Text-to-speech System

SUDIKSHA NAVIK (22111059)

DRASHTANT SINGH RATHOD (22111021)

What is a TTS System?

- Text to speech (TTS) is a natural language modeling process that requires changing units of text into units of speech for audio presentation. This is the opposite of speech to text, where a technology takes in spoken words and tries to accurately record them as text.
- It is **a type of assistive technology that reads digital text aloud**. It's sometimes called “read aloud” technology. With a click of a button or the touch of a finger, TTS can take words on a computer or other digital device and convert them into audio.

Text-to-Speech Synthesizer

- 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”.

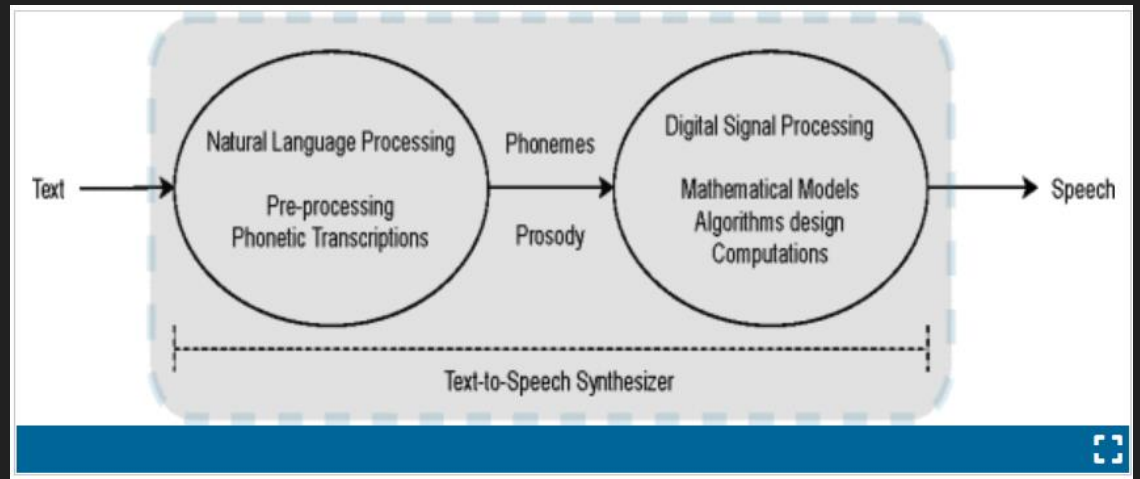

Fig. 1
Simple functionality of TTS synthesis system [3].

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 from the pre-recorded speech and then performs synthesis by concatenating these units in appropriate order.

Types of Concatenative Synthesis

- Domain-specific Synthesis
- Phoneme-based Synthesis
- Unit-selection Synthesis

Domain-Based Concatenative 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.

Phoneme-Based Concatenative 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.
- 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.

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.

Algorithm

- There are three steps:
- 1. Text-to-words:
  - Raw input text is tokenized into a list of words. This will also generally include converting numerical digits into their word equivalents (ex: turn "6" into "six").
- 2. Words-to-phonemes:
  - The array of words is converted into phonemes.
  - Phonemes are the individual sounds in a language. As Hindi has a pretty vast phonetic genre, the hindi alphabetic pronunciation can vary change the pronunciation of the whole word.
  - The system has already mapped the hindi phonetic sounds to their alphabets, so whenever the alphabet is detected, the system just maps to its audio file and return its number. The output is an list of numbers that each correspond to one of the 44 hindi phonemes.
- 3. Phonemes-to-sounds:
  - Each phoneme is paired with an audio file. This is the point where the actual audio is stitched together. It would also be in this step that the correct voice for the audio is selected, assuming multiple voices are supported.

Implementation

- phoneme.py:
  - Contains the list of all sounds used in the conversion.
- convertphonemes.py:
  - Takes list of words and finds appropriate phonemes
- convertsounds.py:
  - Matches each phoneme from list with appropriate wav file
- tts.py:
  - Pulls all three pieces together: words, phonemes and sounds

Result

- An output wav file named “speech.wav”.

Thank You