**NATIONAL COLLEGE OF ENGINEERING**

**(Affiliated to Tribhuvan University)**

**Talchhikhel, Lalitpur**



**[Subject Code: CT 654]**

**A MINOR PROJECT MIDTERM REPORT ON**

**“Text-to-Speech Synthesis System”**

**Submitted by**

**Bimal Adhikari [074/BCT/007]**

**Bishnu Khojwar [074/BCT/011]**

**Bishworam Ghimire [074/BCT/012]**

**Samrat Subedi [074/BCT/034]**

**Submitted to**

**Department of Computer and Electronics Engineering**

**Mangsir, 2077 B.S**

**Text-to-Speech Synthesis System**

**Subject Code: CT 654**

**Submitted by**

**Bimal Adhikari [074/BCT/007]**

**Bishnu Khojwar [074/BCT/011]**

**Bishworam Ghimire [074/BCT/012]**

**Samrat Subedi [074/BCT/034]**

**A minor project midterm report submitted in partial fulfillment of the requirement**

**for the**

**Department of Computer and Electronics Engineering**

**Department of Computer and Electronics Engineering**

**National College of Engineering**

**Talchhikhel, Lalitpur**

**Nepal**

**Mangsir, 2077 B.S.**

# COPYRIGHT

The author has agreed that the Library, Department of Electronics and Computer Engineering, National college of Engineering and Institute of Engineering may make this report freely available for inspection. Moreover, the author has agreed that for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work record here in or, in their absence, by the Head of Department where in the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, National college of engineering, in any use of the material of this project report, copying or publication or the other use of this report for financial gain without approval of the Department of Electronics and Computer Engineering, National College of Engineering and author’s written permission is prohibited.

Request for the permission to copy or to make any other use of the material in this report in whole or in part should be addresses to:

Head of Department

Department of Electronics and Computer Engineering

National College of Engineering

Talchhikhel, Lalitpur

# ACKNOWLEDGEMENT

Foremost, we would like to express our sincere gratitude to our supervisor **Er. Subash Pandey** for the continuous support of our project study, research and accomplishment of project, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped us in all the time of project. We could not have imagined having a better supervisor and mentor as **Er. Subash Pandey** for our project.

Besides our supervisor, we would like to thank Head of the Department **Er. Pradip Adhikari** and the teachers: **Er. Mohan Maharjan, Er. Sharmila Bista, Er. Anup Shrestha and Er. Rebanta Aryal** for their encouragement, insightful comments, and hard questions.

We thank our senior students of National College of Engineering for the stimulating discussions, and for all the fun we have had in the project period.

Any attempt at any level can 't be satisfactorily completed without the support and guidance of our parents and friends. Last but not the least, we would also express our deepest gratitude to NCE administration, which provided us with essential and all kind of logistics support and encouragement throughout the project study.

# ABSTRACT

Listening to the texts has become more popular than reading it thoroughly. Some people having some disorder of reading has become problematic. In this busy schedule of every person, listening to the text while doing other task can help a lot. The proposed system is an application that converts text into spoken word, by analyzing and processing the text to convert this processed text into synthesized speech representation of the text. Here, we developed a useful text-to-speech synthesizer in the form of a simple application that converts inputted text or files with different extensions uploaded, 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 through large volume of text easier to listen. It can also be helpful for those with language barriers.

*Keywords: text, text-to-speech synthesizer, files, extensions*

# TABLE OF CONTENTS

[COPYRIGHT iii](#_Toc56891516)

[ACKNOWLEDGEMENT iv](#_Toc56891517)

[ABSTRACT v](#_Toc56891518)

[TABLE OF CONTENTS vi](#_Toc56891519)

[LIST OF TABLES viii](#_Toc56891520)

[LIST OF FIGURES ix](#_Toc56891521)

[LIST OF ABBREVIATIONS x](#_Toc56891522)

[1. INTRODUCTION 1](#_Toc56891523)

[1.1 Background 1](#_Toc56891524)

[1.2 Problem Statement 7](#_Toc56891525)

[1.3 Aim and Objectives 8](#_Toc56891526)

[1.4 Scope 9](#_Toc56891527)

[1.5 Application 10](#_Toc56891528)

[1.6 Feasibility Analysis 11](#_Toc56891529)

[1.6.1. Economic feasibility 11](#_Toc56891530)

[1.6.2. Technical feasibility 11](#_Toc56891531)

[1.6.3. Operational feasibility 11](#_Toc56891532)

[2. LITERATURE REVIEW 12](#_Toc56891533)

[3. METHODOLOGY 18](#_Toc56891534)

[3.1 File Processing 19](#_Toc56891535)

[3.1.1 .txt file processing 19](#_Toc56891536)

[3.1.2 .pdf file processing 19](#_Toc56891537)

[3.1.3 .docx file processing 19](#_Toc56891538)

[3.2 Text Normalization 20](#_Toc56891539)

[3.2.1 Acronym and Abbreviation Handling 20](#_Toc56891540)

[3.3 Speech Generation 21](#_Toc56891541)

[4. REQUIREMENT ANALYSIS AND SPECIFICATION 22](#_Toc56891542)

[4.1 Functional Requirements 22](#_Toc56891543)

[4.2 Non-Functional Requirements 23](#_Toc56891544)

[4.3 Other Requirements 24](#_Toc56891545)

[4.3.1 Project software Interface 24](#_Toc56891546)

[4.3.2 Project Hardware Interface 24](#_Toc56891547)

[5. SYSTEM DIAGRAM 25](#_Toc56891548)

[5.1 Use case diagram 25](#_Toc56891549)

[5.2 System Sequence Diagram 26](#_Toc56891550)

[5.3 Activity Diagram 27](#_Toc56891551)

[6. EPILOGUE 28](#_Toc56891552)

[6.1 Task Accomplished 28](#_Toc56891553)

[6.2 Task Remaining 29](#_Toc56891554)

[6.3 Gantt Chart 30](#_Toc56891555)

[7. RESULT AND CONCLUSION 31](#_Toc56891556)

[7.1 Result 31](#_Toc56891557)

[7.2 Conclusion 31](#_Toc56891558)

[8. REFERENCES 32](#_Toc56891559)

[9. Appendix 33](#_Toc56891560)

# LIST OF TABLES

[Table 1: List of Consonants 11](#_Toc56785238)

[Table 2:List of Vowels 12](#_Toc56785239)

[Table 3 Comparison of existing systems 14](#_Toc56785240)

# LIST OF FIGURES

[Figure 1 Block diagram of TTS-System 18](#_Toc56885474)

[Figure 2 :Block diagram of Text Normalization 20](#_Toc56885475)

[Figure 3: Use case Diagram 25](#_Toc56885476)

[Figure 4: System Sequence Diagram 26](#_Toc56885477)

[Figure 5: Activity Diagram 27](#_Toc56885478)

[Figure 6: Gantt chart for TTS-System 30](#_Toc56885479)

# LIST OF ABBREVIATIONS

API – Application Programming Interface

DNN – Deep Neural Network

DSP – Digital Signal Processing

GUI – Graphical User Interface

HMM – Hidden Markov Model

IEEE - Institute of Electrical and Electronics Engineers

IEICE - Institute of Electronics, Information and Communication Engineers

IPA- International Phonetic Alphabet

LPC – Linear Predictive Coding

MATLAB - Matrix Laboratory

MLLR – Maximum Likelihood Linear Regression

MOS – Mean Opinion Score

MS – Modulation Spectrum

NAIST-Nara Institute of Science and Technology

NLP -Natural Language Processing

SWT – Standard Widget Toolkit

TTS – Text to Speech

VTTS – Visual Text to Speech

WWW – World Wide Web

# 1. INTRODUCTION

## 1.1 Background

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. The text-to-speech synthesizer (TTS) is the technology that lets the computer speak to you. The TTS system gets the text as the input and then a computer algorithm that calls the 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. [1]

Speech synthesis is a field of computer science that deals with designing computer systems that synthesize written text. It is a technology that allows a computer to convert a written text into speech via a speaker. Automatic speech synthesis is one of the fastest developing fields in the framework of speech science and engineering. The basic idea of text-to-speech (TTS) technology is to convert written input to spoken output by generating synthetic speech. Speech synthesis can be described as the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer.

The text-to-speech (TTS) synthesis procedure consists of three main phases:

The first is either direct input to the user or processing of the file. The file of different extensions are uploaded with the validation of required file extension. The text from those files are firstly extracted. Incase of pdf files, the number of pages in the file are extracted and the text from all the pages are stored in a string and sent for further processes. Incase of docx file, the number of paragraph are extracted and text from all the paragraph are stored in string and sent for further processing. Incase of txt files, normal file read operation can extract the text from the file which are then sent for further processing.

For analyzing and processing the text, we are using Natural Language Processing (NLP). The NLP helps in removal of white spaces, divide into sentences and further dividing into tokens, removal of unwanted symbols, acronym and abbreviation handling.

Finally, using Pyttsx3 module, this processed text is converted into a synthesized speech representation of the text.

Despite there being just 26 letters in the English language there are approximately 44 unique sounds, also known as phonemes. The 44 sounds help distinguish one word or meaning from another. Various letters and letter combinations are known as graphemes are used to represent the sounds.

The 44 English sounds fall into two categories: consonants and vowels. Also, they are distinguished into two types on the basis of vibration they cause voice and voiceless sound. There may be discovered lists with more or less than these 44 sounds.

Below is a list of the 44 phonemes along with their International Phonetic Alphabet symbols.

**Consonants:**

Table 1: List of Consonants

|  |  |  |  |
| --- | --- | --- | --- |
| Phoneme | IPA Symbol | Graphemes | Examples |
| 1 | b | b, bb | bug, bubble |
| 2 | d | d, dd, ed | dad, add, milled |
| 3 | f | f, ff, ph, gh, lf, ft | fat, cliff, phone, enough, half, often |
| 4 | g | g, gg, gh,gu,gue | gun, egg, ghost, guest, prologue |
| 5 | h | h, wh | hop, who |
| 6 | dʒ | j, ge, g, dge, di, gg | jam, wage, giraffe, edge, soldier, exaggerate |
| 7 | k | k, c, ch, cc, lk, qu, q(u), ck, x | kit, cat, chris, accent, folk, bouquet, queen, rack, box |
| 8 | l | l, ll | live, well |
| 9 | m | m, mm, mb, mn, lm | man, summer, comb, column, palm |
| 10 | n | n, nn,kn, gn, pn | net, funny, know, gnat, pneumonic |
| 11 | p | p, pp | pin, dippy |
| 12 | r | r, rr, wr, rh | run, carrot, wrench, rhyme |
| 13 | s | s, ss, c, sc, ps, st, ce, se | sit, less, circle, scene, psycho, listen, pace, course |
| 14 | t | t, tt, th, ed | tip, matter, thomas, ripped |
| 15 | v | v, f, ph, ve | vine, of, stephen, five |
| 16 | w | w, wh, u, o | wit, why, quick, choir |
| 17 | z | z, zz, s, ss, x, ze, se | zed, buzz, his, scissors, xylophone, craze |
| 18 | ʒ | s, si, z | treasure, division, azure |
| 19 | tʃ | ch, tch, tu, ti, te | chip, watch, future, action, righteous |
| 20 | ʃ | sh, ce, s, ci, si, ch, sci, ti | sham, ocean, sure, special, pension, machine, conscience, station |
| 21 | θ | th | Thongs |
| 22 | ð | th | Leather |
| 23 | ŋ | ng, n, ngue | ring, pink, tongue |
| 24 | j | y, i, j | you, onion, hallelujah |

**Vowels:**

Table 2:List of Vowels

|  |  |  |  |
| --- | --- | --- | --- |
| Phoneme | IPA Symbol | Graphemes | Examples |
| 25 | æ | a, ai, au | cat, plaid, laugh |
| 26 | eɪ | a, ai, eigh, aigh, ay, er, et, ei, au, a\_e, ea, ey | bay, maid, weigh, straight, pay, foyer, filet, eight, gauge, mate, break, they |
| 27 | e | e, ea, u, ie, ai, a, eo, ei, ae | end, bread, bury, friend, said, many, leopard, heifer, aesthetic |
| 28 | i: | e, ee, ea, y, ey, oe, ie, i, ei, eo, ay | be, bee, meat, lady, key, phoenix, grief, ski, deceive, people, quay |
| 29 | ɪ | i, e, o, u, ui, y, ie | it, england, women, busy, guild, gym, sieve |
| 30 | aɪ | i, y, igh, ie, uy, ye, ai, is, eigh, i\_e | spider, sky, night, pie, guy, stye, aisle, island, height, kite |
| 31 | ɒ | a, ho, au, aw, ough | swan, honest, maul, slaw, fought |
| 32 | oʊ | o, oa, o\_e, oe, ow, ough, eau, oo, ew | open, moat, bone, toe, sow, dough, beau, brooch, sew |
| 33 | ʊ | o, oo, u,ou | wolf, look, bush, would |
| 34 | ʌ | u, o, oo, ou | lug, monkey, blood, double |
| 35 | u: | o, oo, ew, ue, u\_e, oe, ough, ui, oew, ou | who, loon, dew, blue, flute, shoe, through, fruit, manoeuvre, group |
| 36 | ɔɪ | oi, oy, uoy | join, boy, buoy |
| 37 | aʊ | ow, ou, ough | now, shout, bough |
| 38 | ə | a, er, i, ar, our, ur | about, ladder, pencil, dollar, honour, augur |
| 39 | eəʳ | air, are, ear, ere, eir, ayer | chair, dare, pear, where, their, prayer |
| 40 | ɑ: | a | Arm |
| 41 | ɜ:ʳ | ir, er, ur, ear, or, our, yr | bird, term, burn, pearl, word, journey, myrtle |
| 42 | ɔ: | aw, a, or, oor, ore, oar, our, augh, ar, ough, au | paw, ball, fork, poor, fore, board, four, taught, war, bought, sauce |
| 43 | ɪəʳ | ear, eer, ere, ier | ear, steer, here, tier |
| 44 | ʊəʳ | ure, our | cure, tourist |

The most important qualities of modern speech synthesis systems are its naturalness and intelligibility. By naturalness, we mean how closely the synthesized speech resembles real human speech. Intelligibility, on the other hand, describes the ease with which the speech is understood. The maximization of these two criteria is the main development goal in the TTS field.

Text to speech software can be enormously helpful for anyone who's visually impaired or has a condition like dyslexia that makes reading on screens tricky. It can also help overcome language barriers for people who are in the process of learning. Text to speech software is also ideal if we want to listen to a document while doing something else, or if we want to sense-check something we've written. It is also a valuable computational aid for those with speech disorders.

There are many existing systems available. Some of the existing system with the strength and weakness are:

Table 3 Comparison of existing systems

|  |  |  |  |
| --- | --- | --- | --- |
| S. N | Systems | Strength | Weakness |
| 1 | Acapela | Covers variety of language  Good voice quality  Flexibility of voice control | Chinese language not available |
| 2. | NeoSpeech | Very high-quality natural voice  Ease to use  Chinese, Italian, French, English, etc. accepted | Only available for few languages. |
| 3 | Loquendo | Covers most languages  Great support resources | User interface not friendly  Unstable server |
| 4 | Panopreter basic | Quick and simple to use  Export sound in WAV and MP3 formats | For windows only |
| 5 | Wordtalk | Integrates with MS Word | Not all functions are clear |

Between these systems following features are common:

* Multiple languages available
* User lexicon
* Good voice quality
* Ease of user interface on some systems
* Mostly available for windows.

## 1.2 Problem Statement

In Text-To-Speech-Synthesis systems, there are several problems in text preprocessing, such as numerals, abbreviations, and acronyms. Correct prosody and pronunciation analysis from the written text is also a major problem today. Also, the unclarities of the spoken word is another problem. The other problem is the unattractive and not user-friendly GUI.

For the user who is in the process of learning the correct pronunciations of the word, this problem is severe. It doesn’t help those who want to overcome language barriers.

For those with less technical skills face difficulty in using the software. Due to the difficulty of using the GUIs for the user, the user faces a lot of problems. Also, the unattractiveness of the GUI doesn’t encourage a user to use the system.

Sometimes you may feel tired to read an e-book and you just want to liberate your eyes for a bit but on other hand you don’t want to give up your books. This has been problem for many readers.

It is expected that the new system will reduce and improve most of the problems encountered in the old system.

## 1.3 Aim

* The aim of this project is to convert the text into speech.

## 1.4 Objectives

* To make familiar with correct pronunciation and prosody.
* To make ease on listening e-book.
* To provide verbal communication for individual who are unable to speak.
* To help students work around their reading difficulties and access the classroom materials.

## 1.5 Scope

The scope of this project is to facilitate people for correct prosody and pronunciation. For correct pronunciation and exact prosody for the word, this system can be helpful. Those individuals learning the correct pronunciation, this system can be helpful. Also, the synthesized speech is of good quality. The English language which it delivers is of high quality. So, the sound is clear and realistic. This system overcomes the problem of language barrier.

This system consists of simple user interface. So, the person with less technical skill can easily use the system. The attractive and simple user interface makes easier for the user to use the system. Due to simple and easy user interface, this encourages the user to use the system.

For those users who are tired of reading the e-books and are on long exposure to laptop or desktop and on the other hand don’t want to give up the books, this system is helpful. User can enjoy the eBooks and continue doing other jobs or rest using this system.

## 1.6 Application

This system can be used by everyone who wants to listen text documents instead of reading them. Text files can be easily translated into audio with the help of our system. People with learning disabilities who have difficulty reading large amounts of text due to dyslexia or other problems really benefit from TTS, offering them an easier option for experiencing contents. Those readers suffering from these problems gets benefit from this TTS system.

For those users tired of reading the documents can get real benefit from this software. Also, users exposed to digital gadgets for long time can enjoy this system which provides real time natural listening sound.

Synthesized speech can be used also in many educational situations. It can be used for special tasks like spelling and pronunciation teaching. It can also be used with interactive educational applications.

## 1.7 Feasibility Analysis

### 1.7.1. Social feasibility

The system provides good impact to the people who are affected by this introduction. It is highly applicable to the blind people of the society for listening e-books, other messages and text.

### 1.7.2. Technical feasibility

The Technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

We find positive answer of all the question mentioned above. We have all the equipment, software and other technical requirement for the project and the system can also be upgraded if necessary.

### 1.7.3. Operational feasibility

The aspect of study is to check the level of acceptance of the system by the user. This include the process of training the user to use the system efficiently. No training is required as it can be handling by just reading the manual given.

### 1.7.4. Legal feasibility

The legal perception of the project is clean. The system is not associated with any activity which is illegal. The system is not out of the constitution of the country.

# 2. LITERATURE REVIEW

Isewon I and his team [1] researched that text- to- speech synthesizer project is done 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. Their system interfaces with a text to speech engine developed for American English. They did it in the form of a simple application that converts inputted text into synthesized speech and reads out to the user which can then be saved as a .mp3 file. The system was developed using Java programming language. They have used a method called Concatenative Synthesis to produces the most natural-sounding synthesized speech. Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. According to this paper, 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.

Their system converts text to speech either by typing the text into the text field provided or by coping from an external document in the local machine and then pasting it in the text field provided in the application. It also provides functionality that allows the user to browse the World Wide Web (www) on the application. The system is capable of reading any portion of the web page the user browses. This can be achieved by the user highlighting the portion he wants to be read out loud by the system and then clicking on the “Play” button. The application was divided into two main modules - the main application module which includes the basic GUI components which handle the basic operations of the application such as the input of parameters for conversion either via file or direct keyboard input or the browser. This would make use of the open-source API called SWT and DJ Native Swing. In the second module, the main conversion engine integrated into the main module was for the acceptance of data hence the conversion. This would implement the API called free TTS.

Shetake P.S and her team [2] in the paper said that a text to speech converter convert’s normal language text into speech. Text to speech converter is useful in different applications. Customer support dialog systems Interactive voice response (IVR) systems etc. and are also useful in applied research. This application is more helpful in banking, toys, and many other applications like checking marks, railways, aid to the physically challenged persons, language education, and fundamental and applied research. etc. But text to speech conversion is not that easy for the machine as it is for human. Basic steps that machine has to follow for text to speech analysis are database creation, character recognition, and text to speech conversion. This paper surveys methods related to character recognition as well as approaches used for text to speech conversion for the machine. In this paper, there is discussed character recognition and speech synthesis techniques which are very useful to perform the task text to speech conversion. As stated, TTS is divided into two subproblems: character recognition and speech. To get the best speech synthesis rate, the database of the system should be large. So, there is scope to increase the database of the proposed system.

Swetha N. and Anuradha N. [3] In the research paper focused on the conversion of character-to-voice. In this paper, they presented the idea about developing a pc-based text-to-speech synthesizer using MATLAB. Here Text-To-Speech or speech synthesis is taken as an automatic production of speech, by ‘grapheme to phoneme’ transcription. In this paper, it is stated that the database of various recorded alphabets and digits in the form of wave files is created. Then the texts files of the text to be read are created. Then the .txt files are opened in MATLAB. The file is opened and read. For every character read, corresponding wave files are read. In this system, there are various problems in text preprocessing, such as numerals, abbreviations, and acronyms.

Jacob A. and Mythili P. [4] discussed the implementation details of a child-friendly, good quality, English TTS system that is phoneme-based, concatenative, easy to use with little memory. Direct waveform concatenation and linear prediction coding (LPC) are used. Most existing TTS systems are unit-selection based, which use standard speech databases available in neutral adult voices. Here reduced memory is achieved by the concatenation of phonemes and by replacing phonetic wave files with their LPC coefficients. The linguistic analysis was used to reduce the algorithmic complexity instead of signal processing techniques. Suﬃcient degree of customization and generalization catering to the needs of the child user had been included through the provision for vocabulary and voice selection to suit the requisites of the child. Prosody had also been incorporated. This inexpensive TTS system was implemented in MATLAB, with the synthesis presented using a graphical user interface (GUI), thus making it child friendly. This can be used not only as an interesting language learning aid for the normal child but it also serves as a speech aid to the vocally disabled child. The quality of the synthesized speech was evaluated using the mean opinion score (MOS) provision for varying the voice quality. Another feature of this work is that it was implemented using a female voice, whereas most of the successful LPC-based TTS have been implemented in the male voice. This TTS further has an add-on facility in that new words can be synthesized, after adding these words, their transcription, and constituent wave ﬁle names to the irrespective databases. The prosody of the utterance can be designed to vary depending on the nature of the recordings in the speech database from which the phoneme segments are excised. Thus, a simple, ﬂexible, and eﬃcient TTS that can be user deﬁned has been set up with minimum resources to serve multiple purposes. Though this had been developed for English, it can be suitably modiﬁed for any other language. This TTS was found to be a successful vocal aid/language learning aid as the users were able to get a real feel of phonemes, the most basic speech units. The learning environment can be conditioned to any particular accent by using an appropriate combination of database and pronunciation dictionary. Alternately, content speciﬁc learning too can be encouraged implicitly. By suitable design of the TTS vocabulary and database, a child can be familiarized with all common terms associated with any speciﬁc topic. Thus, such a TTS helps the child user get acquainted with the regular as well as any other selective vocabulary.

Tamura M. and his team [5] in the paper describes a technique for synthesizing speech with any desired voice. The technique is based on an HMM (Hidden Markov Model)-based text-to-speech (TTS) system and MLLR (Maximum Likelihood Linear Regression) adaptation algorithm. To generate speech of an arbitrarily given target speaker, speaker-independent speech units, i.e., average voice models, are adapted to the target speaker using the MLLR framework. In addition to spectrum and pitch adaptation, they derived an algorithm for adaptation of state duration. They demonstrated that a few sentences uttered by a target speaker are sufficient to adapt not only voice characteristics but also prosodic features. The synthetic speech generated from adapted models using only four sentences is very close to that from speaker-dependent models trained using a large amount of speech data. To convert the duration model parameters, they derived the MLLR algorithm for duration models. They have shown that synthetic speech generated from adapted models using average voice models becomes closer to the target speaker’s voice. Their future work is a subjective and objective evaluation of this technique.

Toda T. and Tokuda K. [6] present a text-to-speech system developed at Nara Institute of Science and Technology (NAIST) for the Blizzard Challenge 2015. They have developed their TTS system based on a statistical parametric speech synthesis technique using a hidden Markov model (HMM). To improve the quality of synthetic speech, they have newly implemented two techniques for the traditional HMM-based speech synthesis framework, pre-processing for producing smooth parameter trajectories to be modeled with HMM and speech parameter generation considering the modulation spectrum. The developed system has been submitted to the mono-lingual task and its performance has been demonstrated from the results of large-scaled subjective evaluation. Their research group, a speech synthesis group of Augmented Human Communication Laboratory, Nara Institute of Science and Technology (NAIST), studies various speech synthesis techniques, such as high-quality statistical parametric speech synthesis techniques, real-time voice conversion techniques for augmented speech production (e.g., voice/vocal effector or a speaking aid system for laryngectomees, towards the development of technologies to break down existing barriers in their speech communication. To submit a TTS system from their group the Blizzard Challenge 2015, they have developed their system, the NAIST TTS system based on a statistical parametric speech synthesis technique using Hidden Markov Model (HMM). To improve the quality of synthetic speech, two techniques are newly implemented for the traditional HMM-based speech synthesis framework preprocessing for producing smooth parameter trajectories to be modeled with HMM and speech parameter generation considering the modulation spectrum (MS) of speech parameters. The developed system has been submitted to the monolingual task and its performance has been demonstrated from there sorts of large-scaled subjective evaluations. This paper describes the details of the NAIST TTS system. They also brieﬂy discuss the results of large-scaled subjective evaluations on naturalness, similarity to the original speaker, and intelligibility, which they’re provided from the organizers. This paper has presented the NAIST TTS system for the Blizzard Challenge 2015. The pre-processing for smoothing parameter trajectories and the speech parameter generation considering the modulation spectrum have been implemented in their system. The results in the challenge have demonstrated that their system is capable of synthesizing naturally sounding speech.

Tokuda K. and his team [7] in their research paper gave a general overview of Hidden Markov model (HMM)-based speech synthesis, which has recently been demonstrated to be very effective in synthesizing speech. The main advantage of this approach is its ﬂexibility in changing speaker identities, emotions, and speaking styles. This paper also discusses the relation between the HMM-based approach and the more conventional unit-selection approach that has dominated over the last decades. Finally, advanced techniques for future developments are described. Index Terms—text-to-speech synthesis, hidden Markov model, HMM-based speech synthesis, statistical parametric speech synthesis, HTS. This paper gave a general overview of HMM-based speech synthesis and its recent advances. HMM-based speech synthesis has started to be used in daily life, e.g., cellphones, smartphones, in-car navigation systems, and call centers. Although the quality of synthesized speech generated by HMM-based speech synthesis has been drastically improved recently, its naturalness is still far from that of actual human speech. In the conversational speech, the naturalness of prosody is still insufﬁcient to properly convey non-verbal information, e.g., emotional expressions and emphasis. To ﬁll the gap between natural and synthesized speech, the statistical approaches described in the research paper will be more important in the future.

Parker J. and his team [8] presented a Visual text to speech system (VTTS) based on a deep neural network (DNN). Given an input text sentence and a set of expression tags, the VTTS can produce not only the audio speech, but also the accompanying facial movements. The expressions can either be one of the expressions in the training corpus or a blend of expressions from the training corpus. Furthermore, they present a method of adapting a previously trained DNN to include a new expression using a small amount of training data. Experiments show that the proposed DNN-based VTTS is preferred by 57.9% over the baseline hidden Markov model-based VTTS which uses cluster adaptive training. Index Terms used are Expressive Visual Text to Speech, Expression Adaptation, Deep Neural Network. In this paper, they present a system that, given a text sentence and some expressive tag, produces a photorealistic talking head. Talking heads can be used to improve human-machine interaction systems. Such systems have been used in education, in reading news or eBooks, and even in post-processing of ﬁlm production. In this paper, they present a DNN-based text-driven expressive talking head. The contributions of this work include an adaptation of audio DNN-based text to speech (TTS) to VTTS, a novel method of producing expressive visual speech using a DNN, yielding superior results compared to HMM-based VTTS, and a method of adapting a pre-trained TTS system to incorporate a new expression using a small amount of adaptation data. This paper presented a DNN-based expressive talking head. Unlike other systems, the entire face is modeled, which is important for expressive visual speech. Furthermore, new expressions can be added to the model with a small amount of adaptation data and without retraining the network. Their method outperforms an expressive HMM-based talking head. Attention is being turned to investigate alternate methods of modeling different expressions, in particular using a gated ﬁnal layer, where the ﬁnal layer models a three-way relationship between the penultimate layer, the expression, and the output. Furthermore, they wish to investigate the use of similar techniques to jointly model speaker and expression and so perform both speaker and expression adaptation.

# 3. METHODOLOGY

.txt file

.docx file

Voice Gender

Direct text

.pdf file

.pdf file Processing

.docx file Processing

.txt file Processing

File processing

Extracted text

White space Removal

Divided into sentences.

Divided into tokens.

Acronym and Abbreviation handling

Removal of unnecessary symbols

Text Normalization

Normalized text

Speech Synthesis

Speech

Figure 1 Block diagram of TTS-System

The block diagram is presented in the figure above. It is mainly divided into three subgroups which can be named as File processing, Text Normalization and Speech generation.

In text-to-speech system, the input to the system can either be the direct text from the user or the file (i.e. .txt,.docx, .pdf) which consists of texts. The user also provides the gender type of the voice which the module synthesizes as per need of user.

## 3.1 File Processing

This is the first block of text-to-speech synthesis system. The file from the location given by the user is first distinguished. The unsupported file with extension other than the .pdf, .docx, .txt are discarded.

### 3.1.1 .txt file processing

This is the normal file read operation. Reading a text file into a string stores the content of the file in a single string, replacing the newline characters with spaces.

### 3.1.2 .pdf file processing

In pdf file processing, the PyPdf2 module enables us to extract the texts of all the pages. It lists out the number of pages of the pdf. Then it converts the whole text of the document into a single string datatype.

### 3.1.3 .docx file processing

Word documents contain formatted text wrapped within three object levels. Lowest level- Run objects, Middle level- Paragraph objects and Highest level- Document object. So, we cannot work with these documents using normal text editors. But we can manipulate these words using the python-docx module. The text of the document is converted to a single string data type.

## 3.2 Text Normalization

The direct text from the user or the extracted text from document is then passed to the text normalization block where the normalization of the whole text occurs. First of all, the extra white spaces are removed from the text. Then the sentences are divided into collection of tokens (such as words, numbers, dates, and other types). The symbols which don’t have any significance while speaking are discarded. Non-natural language tokens such as abbreviations and acronyms must be transformed into natural language tokens. Most of these tasks are handled by the module. We need to work on acronym and abbreviation handling.

White space Removal

Divided into sentences.

Divided into tokens.

Removal of unnecessary symbols

Acronym and Abbreviation handling

Unnormalized text

Normalized text

Figure 2 :Block diagram of Text Normalization

## 3.2.1 Acronym and Abbreviation Handling

The acronym and abbreviations which aren’t handled by the module but are used in daily life are handled in this block. These abbreviations and acronym are replaced by natural language tokens and are made ready for synthesis.

## 3.3 Speech Generation

For the purpose of speech generation, we used an external library. This TTS Engine handles most of the text normalization process and speech synthesis process. The text-to-speech features for this engine are based on languages installed in the operating system. By default, it come together with the language pack during the installation of the operating system. We need to install the language pack manually if you intend to use other languages. It is a simple and offline library for conversion of text to speech.

Text as an input along with the gender from user is provided to the module. The speech rate and volume are provided as a default value for the TTS Engine. The TTS engine by processing the text gives audio as an output for the user.

# 4. REQUIREMENT ANALYSIS AND SPECIFICATION

## 4.1 Functional Requirements

Functional requirement explains what has to be done by identifying the necessary task, action or activity that must be accomplished. Some of the requirement that resemble to our project are listed below:

**Interface:**

The Graphical User Interface (GUI) that communicates with the user with ease.

**Storage:**

Device storage is used to store the files that was synthesized using the application.

**Easy Integration:**

The application should be easy to integrate with the existing system.

**Supportability:**

Support multiple voices and file types.

## 4.2 Non-Functional Requirements

Non-functional requirements are the requirement that specify criteria that can be used to judge the operation of a system rather than specific behavior.

**Reliability**

It is required that the system should be available all the time and detect and tolerate fault in the system and so ensure that these faults don’t lead to system failure. Also, the system is built by Python and this add more confidence to the system.

**Performance**

The application would be used by numerous users. So, it is required that the system should take minimum time to produce output. The system don’t spend time on requirements that are not necessary hence improving performance of the system.

**Accuracy**

The application is proposed to be real time. So, it is required that the high accuracy is maintained. Our system has very high accurate. The system reads almost all text correctly and it handles almost all numerals and symbol. Some acronyms and abbreviations are still need to be handled.

**Maintainability:**

The system should be capable of being retained, restored and serviceable. The system should enable to fix the bugs, optimize existing functionality and adjust code to prevent future issues.

**Efficiency:**

It is required that the software makes optimal use of resources, and optimize the speed and memory with which the system executes and consumes for its operation respectively.

## 4.3 Other Requirements

### 4.3.1 Project software Interface

**Front End:**

* Dual Core Processor Based Computer
* 2 GB or higher RAM
* 100 MB or higher Hard Disk Space
* Python version 3.7 or above
* Window 7 ,8,8.1 and 10

**Back End**

* Flask, Python and its libraries

### 4.3.2 Project Hardware Interface

* Computer or Laptop

# 5. SYSTEM DIAGRAM

## 5.1 Use case diagram

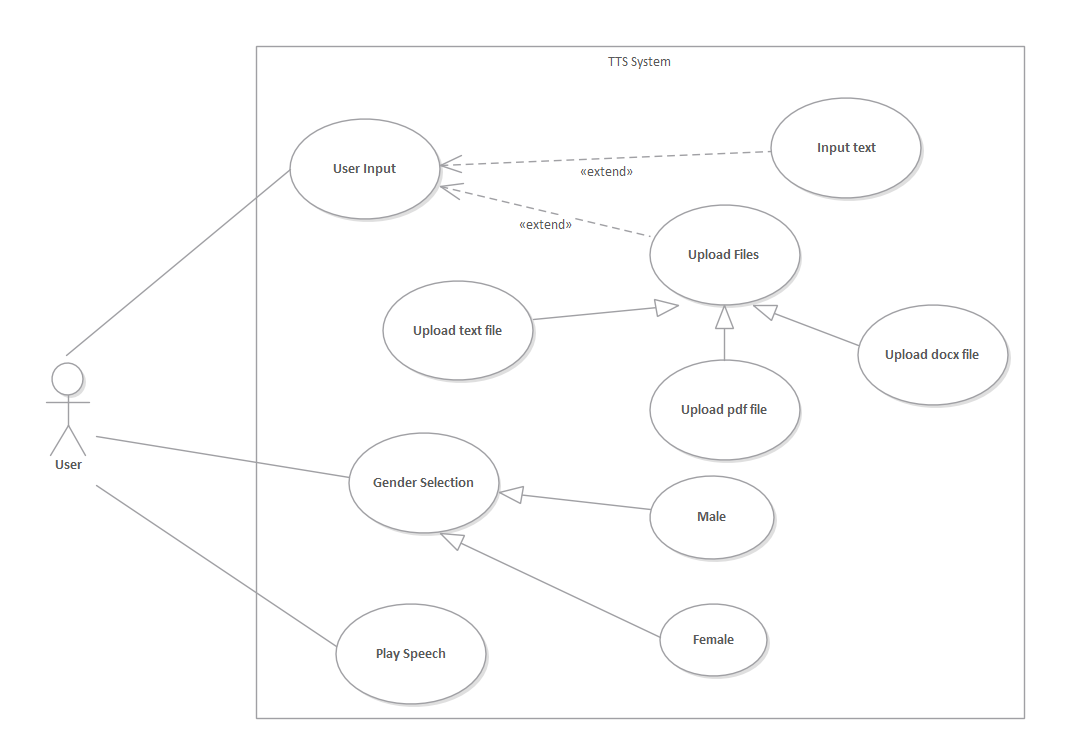


Figure 3: Use case Diagram

As shown in figure 3:Use case Diagram, the input can be either direct input text or the file with the extension supported i.e. .pdf,.docx,.txt. The user is able to select the voice according to gender and finally play the speech.

## 5.2 System Sequence Diagram

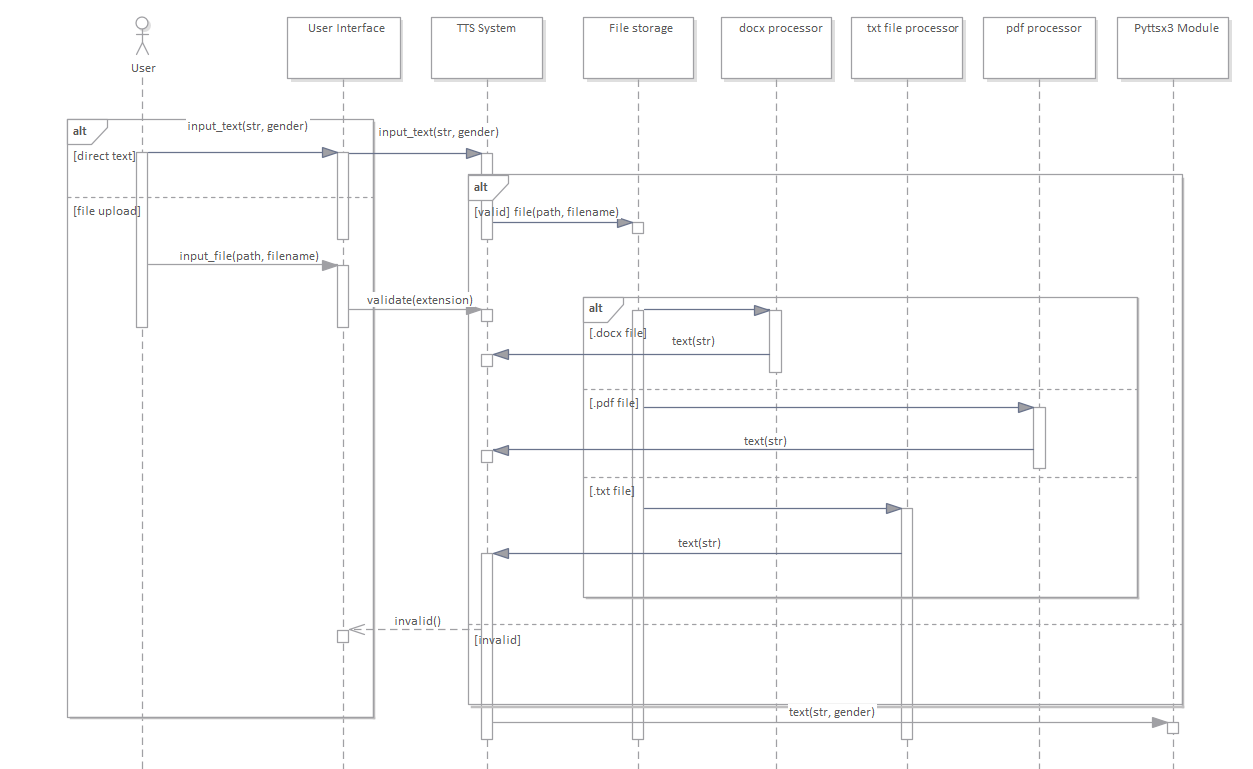


Figure : System Sequence Diagram

As shown in figure 4: System Sequence Diagram, the input from user to the user interface is either direct input\_text or the input\_file. If it is direct text or file with the valid extension, message is forwarded to TTS System. For the file uploaded, the file is passed to respective processor which extract text as string and passes to TTS System which further passes the text and gender to Pyttsx3 module for speech generation.

If the file extension is invalid, TTS System sends message to user interface.

## 5.3 Activity Diagram

Figure 5: Activity Diagram

As shown in figure 5: Activity Diagram, user visit the user interface. The input can be either direct text or the file upload. The user selects the gender and send the text and gender to synthesize the speech in case of direct input.

For file upload, the user selects the file. The system checks for empty file upload and then for supporting extensions(.pdf,.docx,.txt). If both the conditions are true then the file is processed and sent for speech synthesis. If there is empty upload or invalid extension, the error message is displayed and ask for reuploading the file.

# 6. EPILOGUE

## Task Accomplished

**Direct conversion of text to speech.**

The system is capable to convert the direct text written in text area into speech. We can also paste the textual content from other source into the text area which eventually converted into speech.

**Upload file of various extension:**

The system is capable to accept files of three selected extension, namely .txt file, .docx file, .pdf file.

**File extension validation:**

The system is capable to validate the extension of the file. The file extensions .txt, .pdf, .docx are only accepted. All other extension rather than these gets rejected.

**Empty upload validation:**

The system is capable to detect if the file is selected or not. If no file is selected then the system notifies the user.

**Conversion of file into string:**

The system is capable to convert the files with valid extension into string which is eventually converted to speech.

**UI designing:**

For proper interaction between user and system, simple and attractive user interface has been developed using a Python based micro web framework-Flask.

**Gender selection:**

The system is able to synthesize the speech into two different voice i.e. male voice and female voice. User can only choose the gender in case of direct conversion of text to speech.

## 6.2 Task Remaining

**Acronym and abbreviation handling:**

The system will be able to handle the acronym and abbreviation.

**Gender selection for uploaded file conversion:**

Currently for uploaded file conversion, the voice is set to male. The system will be able to select different gender in this scenario also.

**Adding extra feature**

i.e. playing from desired page number in pdf file.

## 6.3 Gantt Chart

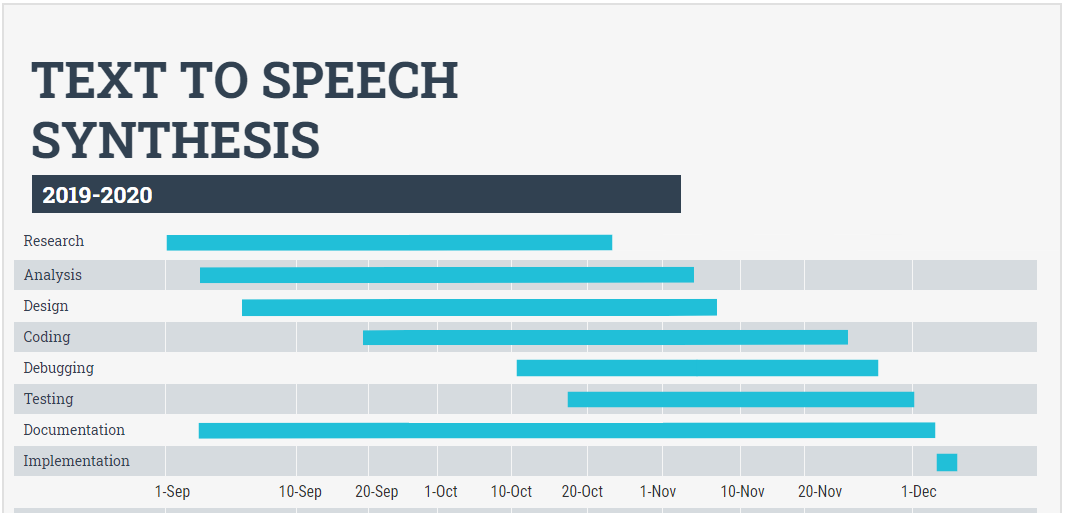


Figure 6: Gantt chart for TTS-System

# 7. RESULT AND CONCLUSION

## 7.1 Result

The text to speech synthesizing system converts given direct text in string format into speech. The document file of valid extensions(.txt,.pdf,.docx) is uploaded into given location, converted first into a single string and then eventually converted to speech.

## 7.2 Conclusion

In this text to speech synthesizing system, the text is analyzed, white spaces are removed, abbreviations and acronym are handled and a clear voice is heard. The system handles the files and extract text from the files which gets converted to speech providing good clear quality voices. The simple user interface facilitates the user with basic knowledge of using computer to use this system easily and effectively.

# 8. REFERENCES

[1] I.Isewon, J.Oyelade, O.Oladipupo*. "Design and Implementation of Text to Speech Conversion for Visually Impaired People"*, International Journal of Applied Information Systems,2014

[2] P.S.Shetake, S.A.Patil, P.M.Jadhav. *"Review of Text To Speech Conversion Methods"*, International Journal of Industrial Electronics and Electrical Engineering (IJIEEE),2014

[3] N.Swetha,K.Anuradha, *"Text-to-speech Conversion”,* International Journal of Advanced Trends in Computer Science and Engineering,2013

[4] A.Jacob and P.Mythili. *"Developing a Child Friendly Text-to-Speech System".,* https://doi.org/10.1155/2008/597971,2008

[5] Masatsune Tamura, Takashi Masuko, Keiichi Tokuda, Takao Kobayashi *"Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR”, IEEE* International Conference on Acoustics, Speech, and Signal Processing. Proceedings, 2001

[6] T.Toda, K.Tokuda, *"A speech parameter generation algorithm considering global variance for HMM-based speech synthesis",* IEICE Transactions on Information and Systems, 2007

[7] K.Tokuda, Y. Nankaku, T.Toda, H.Zen, J.Yamagishi, *"Speech synthesis based on hidden Markov models",* Proceedings of the IEEE, 2013

[8] J.Parker,Y.Stylianou, R.Cipolla,R.Maia, *"Expressive visual text to speech and expression adaptation using deep neural networks",* IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),2018

[9] Clark J., and Yallop C., *“An Introduction to Phonetics and Phonology”*, Blackwell Publishing,3rd edition, 2007

# 9. Appendix

