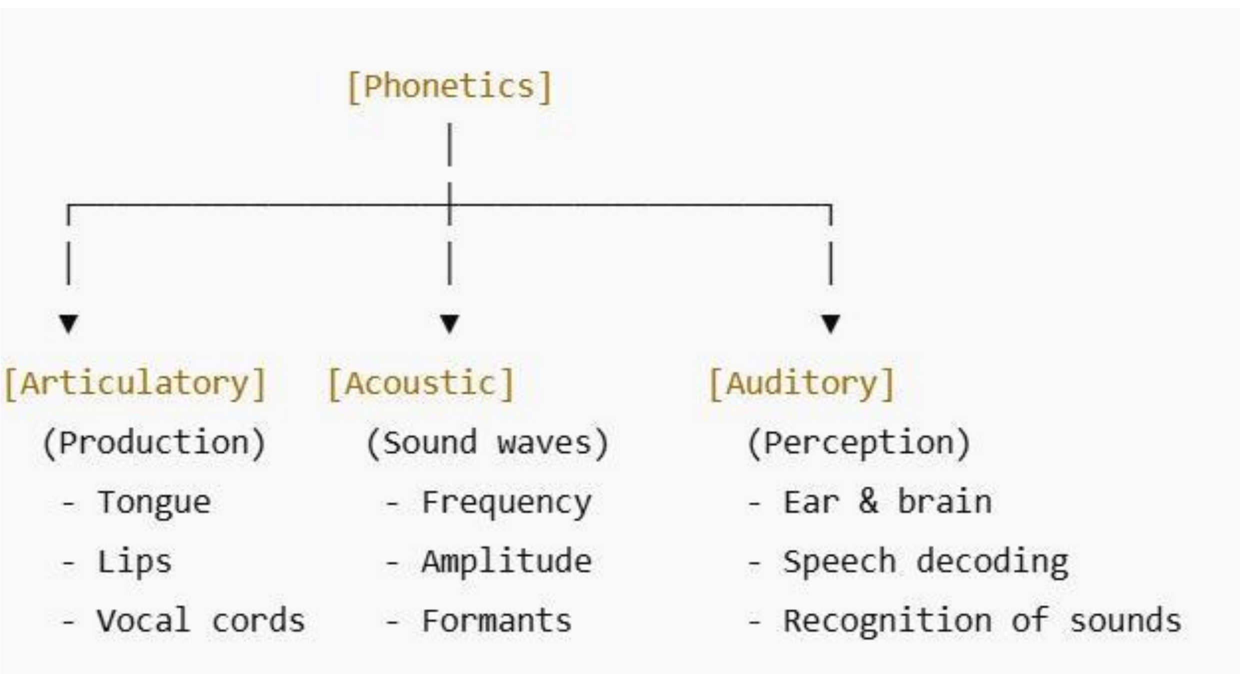
- **Allophone:** variation of a phoneme that doesn't change meaning (*top* vs. *stop*).

5. Applications in NLP & Speech Processing

- **Automatic Speech Recognition (ASR):** converts speech to text.
- **Text-to-Speech (TTS):** synthesizes natural-sounding speech.
- **Speaker Identification:** analyzing unique phonetic-acoustic features.
- **Language Learning Tools:** pronunciation training.

6. Challenges in Acoustic Phonetics

- **Coarticulation:** overlap of sounds in continuous speech.
- **Noise & distortions:** affect accuracy of acoustic features.
- **Accent & dialect variation.**



<u>DIGITAL SIGNAL PROCESSING IN SPEECH ANALYSIS</u>

DSP is the backbone of modern **speech analysis**. By transforming speech into digital form and extracting features, it enables powerful applications in **ASR, TTS, speaker recognition, and healthcare AI**.

1. Introduction

- **Digital Signal Processing (DSP)** involves applying mathematical and computational techniques to analyze, modify, and synthesize speech signals.
- In **speech analysis**, DSP helps extract features from speech for recognition, synthesis, and enhancement.

2. Why DSP in Speech?

- Speech is an **analog waveform** → converted into **digital form** for processing.
- DSP enables:
    - Noise reduction
    - Compression
    - Feature extraction (for ASR, speaker ID)
    - Enhancement of intelligibility

3. Steps in Speech Signal Processing

1. **Speech Acquisition**
    - Microphone records speech (analog).
    - **Sampling** converts it to digital (e.g., 16 kHz, 44.1 kHz).
    - **Quantization** approximates amplitude into discrete levels.
2. **Pre-Processing**
    - **Pre-emphasis filter** → boosts high frequencies.
    - **Framing** → divide signal into short frames (10–30 ms).
    - **Windowing (Hamming/Hanning)** → reduce discontinuities at frame edges.
3. **Spectral Analysis**

- **Fourier Transform (FFT):** converts time domain → frequency domain.
- **Spectrograms:** visualize energy distribution across frequency & time.
- **Formant Analysis:** resonances used in vowel classification.

4. **Feature Extraction**
   - **Mel-Frequency Cepstral Coefficients (MFCCs):** widely used in ASR.
   - **Linear Predictive Coding (LPC):** models speech production.
   - **Pitch & Energy estimation:** for prosody analysis.
5. **Post-Processing / Applications**
   - **Automatic Speech Recognition (ASR)**
   - **Speaker Identification & Verification**
   - **Speech Synthesis (TTS)**
   - **Emotion Recognition**

4. Challenges in DSP for Speech

- **Background Noise** (affects accuracy).
- **Variability in speakers** (age, accent, health).
- **Real-time processing** requirements.

5. Applications in NLP & AI

- Virtual Assistants (Alexa, Siri, Google Assistant).
- Forensic speaker identification.
- Real-time translation systems.
- Medical diagnostics (speech-based Parkinson's detection).

```
  [Speech Input]
        |
        ▼
[Analog-to-Digital Conversion]
   (Sampling & Quantization)
        |
        ▼
   [Pre-processing]
(Pre-emphasis, Framing, Windowing)
        |
        ▼
  [Spectral Analysis]
(FFT, Spectrogram, Formants)
        |
        ▼
  [Feature Extraction]
(MFCC, LPC, Pitch, Energy)
        |
        ▼
    [Recognition /
  Classification System]
(ASR, Speaker ID, TTS, Emotion Recognition)
```

FEATURE EXTRACTION IN SPEECH

Feature extraction transforms **raw, complex speech data** into **compact, discriminative representations**, making it the **core of speech analysis and NLP-based applications**.

UNIT V

1. Introduction

- **Feature Extraction** in speech processing refers to transforming raw speech signals into a set of **compact, discriminative, and meaningful parameters**.
- Goal: represent speech **efficiently** for tasks like **ASR (Automatic Speech Recognition), TTS (Text-to-Speech), Speaker Identification, and Emotion Recognition**.

2. Why Feature Extraction?

- Speech signals are **high-dimensional and redundant**.
- Features reduce complexity while preserving **linguistic and speaker information**.
- Good features must be:
  - **Robust** to noise and channel variations.
  - **Discriminative** between speakers and phonemes.
  - **Compact** for fast processing.

3. Common Speech Features

(a) Spectral Features

1. **Mel-Frequency Cepstral Coefficients (MFCCs)**
   - Most widely used.
   - Mimic human auditory perception (mel scale).
   - Extract spectral envelope for phoneme recognition.
2. **Linear Predictive Coding (LPC)**
   - Models the vocal tract as a filter.
   - Captures formant structure of speech.
3. **Perceptual Linear Prediction (PLP)**
   - Similar to LPC but incorporates psychoacoustic models.

(b) Prosodic Features

1. **Pitch (Fundamental Frequency, F0):** conveys tone, stress, and intonation.

2. **Energy (Intensity):** loudness; useful in emotion recognition.
3. **Duration / Speaking Rate:** helps in rhythm and language modeling.

(c) Temporal & Other Features

- **Delta & Delta-Delta Coefficients:** represent changes in MFCCs over time.
- **Zero Crossing Rate (ZCR):** useful for voiced/unvoiced classification.
- **Formant Frequencies:** resonant frequencies distinguishing vowels.

4. Feature Extraction Pipeline

1. **Speech Acquisition** → Recording through microphone.
2. **Pre-emphasis** → Boost high frequencies.
3. **Framing** → Divide signal into small segments (10–30 ms).
4. **Windowing** → Apply Hamming/Hanning window.
5. **Spectral Analysis** → FFT or filter banks.
6. **Feature Computation** → MFCC, LPC, pitch, energy.

5. Applications

- **ASR:** Phoneme and word recognition.
- **Speaker Identification:** Using LPC/MFCC for voiceprints.
- **Emotion Detection:** Using prosodic features like pitch and energy.
- **Language Processing:** Intonation and rhythm analysis.

```
   [Speech Input]
          |
          ▼
 [Pre-processing]
 (Pre-emphasis Filter)
          |
          ▼
     [Framing]
 (10–30 ms frames)
          |
          ▼
    [Windowing]
 (Hamming / Hanning)
          |
          ▼
 [Spectral Analysis]
 (FFT / Filter Banks)
          |
          ▼
```

```
[Feature Computation]
(MFCC, LPC, PLP, Pitch, Energy,
 Formants, Delta & Delta-Delta)
         |
         ▼
  [Feature Vectors]
(Used for ASR, TTS, Speaker ID, Emotion Recognition)
```

SHORT-TIME FOURIER TRANSFORM (STFT)

The Short-Time Fourier Transform (STFT) is a fundamental DSP tool in speech analysis, enabling time–frequency analysis of non-stationary signals like human speech.

1. Introduction

- Speech is a **non-stationary signal** (its frequency content changes over time).
- The **Fourier Transform (FT)** gives frequency information but loses time resolution.
- The **Short-Time Fourier Transform (STFT)** solves this by analyzing **small segments (windows)** of the signal, providing **time–frequency representation**.

2. STFT Concept

- Divide the speech signal into **short overlapping frames** (e.g., 20–40 ms).
- Apply a **window function** (Hamming/Hanning).
- Perform **Fourier Transform** on each frame.
- Result: **Spectrogram** → a 2D plot (time vs frequency vs magnitude).

3. STFT Mathematical Definition

For signal $x(t)$:

$$STFT\{x(t)\}(\tau,\omega) = \int_{-\infty}^{\infty} x(t) \cdot w(t-\tau) \cdot e^{-j\omega t} dt$$

Where:

- $w(t-\tau)$ = window function centered at time τ.
- ω = angular frequency.

4. Steps in STFT

1. **Speech Input** → continuous-time signal.
2. **Framing & Windowing** → isolate short segment.
3. **Apply Fourier Transform** → extract frequency components.
4. **Repeat across frames** → capture how spectrum changes over time.
5. **Visualize** → Spectrogram (used widely in speech recognition & phonetics).

5. Applications in Speech Processing

- **Speech Recognition (ASR):** extracting time–frequency features.
- **Speaker Identification:** capturing vocal tract signatures.
- **Emotion Recognition:** analyzing prosodic variations.
- **Music/Speech Separation:** separating overlapping sources.
- **Noise Reduction & Enhancement:** filtering noise in time–frequency space.

6. Advantages & Limitations

Advantages:

- Provides **both time and frequency information**.
- Useful for analyzing **non-stationary signals** like speech.

Limitations:

- **Fixed resolution**: trade-off between time and frequency (uncertainty principle).
- Choice of **window size** is critical:
  - Small window → better time resolution, poor frequency resolution.
  - Large window → better frequency resolution, poor time resolution.

```
  [Speech Signal]
         |
         ▼
    [Framing]
(Divide into short frames)
         |
         ▼
   [Windowing]
(Hamming / Hanning)
         |
         ▼
[Fourier Transform]
(Each frame separately)
         |
         ▼
[Time-Frequency Representation]
       (Spectrogram)
```

MEL-FREQUENCY CEPSTRAL COEFFICIENTS (MFCC) AND PERCEPTUAL LINEAR PREDICTION (PLP)

- **MFCC = widely used, simple, effective, but noise-sensitive.**

- **PLP = psychoacoustic, more noise-robust, better for practical ASR.**

1. Introduction

Speech recognition requires **compact, discriminative, and perceptually motivated features**. Two popular methods are:

- **MFCC** – based on the Mel-scale of human hearing.
- **PLP** – based on auditory perception and linear prediction.

2. Mel-Frequency Cepstral Coefficients (MFCC)

Concept:

- Mimics how humans perceive sound frequencies.
- Uses a **Mel-scale filter bank** that emphasizes low frequencies (where human hearing is more sensitive).
- Produces **cepstral coefficients** representing speech spectrum compactly.

Steps:

1. **Pre-emphasis** – boost high frequencies.
2. **Framing & Windowing** – short segments (20–40 ms).
3. **FFT** – convert to frequency domain.
4. **Mel Filter Bank** – apply triangular filters spaced on Mel scale.
5. **Logarithm** – mimic human loudness perception.
6. **DCT (Discrete Cosine Transform)** – decorrelate features, keep 12–13 MFCCs.

Applications:

- Automatic Speech Recognition (ASR).
- Speaker Identification.
- Emotion Recognition.

3. Perceptual Linear Prediction (PLP)

Concept:

- Developed by **Hermansky (1990)**.
- Based on **psychoacoustic principles** of human hearing.
- Uses **Linear Prediction Analysis** but modifies the spectrum to match auditory perception.

Steps:

1. **Critical Band Analysis** – filters based on Bark scale.
2. **Equal Loudness Pre-emphasis** – compensates for ear sensitivity.
3. **Intensity Loudness Compression** – cube-root compression.
4. **LP Analysis** – models vocal tract with all-pole filter.
5. **Cepstral Coefficients** – converted from LP coefficients.

Applications:

- ASR under noisy conditions.
- Robust speech recognition systems.

4. Comparison: MFCC vs PLP

Aspect	**MFCC**	**PLP**
Scale	Mel scale (human pitch perception)	Bark scale (critical bands of hearing)
Compression	Logarithm (log energy)	Cube-root (loudness perception)
Basis	Cepstral (DCT of log energies)	Linear Prediction (LP + perceptual model)
Robustness	Sensitive to noise	More robust in noisy environments
Applications	ASR, Speaker ID, Emotion recog.	ASR, esp. noisy speech recognition

MFCC Pipeline

```css
[Speech Signal] → [Pre-emphasis] → [Framing & Windowing]
      → [FFT] → [Mel Filter Bank] → [Log] → [DCT] → [MFCCs]
```

PLP Pipeline

```css
[Speech Signal] → [Critical Band Analysis]
      → [Equal Loudness] → [Intensity Compression]
      → [LP Analysis] → [Cepstral Coefficients]
```

HIDDEN MARKOV MODELS (HMMS) IN SPEECH RECOGNITION

HMMs were the **foundation of modern speech recognition**, providing the statistical framework for modeling phoneme sequences.
While **deep learning has surpassed HMMs**, many systems still use **hybrid HMM-DNN models** in real-world ASR.

1. Introduction

- **Hidden Markov Model (HMM):** A probabilistic model used to represent **sequential data** such as speech.
- Speech is a **time-varying signal** where sounds (phonemes) unfold over time.
- HMM captures **temporal dynamics** + **probabilistic state transitions**.

2. Why HMM for Speech?

- Speech has **variability** (speaker, rate, noise).
- Phonemes are not directly observable; they are **hidden states**.
- Acoustic signals are observable outputs generated by these hidden states.

- HMM provides a **mathematical framework** to model:
  - **State transitions** (sequence of phonemes).
  - **Observation likelihoods** (acoustic features).

3. Structure of HMM

- **States (S):** Represent phonemes or sub-phonemes (e.g., S = {s1, s2, ..., sn}).
- **Transitions (A):** Probabilities of moving from one state to another.
- **Observations (O):** Acoustic feature vectors (MFCCs, PLPs).
- **Emission Probabilities (B):** Likelihood of observing feature vector given a state.
- **Initial Probabilities (π):** Probability distribution of starting states.

Mathematically:

HMM = (A, B, π)

- **A** = state transition probabilities
- **B** = observation probability distribution
- **π** = initial state distribution

4. Key Problems in HMM

1. **Evaluation Problem:**
   - Given model (λ) and observation sequence (O), compute P(O|λ).
   - Solved using **Forward Algorithm**.
2. **Decoding Problem:**
   - Find most likely state sequence for given observation sequence.
   - Solved using **Viterbi Algorithm**.
3. **Training Problem:**
   - Adjust model parameters (A, B, π) to maximize P(O|λ).
   - Solved using **Baum-Welch Algorithm (EM algorithm)**.

---

5. Application in Speech Recognition

Pipeline:

```java
Speech Signal → Feature Extraction (MFCC/PLP) → HMM Modeling → Decoding → Recognized Words
```

- **Acoustic Model (HMM):** Maps feature vectors to phonemes.
- **Lexicon (Pronunciation Dictionary):** Maps phonemes to words.
- **Language Model (n-grams):** Provides word sequence probability.

---

6. Example: Word Recognition

Suppose we want to recognize the word "**cat**":

- HMM states represent phonemes: **/k/ /æ/ /t/**.
- Each phoneme modeled with 3 HMM states (beginning, middle, end).
- Input speech is converted into MFCC vectors.
- HMM computes the **most probable state sequence** that generated the observation.
- Result: recognized word.

7. ASCII Diagram of HMM in Speech

```rust
States (Hidden):   s1 ----> s2 ----> s3 ----> s4 ----> s5
                     ↘        ↘        ↘        ↘
Observations:       o1       o2       o3       o4 ... (MFCC vectors)
```

8. Advantages & Limitations

Advantages:

- Captures temporal dynamics of speech.
- Provides efficient algorithms (Forward, Viterbi).
- Widely used in ASR before deep learning.

- **Limitations:**
  - Assumes **Markov property** (current state depends only on previous).
  - Observation distributions often Gaussian → limited expressiveness.
  - Struggles with long-term dependencies.
  - Replaced by **Deep Neural Networks (DNN-HMM hybrids, end-to-end models like RNNs/Transformers)**.