**Text To Speech**

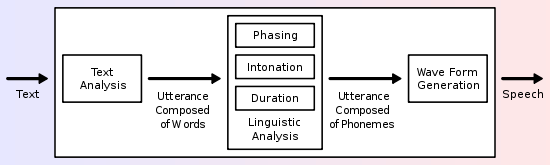
**Abstract:**

Speech synthesis is the artificial production of human [speech](https://en.wikipedia.org/wiki/Speech). A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in [software](https://en.wikipedia.org/wiki/Software) or [hardware](https://en.wikipedia.org/wiki/Computer_hardware) products. A text-to-speech (TTS) system converts normal language text into speech; other systems render [symbolic linguistic representations](https://en.wikipedia.org/wiki/Symbolic_linguistic_representation) like [phonetic transcriptions](https://en.wikipedia.org/wiki/Phonetic_transcription) into speech. Text to speech finds its applications in various fields which are discussed below in the document.

**Introduction:**

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a [database](https://en.wikipedia.org/wiki/Database). Systems differ in the size of the stored speech units; a system that stores [phones](https://en.wikipedia.org/wiki/Phone_(phonetics)) or [diaphones](https://en.wikipedia.org/wiki/Diphone) 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](https://en.wikipedia.org/wiki/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 clearly. An intelligible text-to-speech program allows people with [visual impairments](https://en.wikipedia.org/wiki/Visual_impairment) or [reading disabilities](https://en.wikipedia.org/wiki/Reading_disability) to listen to written works on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s.

[](https://en.wikipedia.org/wiki/File:TTS_System.svg)

Overview of a typical TTS system

A text-to-speech system (or "engine") is composed of two parts:[[3]](https://en.wikipedia.org/wiki/Speech_synthesis" \l "cite_note-3) a [front-end](https://en.wikipedia.org/wiki/Front_end_processor_(program)) and a[back-end](https://en.wikipedia.org/wiki/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*](https://en.wikipedia.org/wiki/Tokenization_(lexical_analysis)). The front-end then assigns [phonetic transcriptions](https://en.wikipedia.org/wiki/Phonetic_transcription) to each word, and divides and marks the text into [prosodic units](https://en.wikipedia.org/wiki/Prosody_(linguistics)), like [phrases](https://en.wikipedia.org/wiki/Phrase), [clauses](https://en.wikipedia.org/wiki/Clause), and [sentences](https://en.wikipedia.org/wiki/Sentence_(linguistics)). The process of assigning phonetic transcriptions to words is called *text-to-phoneme* or [*grapheme*](https://en.wikipedia.org/wiki/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.

**Literature Survey:**

We used 2 IEEE papers for our reference:

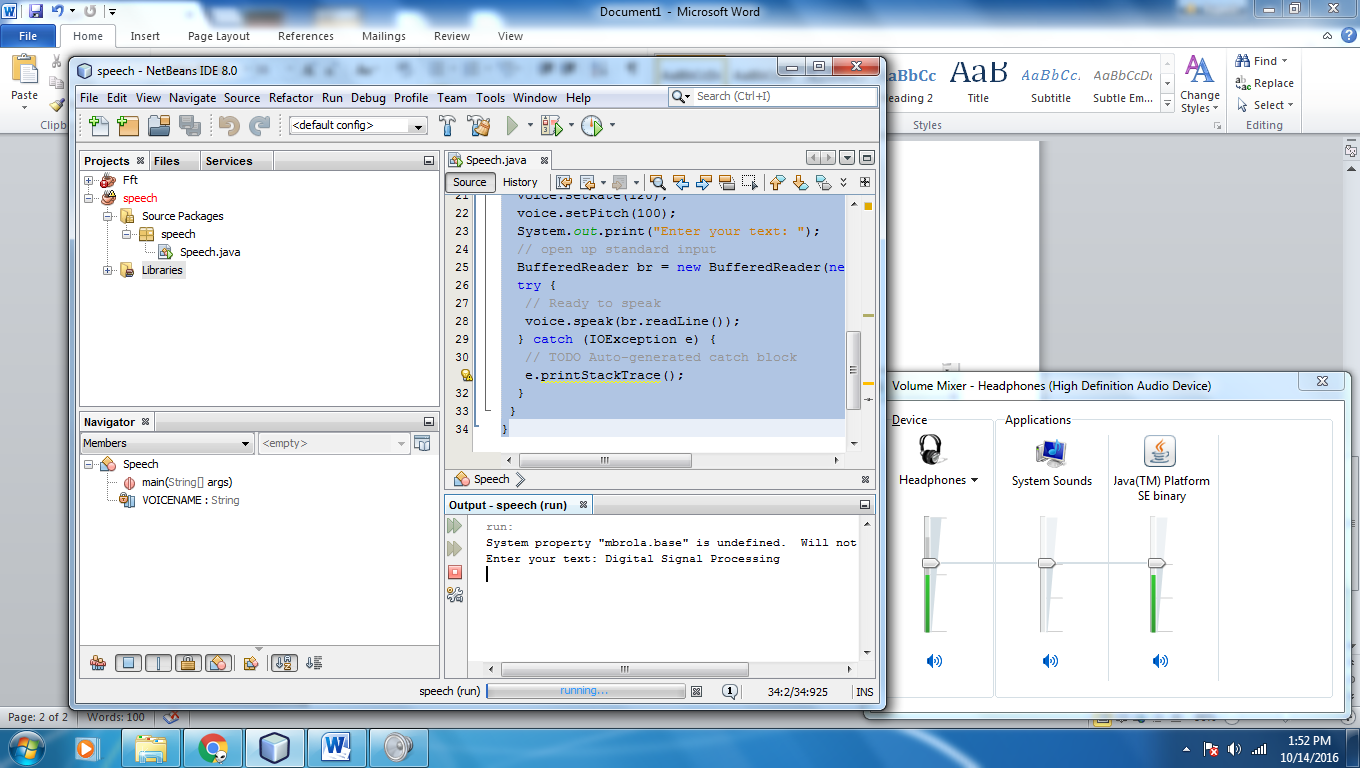
1. A Mapper and Combiner based Marathi Text to speech synthesis using English TTS Engine

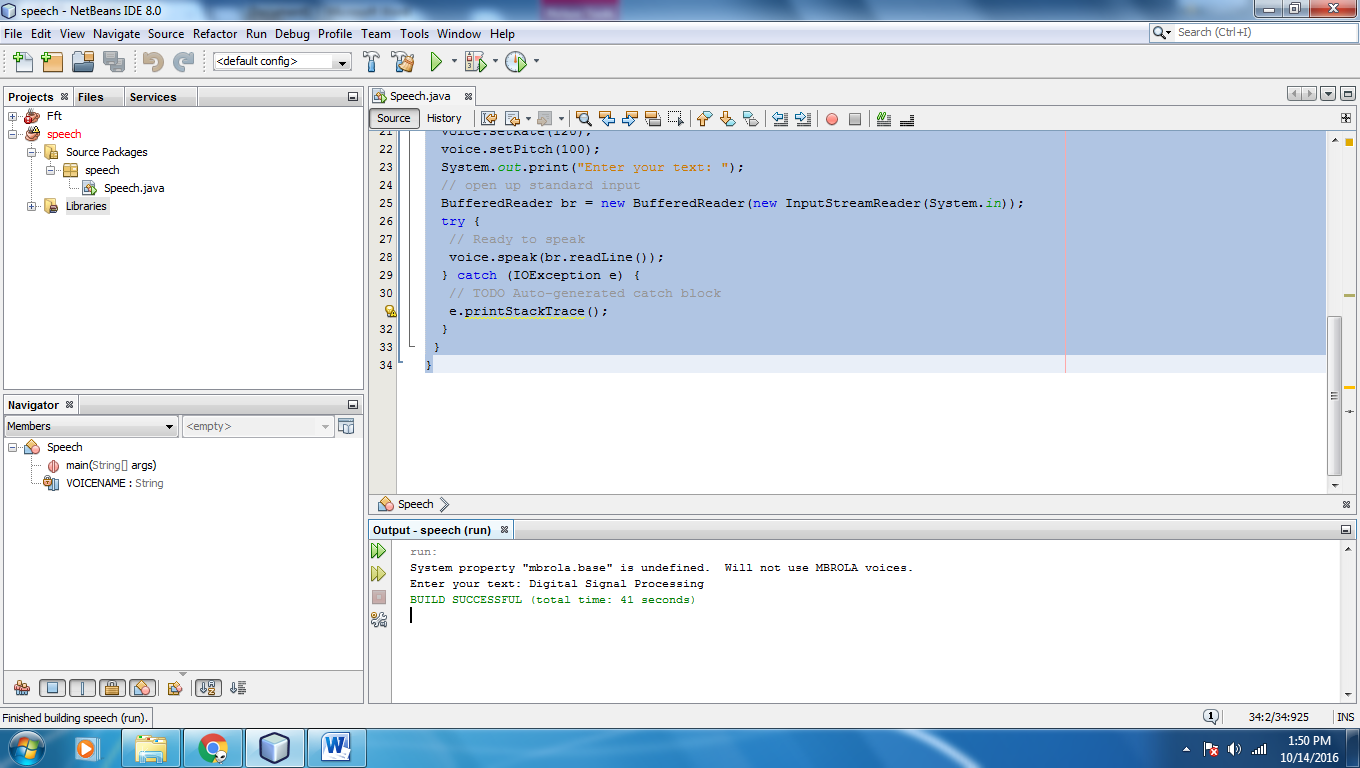
* The first paper concentrated more on improving the efficiency of the traditional TTS systems where the major drawbacks included glitches, reverberations, spectral mismatch and requirements for huge databases. We thus implemented a system with fewer drawbacks using the influences from first paper.
* It curbed the glitches, reverberations, spectral mismatch etc.
* The basic structure of our system was inspired by the structure of the system proposed in the above paper, which are: text processing and speech generation.

1. Enhanced synthesized text reader for visually impaired users

* The basic functions of the system are reading arbitrary files, converting text into speech using different synthesized voices and spatializing synthesized speech.
* The core of the system is based on Java platform using FreeTTS speech synthesizer.
* The main part of our work is the development of an extension to the FreeTTS software.
* The input text is read from an arbitrary file, which, however, has to be pre-tagged with the information on the abovementioned voice parameters.
* One possible application of such systems is an enhanced version of the electronic book reader, which can use various voices with different characteristics that are located at different spatial positions.
* In this way, each voice can, for example, represent a different character in the story or in the play.

**Result:**





**Applications:**

Speech synthesis has long been a vital [assistive technology](https://en.wikipedia.org/wiki/Assistive_technology) tool and its application in this area is significant and widespread. It allows environmental barriers to be removed for people with a wide range of disabilities. The longest application has been in the use of [screen readers](https://en.wikipedia.org/wiki/Screen_reader) for people with [visual impairment](https://en.wikipedia.org/wiki/Visual_impairment), but text-to-speech systems are now commonly used by people with [dyslexia](https://en.wikipedia.org/wiki/Dyslexia) and other reading difficulties as well as by pre-literate children. They are also frequently employed to aid those with severe [speech impairment](https://en.wikipedia.org/wiki/Speech_impairment) usually through a dedicated [voice output communication aid](https://en.wikipedia.org/wiki/Voice_output_communication_aid).

Speech synthesis techniques are also used in entertainment productions such as games and animations. In 2007, Animo Limited announced the development of a software application package based on its speech synthesis software Fine Speech, explicitly geared towards customers in the entertainment industries, able to generate narration and lines of dialogue according to user specifications. The application reached maturity in 2008, when NEC [Biglobe](https://en.wikipedia.org/wiki/Biglobe) announced a web service that allows users to create phrases from the voices of [Code Geass: Lelouch of the Rebellion R2](https://en.wikipedia.org/wiki/Code_Geass:_Lelouch_of_the_Rebellion_R2) characters.

In recent years, Text to Speech for disability and handicapped communication aids has become widely deployed in Mass Transit. Text to Speech is also finding new applications outside the disability market. For example, speech synthesis, combined with [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), allows for interaction with mobile devices via processing interfaces.

Text-to speech is also used in second language acquisition. Voki, for instance, is an educational tool created by Oddcast that allows users to create their own talking avatar, using different accents. They can be emailed, embedded on websites or shared on social media.

In addition, speech synthesis is a valuable computational aid for the analysis and assessment of speech disorders. A [voice quality](https://en.wikipedia.org/wiki/Voice_quality) synthesizer, developed by Jorge C. Lucero et al. at [University of Brasilia](https://en.wikipedia.org/wiki/University_of_Bras%C3%ADlia), simulates the physics of [phonation](https://en.wikipedia.org/wiki/Phonation) and includes models of vocal frequency jitter and tremor, airflow noise and laryngeal asymmetries. The synthesizer has been used to mimic the [timbre](https://en.wikipedia.org/wiki/Timbre) of [dysphonic](https://en.wikipedia.org/wiki/Dysphonic) speakers with controlled levels of roughness, breathiness and strain.

Today, the major application of TTS can be found in the latest operating systems of smartphones. Android and iOS both have been using TTS in the recent years. Even Amazon Kindle now come with online TTS engine. TTS involves artificial intelligence and machine learning as well, such as google search engine results, where the results are presented as text but they can also be synthesized to speech and heard by the end user.

**Conclusion:**

FreeTTS is a speech synthesis system written entirely in the Java programming language. It is based upon [Flite](http://www.cmuflite.org/): a small run-time speech synthesis engine developed at Carnegie Mellon University. Flite is derived from the [Festival](http://www.cstr.ed.ac.uk/projects/festival/) Speech Synthesis System from the University of Edinburgh and the [FestVox](http://festvox.org/) project from Carnegie Mellon University. Kevin and Alan generated the data used by FreeTTS. In addition, Kevin is the voice behind the diphone voices (kevin 8k, kevin 16k), and Alan is the voice behind the speaking clock.

Support for MBROLA voice output was contributed by Marc Schröder, text-to-speech Researcher in the [Language Technology Lab at DFKI](http://www.dfki.de/lt), Saarbrücken, Germany.