PABNA UNIVERSITY OF SCIENCE AND TECHNOLOGY

Faculty of Engineering & Technology

Department of Information and Communication Engineering

Lab Report

Course Title: Neural Networks Sessional

Course Code: ICE-4206

SUBMITTED BY:	SUBMITTED TO:
Name: Md. Omar Faruk	Dr. Md. Imran Hossain
Roll:190605	Associate Professor
Session:2018-2019	Department of Information and Communication Engineering
4th Year 2nd Semester	Pabna University of Science and Technology
	……. Signature

Submission Date: August 9, 2023

1. Explain in detail text-to speech conversion system.

A Text-to-Speech (TTS) conversion system is a technology that converts written text. Here's a detailed explanation of how a typical TTS system works into spoken speech. It enables computers or devices to read aloud text in a human-like virtual assistants, language learning apps, audiobooks, navigation systems, and more voice. TTS systems are widely used in various applications, including accessibility tools,

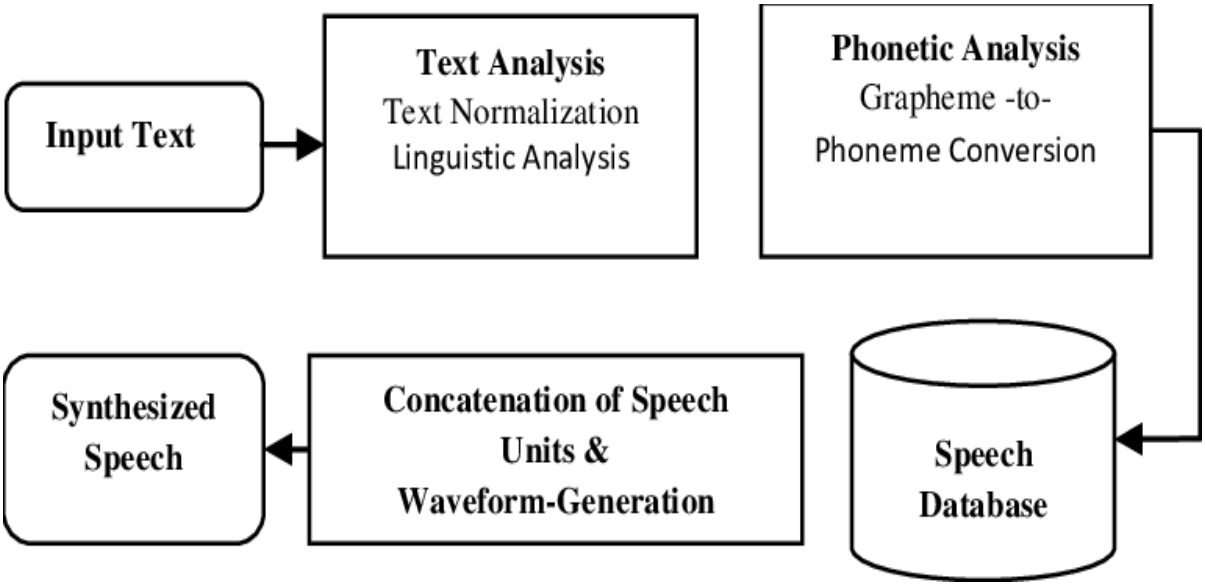


Fig-01: Text to Speech

Text Preprocessing: The input text is first preprocessed to remove any special characters, formatting, or other non-textual elements. The system may also handle language-specific nuances, such as tokenization, part-of-speech tagging, and linguistic analysis.

Text Analysis: The preprocessed text undergoes linguistic analysis, which involves parsing the sentence structure, determining the syntactic relationships between words, and identifying the appropriate pronunciation for each word.

Phonetic Representation: The TTS system converts the analyzed text into phonetic representation. Phonetics refers to the study of the sounds used in human speech. Each word is broken down into phonemes, which are the smallest units of sound in a language.

Prosody and Emphasis: The TTS system also considers prosody, which includes aspects like pitch, duration, and intensity of speech. Emphasis and intonation patterns are determined based on the context and punctuation marks in the input text.

Language and Voice Selection: Modern TTS systems often support multiple languages and voices. The user can select the desired language and voice style to customize the output.

Voice Generation (Synthesis): This is the core process where the TTS system synthesizes the speech waveform based on the phonetic representation, prosody, and voice characteristics. There are generally two main approaches for voice generation:

- **Concatenative TTS:** This method uses a large database of recorded human speech, known as a speech corpus. The system selects and concatenates small audio units (phonemes, diphones, or triphones) from the corpus to create the desired speech. While this approach can produce high-quality, natural-sounding speech, it requires a large amount of data.
- **Parametric TTS:** This method relies on statistical models to generate speech. These models are trained on smaller datasets and use mathematical functions to represent speech. Parametric TTS systems can be more compact and require less storage but might sacrifice some naturalness.

Post-processing: The synthesized speech might undergo some post-processing to improve the output quality. Techniques like signal processing, smoothing, and noise reduction may be applied to make the speech sound more natural and clear.

Output: The final output is an audio file (usually in WAV, MP3, or other audio formats) containing the synthesized speech. This audio can be played through speakers or headphones to allow users to hear the converted text. It's important to note that with advancements in neural network-based TTS models, many modern systems use deep learning techniques like WaveNet and Tacotron to achieve highlynatural and expressive speech synthesis, often rivaling human speech. These models are capable of capturing subtle nuances and emotions in the generated speech, making the overall TTS experience more engaging and human-like.

2.Explain in detail estimation of formant and pitch parameters using cepstrum.

Cepstrum is a transformation of the logarithm of the power spectrum of a signal. It is often used in speech processing for the estimation of formant and pitch parameters.

The basic idea behind cepstral formant estimation is that the cepstrum of a voiced speech signal has a series of peaks at frequencies that are multiples of the fundamental frequency (F0). These peaks correspond to the formant frequencies of the vocal tract.

To estimate the formant frequencies, we can first compute the cepstrum of the speech signal. Then, we can use a peak detection algorithm to find the peaks in the cepstrum that are multiples of F0. The frequencies of these peaks are the formant frequencies.

The pitch frequency can also be estimated from the cepstrum. The pitch frequency corresponds to the frequency of the most prominent peak in the cepstrum.

Here is a more detailed explanation of the cepstral formant estimation procedure:

1. The speech signal is divided into frames of equal length.
2. The power spectrum of each frame is computed.
3. The logarithm of the power spectrum is taken.
4. The cepstrum of the logarithm of the power spectrum is computed.
5. A peak detection algorithm is used to find the peaks in the cepstrum that are multiples of F0.
6. The frequencies of the peaks are the formant frequencies.
7. The frequency of the most prominent peak in the cepstrum is the pitch frequency.

Cepstral formant estimation is a relatively simple and efficient method for estimating formant and pitch parameters. However, it can be sensitive to noise and distortions in the speech signal. Here are some additional considerations for cepstral formant estimation:

- The length of the frames used to compute the power spectrum and cepstrum can affect the accuracy of the formant and pitch estimates. Longer frames are more robust to noise, but they can also be less accurate for high-pitched voices.
- The peak detection algorithm used to find the formant peaks can also affect the accuracy of the estimates. Some peak detection algorithms are more sensitive to noise than others.
- The cepstral formant estimation procedure can be improved by using a technique called pre-emphasis. Pre-emphasis is a process that boosts the high-frequency components of the speech signal. This can help to improve the accuracy of the formant and pitch estimates, especially for high-pitched voices.

Overall, cepstral formant estimation is a powerful and versatile tool for estimating formant and pitch parameters in speech signals. It is relatively simple to implement and can be used with a variety of speech processing applications.

3. Draw and explain, HMM based isolated word speech recognition system.

A Hidden Markov Model (HMM) based isolated word speech recognition system is a fundamental architecture for recognizing individual words in spoken language. It is called "isolated word" because each word to be recognized is assumed to be spoken in isolation, without any contextual information. Here's a high-level explanation of the system along with a simple diagram:

System Components:

- Feature Extraction: The speech signal is preprocessed to extract relevant features that represent the characteristics of the speech. Commonly used features include Mel Frequency Cepstral Coefficients (MFCCs) or filter banks, which capture the spectral content of the speech.

- **HMM Training**: In the training phase, a set of HMMs is created, each representing a specific word to be recognized. The HMMs are trained on a large dataset of labeled speech samples, where the alignment of each speech sample with its corresponding word is known. The Baum-Welch algorithm (an expectation-maximization algorithm) is often used for training the HMMs.

- **3.HMM Construction:** Each HMM consists of three main components:

    1. **State Set:** A set of hidden states, where each state represents a different phonetic unit or acoustic context.
    2. **State Transitions**: Probabilities of transitioning from one state to another, modeling the dynamics of speech production.

3. **Emission Probabilities:** Probabilities of emitting each feature (MFCC or filter bank) given the current state, representing the likelihood of observing a feature in that state.

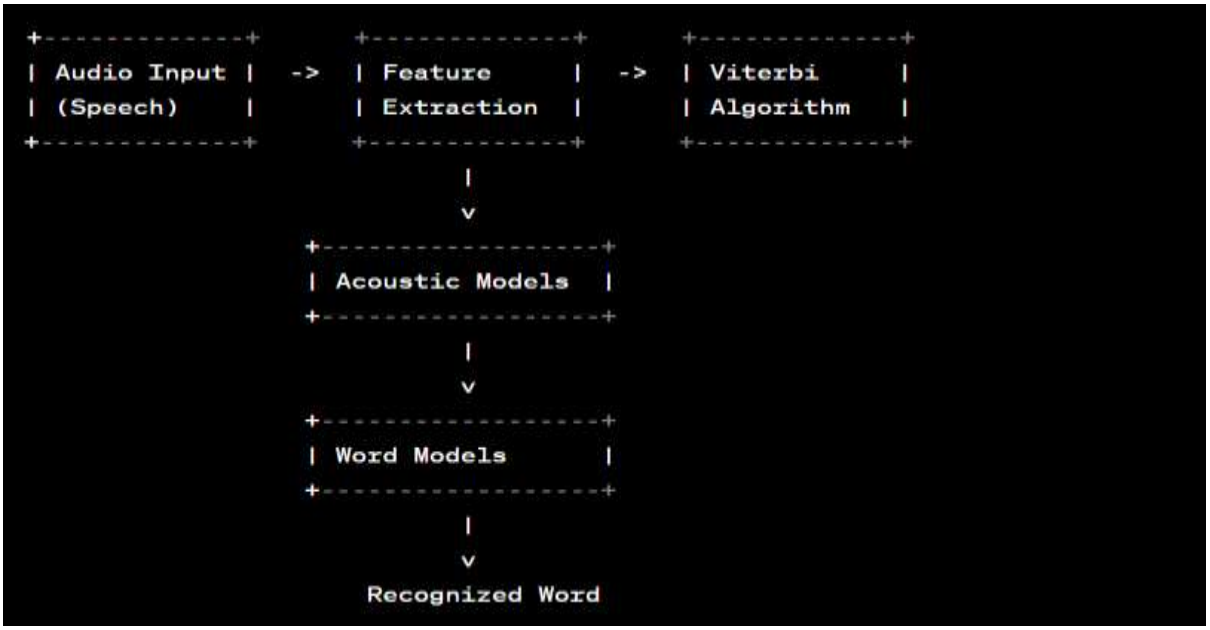
- **Acoustic Model:** The collection of HMMs trained in the previous step forms the acoustic model. It maps acoustic features from the speech signal to sequences of hidden states.
- Decoding: During the recognition phase, the system takes the input speech signal and extracts its features using the same method as in training. The extracted features are then used as observations for the HMMs.

- **6.Word Level HMM:** A separate HMM is used at the word level to model the temporal relationships between the phonemes of the word. This word-level HMM is called a "triphone" HMM and can capture context-dependent variations of phonemes.

System Workflow:

- **Feature Extraction:** The speech signal is transformed into a sequence of feature vectors (MFCCs or filter banks).
- **Acoustic Modeling:** The feature sequence is used as input to the acoustic model, which consists of the collection of HMMs trained for each word.
- **Decoding:** The system employs algorithms like the Viterbi algorithm or the Forward-Backward algorithm to find the most likely sequence of hidden states (phonemes) that generated the observed features.
- **Word Recognition:** The recognized sequence of phonemes is mapped to the corresponding words using the word-level HMM (triphone HMM). The word with the highest likelihood is considered the recognized word.

Diagram:



In this diagram, the system takes an audio input, extracts features, and uses the Viterbi algorithm to find the most likely sequence of states in the HMMs associated with word models. The recognized word corresponds to the final state in the Viterbi path.

Keep in mind that this is a simplified overview, and actual speech recognition systems are more complex and involve various optimizations and techniques to improve accuracy and efficiency.

4. Explain in detail automatic speech recognition system with suitable example.

Automatic Speech Recognition (ASR) is a technology that converts spoken language into written text. It's a crucial component in various applications like voice assistants, transcription services, language translation, and more. ASR systems utilize a combination of signal processing, machine learning, and linguistic knowledge to perform this task. Here's a detailed explanation of how an ASR system works along with a suitable example:

Components of an ASR System:

Audio Input: The ASR process begins with an audio input containing spoken language. This input could come from various sources like microphones, recordings, phone calls, etc.

Signal Preprocessing: The raw audio signal is preprocessed to enhance its quality and remove noise. This step involves techniques like noise reduction, filtering, and resampling to ensure optimal input for further processing.

Feature Extraction: The preprocessed audio signal is transformed into a format that is suitable for machine learning. The most commonly used feature representation is the Mel-Frequency Cepstral Coefficients (MFCCs). These coefficients capture the spectral characteristics of the sound signal over time.

Acoustic Modeling: Acoustic modeling aims to map the extracted features to phonetic units (subword speech units). This involves building a model that represents the relationship between audio features and phonemes, which are the smallest units of sound in a language. Hidden Markov Models (HMMs) and Deep Neural Networks (DNNs) are commonly used for this purpose.

Language Modeling: Language modeling involves predicting the sequence of words based on the context of the speech. It helps the ASR system understand the likelihood of word sequences in a given language. Statistical language models and neural language models like Recurrent Neural Networks (RNNs) and Transformers are used for this task.

Decoding and Alignment: In this step, the ASR system uses the acoustic model and the language model to decode the audio input into a sequence of words or phonemes. The goal is to find the most probable word sequence that corresponds to the input audio.

Post-processing: The output of the decoding step might contain errors, as ASR is inherently noisy due to variations in pronunciation, background noise, and other factors. Post-processing techniques like language model rescoring and word-level confidence scoring help improve the accuracy of the transcription.

Example Scenario:

Let's say you have an ASR system that's part of a voice assistant application. You speak into your smartphone's microphone, saying, "What's the weather like today?"

Audio Input: Your spoken sentence is captured by the smartphone's microphone.

Signal Preprocessing: The background noise is filtered out, and the audio is resampled to a standard frequency.

Feature Extraction: The audio is transformed into a sequence of MFCCs that represent the spectral characteristics of the sound.

Acoustic Modeling: The ASR system's acoustic model analyzes the MFCCs and identifies phonetic units that correspond to each sound in your sentence.

Language Modeling: The language model considers the context of the sentence and predicts the most likely word sequence based on the identified phonetic units.

Decoding and Alignment: The ASR system uses the acoustic and language models to decode the phonetic units into words. It finds the most probable sequence of words that matches the phonetic units.

Post-processing: The ASR output might be something like "What's the wether like today?" due to pronunciation variations. The post-processing step corrects errors and produces the final transcription: "What's the weather like today?"

In this example, the ASR system successfully converted your spoken sentence into a written form despite challenges like pronunciation differences and potential background noise.

It's important to note that ASR technology has evolved significantly, and more advanced systems use deep learning techniques, such as end-to-end neural ASR models, which combine acoustic and language modeling into a single neural network. These models learn directly from audio features and produce transcriptions without explicitly separating the steps mentioned above.

5.What is the difference between speaker identification and speaker verification? What are the features used for speaker recognition / verification system and how?

Speaker identification and speaker verification are both techniques for identifying a speaker from their voice. However, they have different goals.

- Speaker identification is the task of identifying who spoke something among a determined list of speakers. For example, a speaker identification system could be used to identify the caller of a phone call, or the speaker in a video recording.

- Speaker verification is the task of accepting or rejecting the identity claim of a speaker. For example, a speaker verification system could be used to authenticate a user's voice for access to a computer system, or to make a payment using a voice-activated payment system.

The features used for speaker recognition / verification systems can be divided into two categories:

- Physical features are features that are related to the speaker's vocal tract, such as the length of the vocal cords, the size of the mouth, and the shape of the tongue. These features are relatively stable over time, so they can be used to identify a speaker even if they have changed their voice over time.
- Environmental features are features that are related to the environment in which the speech is recorded, such as the background noise, the reverberation of the room, and the distance between the speaker and the microphone. These features can vary over time, so they are not as reliable for speaker identification as physical features.

The most common features used for speaker recognition / verification systems are:

- Mel-frequency cepstral coefficients (MFCCs) are a set of features that are derived from the short-term Fourier transform of the speech signal. MFCCs are relatively robust to noise and reverberation, and they have been shown to be effective for speaker recognition.
- Linear predictive coding (LPC) coefficients are a set of features that are derived from the autoregressive model of the speech signal. LPC coefficients are also relatively robust to noise and reverberation, and they have been shown to be effective for speaker recognition.
- Formant frequencies are the frequencies at which the vocal cords vibrate. Formant frequencies are unique to each speaker, and they can be used for speaker identification.
- Energy is the loudness of the speech signal. Energy can be used to distinguish between different speakers, especially if the speakers have different vocal ranges.

Speaker recognition / verification systems typically use a combination of these features to identify a speaker. The features are first extracted from the speech signal, and then they are used to train a model of the speaker's voice. When a new speech sample is received, the features are extracted from the sample and compared to the model. If the features are similar enough, the system will identify the speaker. Speaker recognition / verification systems are still under development, but they have become increasingly accurate in recent years. These systems are now being used in a variety of applications, such as:

- Access control
- Payment systems
- Biometric authentication
- Telephone banking
- Virtual assistants

As speaker recognition / verification systems become more accurate, they are likely to be used in even more applications in the future.

6.Explain the concept of linear prediction coding (LPC) in the analysis of speech.

Linear Predictive Coding (LPC) is a widely used technique in the analysis and modeling of speech signals. It is particularly effective in representing the vocal tract characteristics of a speaker and is commonly used in speech compression, speech synthesis, and speaker recognition systems. The main idea behind LPC is to model a speech signal as a linear combination of past samples, enabling the estimation of the formant frequencies and vocal tract resonances. Let's delve into the concept of Linear Predictive Coding in the analysis of speech:

Basic Principle: The primary assumption in LPC is that a speech signal can be approximated by a linear combination of its past samples. Instead of modeling the entire speech signal, LPC focuses on modeling the short-term characteristics of the signal, typically within a small time frame (20-30 milliseconds).

Auto-regressive Model: In LPC, the speech signal is considered an auto-regressive process, where each sample is predicted as a weighted sum of its past samples. The model equation for LPC is:

$x(n) = \sum(a\_i * x(n-i)) + e(n)$

where:

● x(n) is the current speech sample at time n.

● a_i are the LPC coefficients (weights) that represent the influence of past samples on the current sample.

● e(n) is the prediction error, which is the difference between the actual speech sample and the predicted sample.

Estimating LPC Coefficients: The LPC coefficients (a_i) are estimated using various methods, with the most common being the autocorrelation method or the covariance method. These methods involve solving a set of linear equations to find the optimal values of the LPC coefficients that minimize the prediction error.

Applications of LPC:

- Speech Compression: LPC can be used to represent speech signals more efficiently by transmitting only the estimated LPC coefficients and the prediction error, rather than the original speech samples. This reduces the amount of data required for transmission or storage.
- Speech Synthesis: LPC is used in speech synthesis to generate speech waveforms by filtering noise through an LPC filter with estimated coefficients. This technique is particularly useful for text-to-speech systems.
- Speaker Recognition: LPC can be used as a feature extraction method for speaker recognition systems, capturing the unique vocal tract characteristics of individual speakers.

- Voice over IP (VoIP): LPC is used in VoIP applications to compress and transmit speech over networks efficiently.
- Formant Frequencies: One of the significant benefits of LPC is that it can estimate the formant frequencies of a speech signal. Formants are the resonant frequencies of the vocal tract during speech production and play a crucial role in determining the quality of speech sounds. By estimating the LPC coefficients, one can calculate the formant frequencies and gain valuable insights into the articulation and characteristics of speech sounds.

In summary, Linear Predictive Coding (LPC) is a technique used in the analysis of speech signals, allowing the estimation of vocal tract characteristics and formant frequencies. It has diverse applications in speech processing and communication systems due to its ability to efficiently represent speech signals and capture the unique characteristics of individual speakers.

7.Explain dynamic 'time-warping' with regards to speech recognition.

Dynamic Time Warping (DTW) is a technique used in speech recognition and various other fields to align and compare sequences that have different lengths or variable time durations. It's particularly useful when comparing sequences that might be similar but have variations in timing, speed, or rate of change. In the context of speech recognition, DTW helps match spoken utterances to reference templates, even when the spoken utterance may have variable speech rate or timing.

Basic Idea:

Speech signals are inherently dynamic, and the speed at which we speak can vary. This variability can make direct comparison difficult, especially when you want to compare two sequences with different speeds or rates of change. DTW addresses this problem by warping or stretching one sequence in the time domain to optimally align with another sequence, minimizing the mismatch between corresponding elements.

DTW Algorithm:

The DTW algorithm follows these general steps:

- **Initialization**: Create a matrix that represents the cost of aligning each element of the two sequences. The dimensions of the matrix are determined by the lengths of the two sequences being compared.
- **Dynamic Programming**: Starting from the top-left corner of the matrix, compute the cumulative cost of aligning each element of one sequence with each element of the other sequence. The cumulative cost accounts for both the element-to-element distance and the accumulated costs from the previous steps.
- 3. **Optimal Path**: Find the path through the matrix that minimizes the total cost. This path represents the optimal alignment between the two sequences. The path is constrained to move only in a diagonal, upward, or rightward direction, ensuring that each element is aligned with a nearby element in the other sequence.

- 4. **Backtracking**: Once the optimal path is found, backtrack through the matrix to determine which elements in the two sequences are aligned.
- 5. **Distance Calculation**: The final DTW distance is the accumulated cost along the optimal path. This distance represents the similarity between the two sequences, accounting for temporal variations.

Speech Recognition with DTW:

In speech recognition, DTW is often used to compare a spoken utterance with reference templates representing known words or phonemes. Since speech signals can vary in speed, pitch, and timing, DTW helps find the best alignment between the spoken utterance and the reference template. The reference templates are typically represented by sequences of feature vectors, such as Mel-Frequency Cepstral Coefficients (MFCCs) or other acoustic features.

For example, if you want to recognize a spoken word "apple," you'd compare the MFCC sequence of the spoken word with the reference template for "apple." DTW would stretch or compress the time axis of the MFCC sequence to optimally align with the reference template, taking into account variations in speech rate or timing.