**DESIGN DOCUMENT**

SOFTWARE ENGINEERING PROJECT

CSE 325

TEXT-TO-SPEECH APPLICATION

**PROF. LYDIA JANE**

**Submitted by:**

**Ishan Krishna 12BCE0501**

**Shobhit Gupta 12BCE0503**

**Slot: B1**

**PURPOSE**

A Text-to-speech synthesizer is an application that converts text into spoken word, by analyzing and processing the text using Natural Language Processing (NLP) and then using Digital Signal Processing (DSP) technology to convert this processed text into synthesized speech representation of the text. Here, we are developing a useful text-to-speech synthesizer in the form of a simple application that converts inputted text into synthesized speech and reads out to the user. The development of a text to speech synthesizer will be of great help to people with visual impairment and make making through large volume of text easier.

Also, we are developing a speech-to-text application which will convert the input speech into written text and display it to the user. This will be of great help to people with hearing disabilities. The converted text can be saved in the form of a word document which can be later used for reference purposes.

**DESIGN OUTLINE**

Text-to-speech synthesis -TTS - is the automatic conversion of a text into speech that resembles, as closely as possible, a native speaker of the language reading that text. Text-to-speech synthesizer (TTS) is the technology which lets computer speak to you. The TTS system gets the text as the input and then a computer algorithm which called TTS engine analyses the text, pre-processes the text and synthesizes the speech with some mathematical models. The TTS engine usually generates sound data in an audio format as the output.



Fig. 1: Functional diagram of a TTS

Speech synthesis can be described as 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. A text-to-speech (TTS) system converts normal language text into speech. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output. The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer.

A text-to-speech system (or "engine") is composed of two parts: **a front-end and a back-end**. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the synthesizer—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

There are different ways to perform speech synthesis. The choice depends on the task they are used for, but the most widely used method is Concatentive Synthesis, because it generally produces the most natural-sounding synthesized speech. Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. There are three major sub-types of concatenative synthesis:

**Domain-specific Synthesis:** Domain-specific synthesis concatenates pre-recorded words and phrases to create complete utterances. It is used in applications where the variety of texts the system will output is limited to a particular domain, like transit schedule announcements or weather reports. The technology is very simple to implement, and has been in commercial use for a long time, in devices like talking clocks and calculators. The level of naturalness of these systems can be very high because the variety of sentence types is limited, and they closely match the prosody and intonation of the original recordings. Because these systems are limited by the words and phrases in their databases, they are not general-purpose and can only synthesize the combinations of words and phrases with which they have been pre-programmed. The blending of words within naturally spoken language however can still cause problems unless many variations are taken into account. For example, in non-rhotic dialects of English the "r" in words like "clear" /ˈklɪə/ is usually only pronounced when the following word has a vowel as its first letter (e.g. "clear out" is realized as /ˌklɪəɾˈʌʊt/). Likewise in French, many final consonants become no longer silent if followed by a word that begins with a vowel, an effect called liaison. This alternation cannot be reproduced by a simple word-concatenation system, which would require additional complexity to be context-sensitive. This involves recording the voice of a person speaking the desired words and phrases. This is useful if only the restricted volume of phrases and sentences is used and the variety of texts the system will output is limited to a particular domain e.g. a message in a train station, whether reports or checking a telephone subscriber’s account balance. .

**Unit Selection Synthesis:** Unit selection synthesis uses large databases of recorded speech. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, diphones, half-phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a "forced alignment" mode with some manual correction afterward, using visual representations such as the waveform and spectrogram. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighbouring phones. At runtime, the desired target utterance is created by determining the best chain of candidate units from the database (unit selection). This process is typically achieved using a specially weighted decision tree.

Unit selection provides the greatest naturalness, because it applies only a small amount of digital signals processing (DSP) to the recorded speech. DSP often makes recorded speech sound less natural, although some systems use a small amount of signal processing at the point of concatenation to smooth the waveform. The output from the best unit-selection systems is often indistinguishable from real human voices, especially in contexts for which the TTS system has been tuned. However, maximum naturalness typically require unit-selection speech databases to be very large, in some systems ranging into the gigabytes of recorded data, representing dozens of hours of speech. Also, unit selection algorithms have been known to select segments from a place that results in less than ideal synthesis (e.g. minor words become unclear) even when a better choice exists in the database.

**STRUCTURE OF A TEXT-TO-SPEECH**

Text-to-speech synthesis takes place in several steps. The TTS systems get a text as input, which it first must analyze and then transform into a phonetic description. Then in a further step it generates the prosody. From the information now available, it can produce a speech signal.

The structure of the text-to-speech synthesizer can be broken down into major modules:

* **Natural Language Processing (NLP) module:** It produces a phonetic transcription of the text read, together with prosody.
* **Digital Signal Processing (DSP) module:** It transforms the symbolic information it receives from NLP into audible and intelligible speech.

The major operations of the NLP module are as follows:

* **Text Analysis:** First the text is segmented into tokens. The token-to-word conversion creates the orthographic form of the token. For the token “Mr” the orthographic form “Mister” is formed by expansion, the token “12” gets the orthographic form “twelve” and “1997” is transformed to “nineteen ninety seven”.
* **Application of Pronunciation Rules:** After the text analysis has been completed, pronunciation rules can be applied. Letters cannot be transformed 1:1 into phonemes because correspondence is not always parallel. In certain environments, a single letter can correspond to either no phoneme (for example, “h” in “caught”) or several phoneme (“m” in “Maximum”). In addition, several letters can correspond to a single phoneme (“ch” in “rich”). There are two strategies to determine pronunciation:
* In dictionary-based solution with morphological components, as many morphemes (words) as possible are stored in a dictionary. Full forms are generated by means of inflection, derivation and composition rules. Alternatively, a full form dictionary is used in which all possible word forms are stored. Pronunciation rules determine the pronunciation of words not found in the dictionary.
* In a rule based solution, pronunciation rules are generated from the phonological knowledge of dictionaries. Only words whose pronunciation is a complete exception are included in the dictionary.

The two applications differ significantly in the size of their dictionaries. The dictionary-based solution is many times larger than the rules-based solution’s dictionary of exception. However, dictionary-based solutions can be more exact than rule-based solution if they have a large enough phonetic dictionary available.

**Prosody Generation:** After the pronunciation has been determined, the prosody is generated. The degree of naturalness of a TTS system is dependent on prosodic factors like intonation modelling (phrasing and accentuation), amplitude modelling and duration modelling (including the duration of sound and the duration of pauses, which determines the length of the syllable and the tempos of the speech).

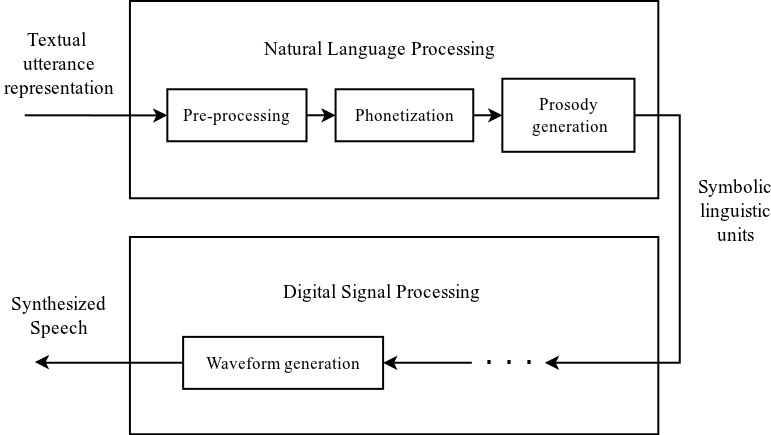


Fig. 2: Simple architecture of NLP and DSP



Fig. 3: Operations of the natural Language processing module of a TTS synthesizer.

The output of the NLP module is passed to the DSP module. This is where the actual synthesis of the speech signal happens. In concatenative synthesis the selection and linking of speech segments take place. For individual sounds the best option (where several appropriate options are available) are selected from a database and concatenated.



Fig. 4: The DSP component of a general concatenation-based synthesizer.