**Slide 1:** Hello Everyone!

This is Mahinder AND Mayank Singh Tomar.

Today our group will be giving a presentation and demo on,

“Text and Audio Processing Application”.

Our project is being mentored by Prof. D.G. Auradhkar .

**Slide 2:** As we all know nowadays people talk a lot about improvement of the human interface to the computer. This is because people don't want to sit in front of the monitor and read the data as it causes strain to eyes. In this aspect, speech synthesis is one of the top most step coming forward to improve human interface to the computer. Another step is audio interface. It allows us to directly communicate to computer interface. There can be no other good example than Google's search engine which we almost use daily. It helps us to interact with the commands by speech directly.

**Slide 3:** So here we come up with one GUI based application called Text and Audio Processing Application. This is the overview of our Proposed System. Here we have 3 features Audio to Text, Text to Audio and Language Translator. Now let’s understand how these features are working.

**Slide 4:** Our first feature is Audio-to-Text i.e., Speech-to-text which is based on Speech Recognition Module. Now let’s see how speech recognition module is used in speech-to-text. The basic principle of [Speech Recognition](https://www.elprocus.com/understanding-voice-recognition/) involves the fact that speech or words spoken by any human being cause vibrations in air, known as sound waves. These continuous or analog waves are digitized and processed and then decoded to appropriate words and then appropriate sentences.

**Slide 5:** Now let’s see the Components of a speech recognition System: -

1. **A speech capturing Device**: It consists of a microphone, which converts the sound wave signals to electrical signals and an Analog to Digital Converter which samples and digitizes the analog signals to obtain the discrete data that the computer can understand.
2. **A Digital Signal Module or a Processor**: It performs processing on the raw speech signal like frequency domain conversion, restoring only the required information etc.
3. **Pre-processed signal storage**: The pre-processed speech is stored in the memory to carry out further task of speech recognition.
4. **Reference Speech patterns**: The computer or the system consists of predefined speech patterns or templates already stored in the memory, to be used as the reference for matching.
5. **Pattern matching algorithm**: The unknown speech signal is compared with the reference speech pattern to determine the actual words or the pattern of words.

**Slide 6:** Now let’s see the Working of a Speech Recognition System: -

* A speech can be seen as an acoustic waveform, i.e., signal carrying message information. A normal human being with the limited rate of motion of his/her articulators (speech organs) can produce speech at an average rate of 10 sounds per second. The average information rate is about 50-60 bits/second. It means actually only 50 bits/second of information is required in the speech signal. This acoustic waveform is converted to analog electrical signals by the microphone. The Analog to Digital converter converts this analog signal to digital samples by taking precise measurements of the wave at discrete intervals.
* The digitized signal consists of a stream of periodic signals sampled at 16000 times per second and is not suitable to carry out actual [speech recognition](https://www.elprocus.com/understanding-voice-recognition/) process as the pattern cannot be easily located. To extract the actual information, the signal in time domain is converted to signal in frequency domain. This is done by the Digital Signal Processor using FFT (i.e. Fast Fourier Transform) technique. In the digital signal, the component after every 1/100th of a second is analysed and the frequency spectrum for each such component is computed. In other words, the digitized signal is segmented into small parts of frequency amplitudes.
* Each segment or the frequency graph represents the different sounds made by human beings. The computer performs the matching of the unknown segments with the stored phonetics of the particular language. This pattern matching is done in 3 ways:

1. **Using a Acoustic phonetic approach**:In the Acoustic phonetic approach, generally the Hidden Markov Model is used. This model develops a non deterministic probability model for the speech recognition. This model consists of two variables – the hidden states of the phonemes stored in the computer memory and the visible frequency segment of the digital signal. Each phoneme has its own probability and the segment is matched with the phoneme according to the probability and the matched phonemes are then collected together to form the correct words according to the stored grammar rules of the language.
2. **Using a pattern recognition approach**:In the pattern recognition approach, the system is trained with a particular speech pattern for any language and the unknown speech pattern is compared with the reference speech pattern by determining the distance between the signals using time wrapping technique.
3. **Using Language models**: A language model takes the words and ties them together to make sentences, i.e.it predicts the most likely sequence of words (or text strings) among several a set of text strings.

**Slide new :**

Now let’s take an example.

1. In Step 1, Here user gives an input “dolphins swim fast” to the computer through microphone, this input is of form analog wave.
2. In Step 2, This analog wave gets converted to digital signals using ADC Converter (in this case it is Microphone).
3. Now In Step 3, to extract the actual information, the signal in time domain is converted to signal in frequency domain. The digitalized signal is segmented into small parts of frequency amplitudes called Phonemes.
4. Before going to step 4 let’s suppose I’m a speech recognizer, now in Step 4, there are 2 layers. Let’s c what happens in this layers. In Layer 1 I heard a sound ‘D’, so the phonemes related to this sound are /t/, /th/, /d/. here /d/ has more probability so I’ll take it. And will go to next layer, the next sound I heard is ‘oh’ and the phoneme related to this layer is /o/ and statistically /o/ is more is probable to be followed by /d/. similarly, for /L/, /L/ is statistically more probable to be followed by /o/. So here I got word dol.
5. In step 4 I got a word dol but it doesn’t have any meaning. So, in step 5, I’ll search a pattern of word similar in pronunciation to dol. here I got a word Doll and yes it has meaning so I’ll take it and will move further. This all steps ill do for phins also, so I got two word for phins i.e., fins and phins. Now ill match it with doll and will try to make a meaningful word. But here as I’m not getting it, So I’ll go to previous step i.e. step 4 and will check this matching of phins word with word dol. And here im getting a match with phins. So our word is Dolphins. Similarly for swim and fast I’ll do all the steps. For fast I got 2 words passed and fast but as passed has more probability I’ll go with passed. So I got a set of words Dolphins swim passed.
6. Now in Step 6 ill check whether the sequence of this words in set are grammatically correct or wrong. And here as grammatically it is not possible to have 2 consecutive verbs in a sentence. I’ll select fast over passed as it is adverb and it is making a meaningful sentence. So here I got a final output as Dolphins swim Fast.

**Slide 7:** Now Let’s see the Factors on which Speech Recognition System depends: -

1. **Isolated Words:** There needs to be a pause between the consecutive words spoken because continuous words can overlap making it difficult for the system to understand when a word starts or ends. Thus, there needs to be a silence between consecutive words.
2. **Single Speaker:** Many speakers trying to give speech input at the same time can cause overlapping of the signals and interruptions. Most of the speech recognition systems used are speaker dependant systems.
3. **Vocabulary size:** Languages with large vocabulary are difficult to be considered for pattern matching than those with small vocabulary as chances of having ambiguous words are lesser in the latter.

**Slide 8:** Now, let’s take a look on the next main feature of Text & Audio processing i.e. Text-to-Speech (TTS) system. This TTS system is basically build of Python based modules like Pyttsx3 & gTTS where we’ll be able to convert any type of inputted text to voice & text file to Audio format.

**Slide 9:** This is the exact working diagram of Text conversion to speech. First of all, after getting text as our source input it’ll be normalized & then passed onto the Natural Language Processing (NLP) block also called as front-end of TTS where Text processing or say linguistic analysis of text will be done like pronounciation generation, prosody generation which we’ll see shortly. After that the generated phonemes & prosody information will be passed onto the Digital Signal Processing (DSP) block where by using any Mathematical computational models or algorithms we’ll be able to generate our speech.

**Slide 10:** Now, to pass inputted text to NLP first we need to normalize it. Because not entire text we get can be in written word format only, there can be some symbols, abbreviations & numbers, etc. Like we see if abbreviations like Dr. , Rs. are there then it should be in written word format like doctor & Rupees and similarily the same process for numbers & other symbols. This processing is also known as Text Pre-processing which helps us to understand text clearly.

**Slide 11:** After Normalization process Text analysing is done in NLP where 1st process is to convert graphemes to phonemes. Grapheme means written text word which is then converted word by word to phonemes. Phonemes basically means pronounciation, they are the smallest units of the word. G to P conversion is actually converting written text words to its respective pronounciations. This process is also knowns as segmental anlaysis. Few of the examples of conversion from words to text are presented in table.

**Slide 12:** Then, the 2nd process in NLP is Prosody generation. Prosody actually means the melody of the spoken form.The prosody generation mainly includes factors such as intonation modelling like phrasing & accent in speech, amplitude modelling & duration modelling means duration of sound & pauses in speech as well intensity of sound. Hence, obviously naturalness of TTS system is mostly depend on prosody generation. This process is also known as supra-segmental analysis.

**Slide 13:** Now, these phonetic transcriptions & prosody information together makes the symbolic linguistic representation which is then passed onto the Digital Signal Processing block as input. This block is also known as back-end part where the symbolic linguistic representation gets converted into actual sound that can be heard by speech synthesis process. Speech synthesis process basically means to produce Human speech artificially. Various speech synthesis techniques & algorithms are used to get synthesized speech. The one which is technically advanced, currently popular & used in TTS modules that we applied is **HMM(Hidden Markov Model) based speech synthesis technique** also known as **HTS** has been used here. The process works in 2 main parts as training phase & synthesize phase.

**Slide 14:** Now, this is the overview or Architecture of HMM based speech synthesis technique. This synthesis technique is actually divided into 2 parts as I said. The 1st part is training phase where excitation parameter related to fundamental frequency f0 vocal chord, intensity & then spectral parameter about frequency spectrum, vocal tract gets extracted from defined speech database. These both corresponds to individual sounds as labels & with duration of speech together, we develop HMM parameter estimation model.

Then, in the 2nd step during synthesis, text or symbolic linguistic representation is analysed & contextual labelling is done. Then depending on that text, we generate parameters about which excitation & spectral parameter is best match for the inputted text. Once we get both the parameters, standard parametric synthesis filter is used which will give us final output as a spoken voice.

**Slide 15:** How Does normal Translation Work?

If you want to translate a sentence in another language one way of doing so is by word-to-word translation, but we all know that any language consists of two things spelling or words and grammar. These two components make the translation difficult to execute.

**Slide 16:** So how does translation work?

Translation of a language is done by passing a sentence through a neural network. It converts the language into digital signal and processes it to output another language.

**Slide 17:** What are Neural networks?

If we define it mathematically then it is a differentiable function which maps a set of set variables to another variable. These neural networks consist of machine learning models which helps in assigning variables to a desired output.

In this case a sentence is passed through a neural network which converts the sentences into digits like vectors and matrices. Now this data is used by the neural network which consists of thousands of data samples to determine the right sequence of the translated sentence. This particular neural network is called as the encoder-decoder architecture which decodes a sequence processes it and encode the sequences into another language.

**Slide 18:** Neural Network Architecture

The NMT (Neural Transmission Model) consists of 3 parts an encoder, decoder and an attention mechanism. Encoder and decoder use lstm’s (Long Short-term memory networks) which predicts the word and assigns a value to it which is decoded back to a sentence. Now the attention mechanism works on the grammar part of the sentences. It correlates the input and samples of output word-by-word as the meaning of a sentences is determined by words coming before and after a certain word.

**Slide 19:** How Google Trans API work?

Google trans uses the neural network in a scaled-up version where instead of using 1 lstm to encode and decode the sentence it uses 8 lstm to figure out the correct translation of the word.