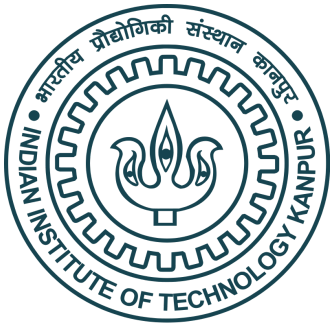
# INDIAN INSTITUTE OF TECHNOLOGY KANPUR

## COMPUTATIONAL LINGUISTICS FOR INDIAN LANGUAGES

## CS-689A

PROJECT REPORT

# TEXT-TO-SPEECH SYSTEM




**SUPERVISED BY**

Prof. Arnab Bhattacharya

**SUBMITTED BY**

Drashtant Singh Rathod

Sudiksha Navik

# TABLE OF CONTENTS

# ABSTRACT

Text-to-speech (TTS) synthesis is one of the rapidly emerging areas of computer-to-human interaction technology. Human-like speech is replicated by the computer with the introduction of input text which is usually very natural. Real-life applications of TTS synthesis technique make users task hassle-free. For example, reading books for the visually impaired people, paying electricity bills through automated call-centers, announcing train information at the railway station, etc. Traditionally, rule-based speech synthesis methods are deployed which find difficulties in obtaining optimal rules, resulting in lack of naturalness in the generated synthesized speech. Alternatively, to meet the desired quality of experience (QoE) of users while using these applications, this paper designs and develops a simple and robust TTS synthesis system for English language using the concatenative speech synthesis method and its variants and finds its suitability in intelligible and/or natural speech production. Various steps involved in processing text for speech production through the TTS system are described. Results demonstrate that the speech generated by the concatenative speech synthesis method, in particular, unit-selection technique is smoother and natural, sounding like a human voice. This is supported by an informal listening test of the generated synthesized speech.

# INTRODUCTION

One of the simplest and effective means of communication between human to human, human to machine and vice-versa is “Speech”. In order to interact from human to machine, speech recognition techniques are used efficiently. However, for communicating from machine to human, speech synthesis (SS) techniques are exploited. Speech synthesis is the technique in which the replica of a speech signal is created for transmitting a message from the machine to the human. In order words, speech synthesis is the artificial production of human speech. The objective of SS is to obtain a synthesized speech which is easily understandable, indistinguishable, and natural from that produced by a human.

Speech synthesis is referred to as text-to-speech (TTS) synthesis as the machine converts text into speech by the application of various natural language processing (NLP) and digital signal processing (DSP) techniques as shown in Fig. 1. It is fairly easy to say that speech synthesis is a computer talking rather than a human”. TTS synthesis systems have wide varieties of applications in real-life. For example, reading textbooks for the visually impaired people, paying electricity bills or booking travel by calling the automated call-center, handling the entire transaction through an automatic dialogue system, making announcements on the railway platforms or airports via automatic dialogue system, etc. These practical applications make the task easy, save the human time, and also save the money of the employer needed to hire employees for performing these tasks in person.

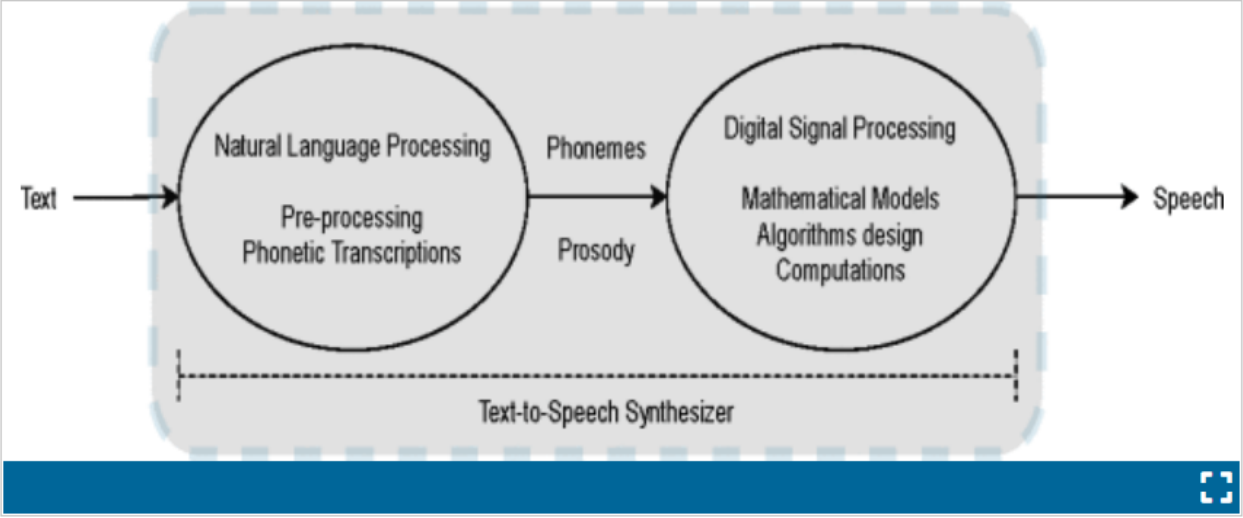



**Fig. 1**
Simple functionality of TTS synthesis system [3].

The TTS synthesis system comprises mainly two steps as shown in Fig. 1. In the first step, the input text is pre-processed, analyzed and then transcribed into phonetic or linguistic representation using NLP techniques. In the second step, synthesized speech in audio format is generated from the phonetic and prosodic information of the text using DSP techniques which involve mathematical models. The complete system is called " TTS Synthesizer" or " TTS Engine". The naturalness and intelligibility[1] of the artificially produced speech are maintained such that the synthesized speech would not be like a mechanical sounding voice which is irritating and discomforting to the listener, rather a human-like voice which is pleasing and comfortable in human listening.

A range of data-driven text-to-speech synthesis methods are developed in literature. For example, Wave-Tacotron generates speech waveforms from the input text files. It is based on training a deep neural network (DNN) on a sequence-to-sequence basis in order to learn all the features/parameters. In, a DNN based speech synthesis system is proposed which maps words into real numbers in order to train the DNN. Generative adversarial networks (GANs) based speech synthesis systems are developed. A speech synthesis model

combining linear prediction with recurrent neural network (RNN), known as LPCNet. However, the data-driven speech synthesis systems need a large amount of data for training and testing DNN/RNN/GANs due to their data hungry nature. It also requires a high performance laptop or computer with highly configured/powerful GPUs or super-computing servers, resulting in high computational complexity and a large amount of carbon emission.

A simple, and robust TTS synthesis system is needed for efficient conversion of text into speech. Along this line, this paper explores the concatenative speech synthesis and its variants in order to design and develop a robust TTS synthesis system in which the generated speech sounds intelligible and natural. Depending on the applications, the TTS synthesis system can be exploited to fulfill the users quality of experience (QoE).

# BACKGROUND

Speech synthesis is the speech encoding by the machine (e.g., computer). In literature, there are mainly three types of speech synthesis methods by which the TTS synthesis systems can be designed and developed as shown in Fig. 2. It includes namely; Formant, Concatenative and Articulatory synthesis. This section discusses these speech synthesis techniques with their limitations or drawbacks.

## A. Formant Synthesis

Formant synthesis (FS) is one of the rule-based speech synthesis methods in which the speech production rules are determined. The resonance frequencies of the vocal-tract are called “formants”. In FS, the speech is synthesized using estimated frequencies of speech signal. In other words, speech production has a source-filter model. The source produces an excitation that passes through a filter in order to model the vocal-tract. The excitation source can be either an impulse train which represents voiced sound or random noise which represents unvoiced sound. The resonances of vocal-tract modify the excitation source for producing the synthesized speech in FS. The main drawback of FS lies in obtaining optimal rules of evolution of formants. As a result, the synthesized speech lacks naturalness. However, the synthesized speech is quite intelligible. Moreover, this technique does not require a database of speech samples, therefore, the memory and processing costs are saved.

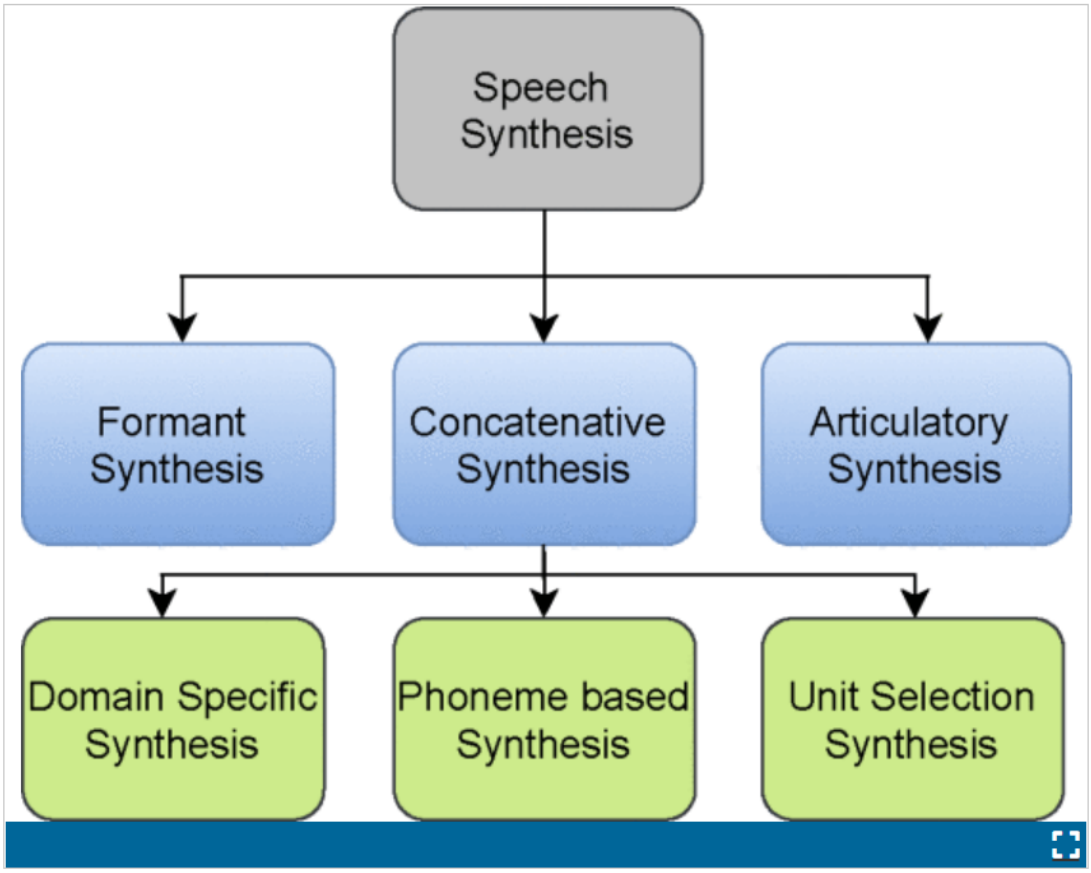



**Fig. 2**
Taxonomy of speech synthesis method.

## B. Articulatory Synthesis

Articulatory synthesis (AS) is another rule-based speech synthesis method. This method generates synthesized speech by direct modeling of human articulators such as lips, tongue, velum, vocal cord, nose, etc. To describe the articulatory motions, a sufficient number of parameters are needed. The drawback of this method is that one can find difficulties in acquiring data to determine various rules, making it computationally intractable. Moreover, it also requires high cost, therefore, impractical for real-life speech synthesis system modeling.

## C. Concatenative Synthesis

Concatenative synthesis (CS) is a widely used speech synthesis method to produce artificial speech as the synthesized speech produced by this method sounds natural. It does not perform explicit vocal-tract modeling, that is, it is not a rule-based speech synthesis method. Instead, it acquires an exhaustive set of units[3] from the pre-recorded speech and then performs synthesis by concatenating these units in appropriate order. DSP techniques are exploited to smoothen the concatenating boundaries in order to construct a complete natural and intelligible utterance. Since all speech units are stored in a database collectively, therefore, this method is also called "Corpus-based speech synthesis". The length of the units affects the intelligibility and naturalness of synthesized speech. For example, longer units increase the naturalness of synthesized speech as it needs less number of concatenation points. However, shorter units need more concatenation points, therefore, it may increase the complexity and not produce natural and intelligible synthesized speech. In order to facilitate the smooth production of synthesized speech depending on the choice of various tasks using the concatenative synthesis method, it is divided into three sub-categories.

1. **Domain-specific Synthesis:** In this type of concatenative technique, the complete synthesized utterance is generated by concatenating the pre-recorded words, phrases, and sentences. It is used in applications where the domain is limited (domain-specific), that is, the text conversions in a variety of cases are repetitive. For example, talking clocks, calculators, railway transit announcements, weather reports, etc. Due to the limited varieties of words, phrases, and sentences present in the database, the generated synthesized utterance closely matches with the original utterance having natural voice. This makes it a quite simple technique but it is not a general-purpose speech synthesis technique due to the lack of use in diverse fields.

2. **Phoneme-based Synthesis:** In this type of concatenative technique, the input text is considered as only syllable word in order to produce natural synthesized speech. Firstly, the input word is given from the keyword of the computer. Then, it is converted into corresponding phonetic transcription which is called " grapheme"-to-phoneme conversion. The grapheme-to-phoneme conversion is done by a Viterbi module and a tree structure is deployed in order to store the aligned grapheme-phoneme pairs [19]. Further, a dictionary-based approach is applied in order to determine the correct word from the dictionary. Further, the phoneme sounds are concatenated according to phonetic transcriptions of words to produce synthesized speech. The flowchart of the phoneme-based speech synthesis is shown in Fig. 3.

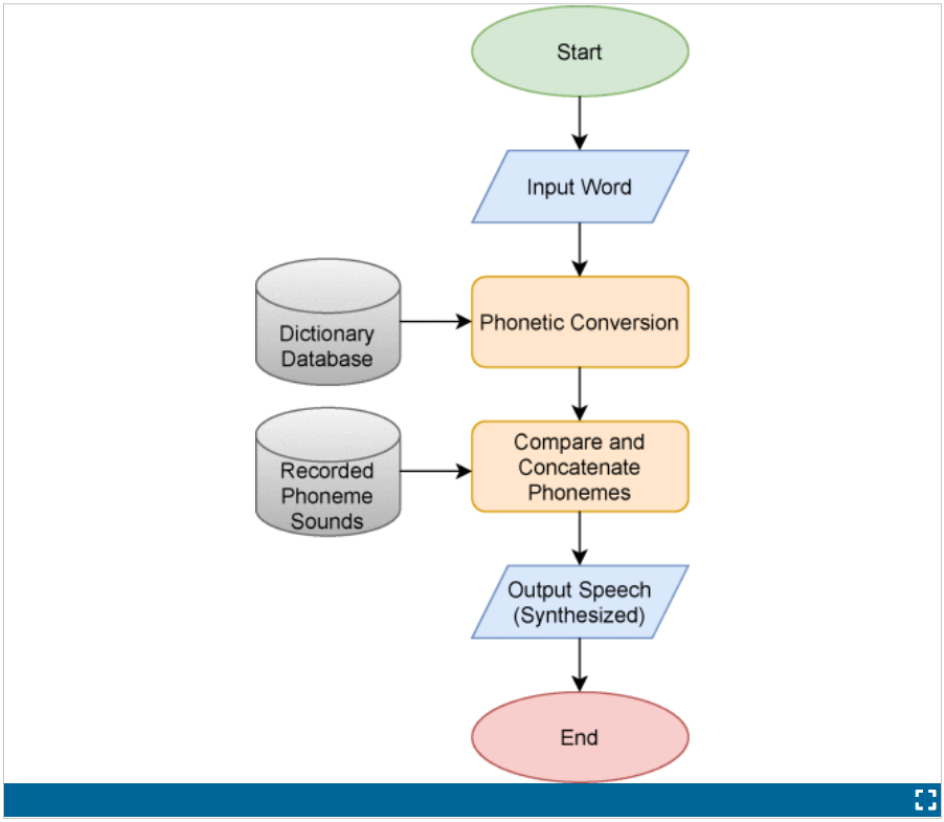



**Fig. 3**
Flowchart for conversion of text-to-speech using phoneme-based speech synthesis.

3. **Unit-selection Synthesis:** It is the most preferred and dominant concatenative synthesis technique due to the high level of naturalness of the generated synthesized speech, which is often indistinguishable from the original speaker voice. It stores multiple instances of each unit in a large speech database, and then the unit selection algorithm (e.g., Viterbi) is exploited to select the most appropriate unit from the database that matches with the target unit, followed by concatenating units to obtain synthesized speech. Two cost functions, namely; target cost and joint cost are jointly minimized in order to select the best unit. Target cost estimates the similarity between the features of the database unit and target unit. Join cost measures the wellness of joining and matching two speech units (database and target unit). The advantage of this technique is that it requires either minimum or no DSP techniques to the recorded database. However, it requires a good amount of memory for storing the database.

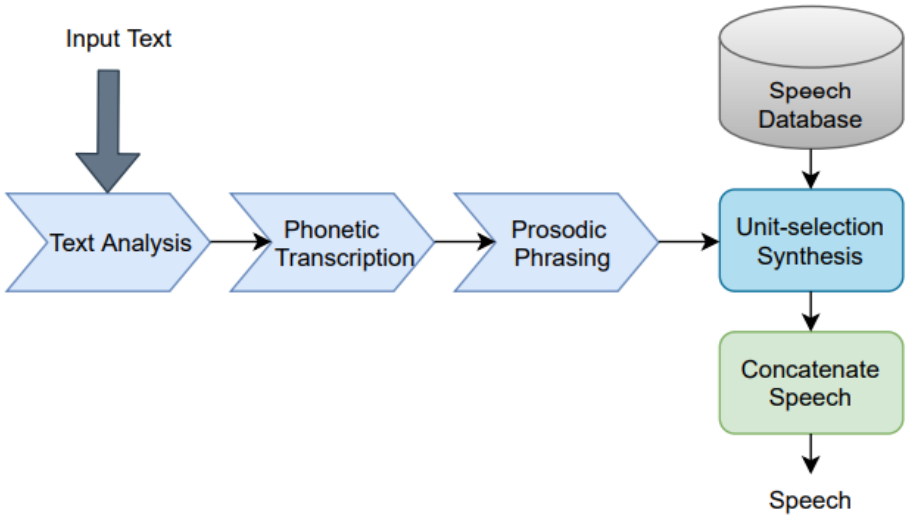



Fig. 4: Block diagram of unit-selection speech synthesis.

# TEXT-TO-SPEECH CONVERSION ALGORITHM

In order to produce smooth and natural synthesized speech from the text using the unit-selection algorithm, we follow the following strategy.

- The input text to be converted into synthesized speech is pre-processed through the NLP block prior to injecting into the algorithm.

- The text is segmented into tokens (known as tokenization process) and then each part is processed in order to determine the best tags for the words.

- The previous step is followed by phonetic conversion using the available dictionary or speech database.

- Prosodic phrasing is performed in which the phrases are assigned to the text using appropriate amplitude modeling, intonation modeling, and duration modeling. All phonetic transcriptions and prosodic information together make up symbolic linguistic representation.

- Backend process begins with the synthesis of linguistic information using a concatenation synthesis method. Here, the input text is matched with the recorded database.

- The output speech waveform is generated by concatenating the speech segments or units.

The algorithm in step-by-step order is as follows:

- Step1: Start
- Step2: Input text
- Step3: Pre-process text
- Step4: Segment text into tokens
- Step5: Convert text into phonemes
- Step6: Phrase prosodic
- Step7: Match text with database
- Step8: Concatenate text/units
- Step9: Show waveform
- Step10: End

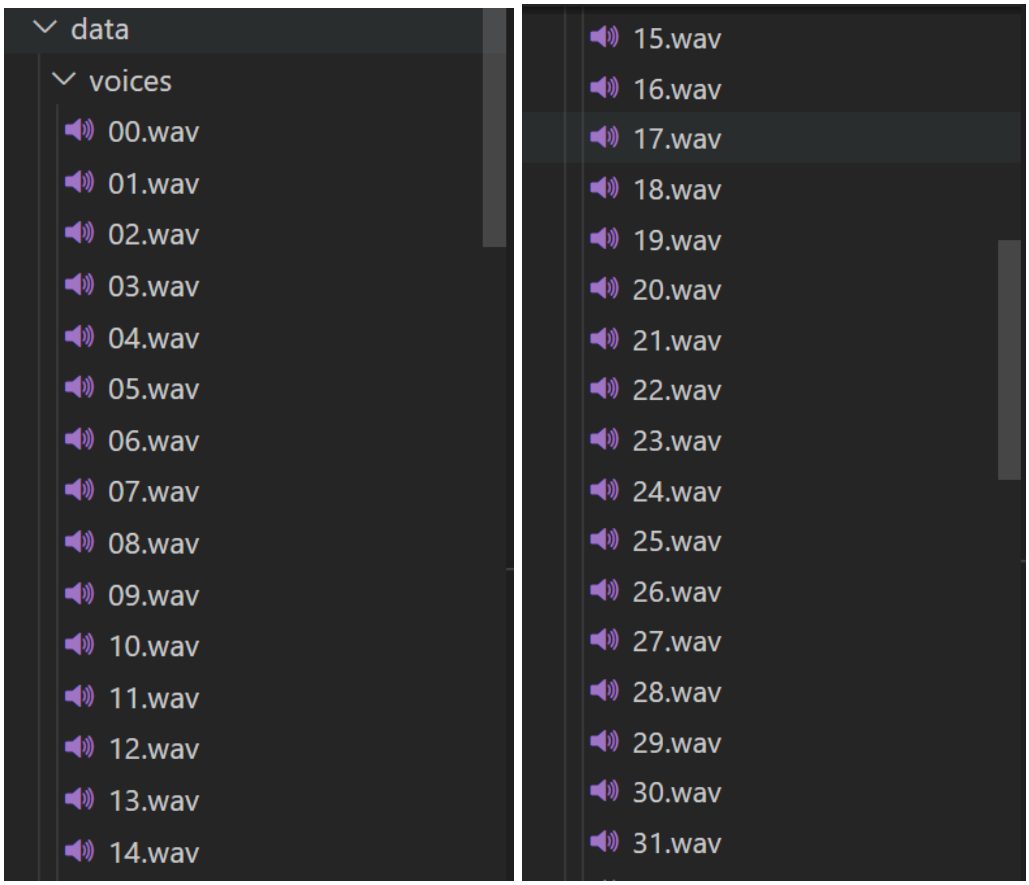
| Vowels | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| IPA | Hindi | | ISO 15919 | Urdu[10] | | | Approxi. English equivalent |
| | Initial | Final | | Final | Medial | Initial | |
| ə[11] | अ | | a | ـَ | -َ- | اَ | about |
| aː | आ | ा | ā | ـا | | آ | far |
| ɪ | इ | ि | i | ی | -ِ- | اِ | still |
| iː | ई | ी | ī | | ـِيـ | اِيـ | fee |
| ʊ | उ | ु | u | ـو | -ُ- | اُ | book |
| uː | ऊ | ू | ū | ـُو | | اُو | moon |
| eː | ए | े | ē | ـے | ـيـ | ايـ | mate |
| ɛː | ऐ | ै | ai | ـَے | ـَيـ | اَيـ | fairy |
| oː | ओ | ो | ō | ـو | | او | force |
| ɔː | औ | ौ | au | ـَو | | اَو | lot (Received Pronunciation) |
| h[12] | | | h | ھ | | | (Aspirated sounds) cake |
| ◌̃[13] | | ◌ँ | m̐ | ں | ـںـ | ںـ | nasal vowel faun ([ãː, õː], etc.) |
| | | ◌ं | ṁ | | | | jungle |

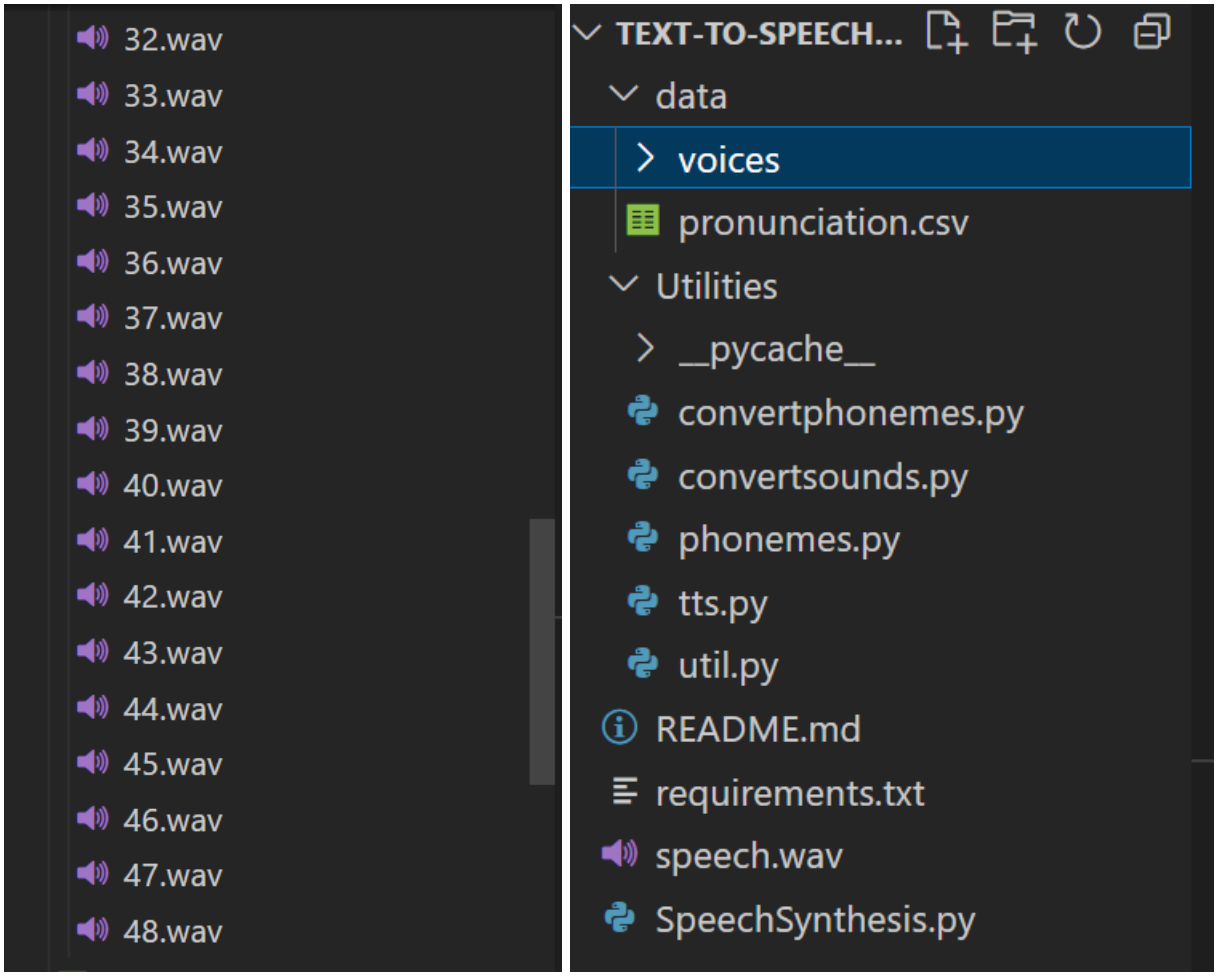
**Consonant phonemes of Hindustani**

| | | Labial | Dental/ Alveolar | Retroflex | Post-alv./ Palatal | Velar | Uvular | Glottal |
|---|---|---|---|---|---|---|---|---|
| **Nasal** | | m | n | (ɳ) | (ɲ) | ŋ | | |
| **Stop/ Affricate** | **voiceless** | p | t | ʈ | t͡ʃ | k | (q) | |
| | **voiceless aspirated** | pʰ | tʰ | ʈʰ | t͡ʃʰ | kʰ | | |
| | **voiced** | b | d | ɖ | d͡ʒ | g | | |
| | **voiced aspirated** | bʱ | dʱ | ɖʱ | d͡ʒʱ | gʱ | | |
| **Fricative** | **voiceless** | f | s | (ʂ) | (ʃ) | (x) | | ɦ |
| | **voiced** | ʋ | z | | (ʒ) | (ɣ) | | |
| **Approximant** | | | l | | j | | | |
| **Tap/Trill** | **unaspirated** | | r | ɽ | | | | |
| | **aspirated** | | | ɽʱ | | | | |

# DATABASE PREPARATION

In order to have appropriate conversion of text into an acoustic waveform which is indistinguishable from the human speech, a database containing different acoustic units of recorded speech is prepared. The database contains different units, for example, words, syllables, half-syllables, phonemes (in our case), phones, diphones, triphones, sentences, etc. It comprises units of utterances from male and female speakers in the Hindi language. To make it more contiguous, words and numbers of sizes one, two, three, four, five and long utterances are recorded. The recordings are carried out in multiple sessions and the recording conditions are maintained in each session in order to avoid spectral or amplitude discontinuity.




32.wav
33.wav
34.wav
35.wav
36.wav
37.wav
38.wav
39.wav
40.wav
41.wav
42.wav
43.wav
44.wav
45.wav
46.wav
47.wav
48.wav
TEXT-TO-SPEECH...
data
voices
pronunciation.csv
Utilities
__pycache__
convertphonemes.py
convertsounds.py
phonemes.py
tts.py
util.py
README.md
requirements.txt
speech.wav
SpeechSynthesis.py

# EVALUATION METHODOLOGY

Along the line of the performance evaluation of the TTS synthesis system, which is highly dependent on two key attributes namely; intelligibility and naturalness, we design experimental setup for each type of concatenative speech synthesis technique individually. We perform an informal listening test of the synthesized speech in which we listen to the generated synthesized speech and then analyze its intelligibility and naturalness. Depending on the application at hand or user requirements, a particular speech synthesis method may be more suitable than another one. For example, in some cases, the message may be fully understood by the listener but they may not be able to recognise each single word. In this case, the synthesized speech is not natural but intelligible. Similarly, in normal conversation, we often miss some parts but we understand the “gist” of the message. However, in the case of reading addresses in the telephony system or reading books for the visually impaired people, it is necessary that the TTS synthesis system should read the content very accurately. In this case, the synthesized speech is natural but not intelligible.

# RESULTS AND DISCUSSIONS

It can be seen that the output speech is quite smooth and natural, which is also validated by the informal listening test.

*"Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware"*. The TTS engine is successfully tested to convert a text into speech. The analysis exhibits the variation pauses in speech sound and the concatenation of phonemes from the phoneme library. Playing the generated audio file sounds like a natural speech.