Urdu Text to Speech Synthesizer

By

Muhammad Hassan Siddiqui

MSCSF15M005

Supervised by

Dr. Muhammad Kamran Malik

Assistant Professor, PUCIT

(June, 2018)

Punjab University College of Information Technology,

University of the Punjab, Lahore, Pakistan.

Urdu Text to Speech Synthesizer

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE

DEGREE OF

MASTER OF PHILOSOPHY

IN

COMPUTER SCIENCE

By

Muhammad Hassan Siddiqui

MSCSF15M005

Supervised by

Dr. Muhammad Kamran Malik

Assistant Professor, PUCIT

(June, 2018)

Punjab University College of Information Technology,

University of the Punjab, Lahore, Pakistan.

Evaluation of M. Phil. Thesis

We have evaluated the M. Phil. thesis titled

Urdu Text to Speech Synthesizer

Submitted by Mr. **Muhammad Hassan Siddiqui**, **MSCSF15M005**, session 2015-2018 in partial fulfillment of the M. Phil. degree in Computer Science. We have also assessed the candidate through viva-voice.

We are satisfied with the thesis and performance of the candidate in the examination and are of the opinion that she fulfills the requirements as set in the rules and regulations for the M.Phil. degree in Computer Science at the University of the Punjab.

Thesis Supervisor: Dr. Muhammad Kamran Malik
Assistant Professor
Punjab University College of Information Technology
University of the Punjab, Lahore

Principal of the College: Dr. Syed Mansoor Sarwar Principal,
Punjab University College of Information Technology
University of the Punjab, Lahore

UNIVERSITY OF THE PUNJAB

Author: **Muhammad Hassan Siddiqui**

Title: **Urdu Text to Speech Synthesizer**

Department: **Punjab University College of Information Technology**

Degree: **M. Phil. (Computer Science)**

Dedicated to

Abstract

The process of transformation of data from textual form into voice output is called speech synthesis. The type of system which plays out this assignment is known as a text to speech (TTS) system or sometimes as speech synthesizer. The synthesized speech sometimes referred as artificial speech. This system can assist individuals with various handicaps like visual weakness in their day by day undertakings and furthermore help in inter language correspondence. These systems are being used in screen reading software, games, animations, machine to machine communication and many other applications. Artificial speech can be synthesized by concatenation of small segments of speech. Formant synthesis and unit selection synthesis are based on this types of synthesis system. Another popular approach which is being used in most recent couple of years is statistical parametric speech synthesis. This approach gives extremely promising results when used with Hidden Markov (HMM) or Deep Neural Networks (DNN).

In this paper, development of statistical parametric speech synthesis system using Festival TTS for Urdu is discussed. HMM based TTS system is developed using 70 minutes of speech data. This paper divides complete process of speech synthesis in two sub processes i.e. text preprocessing and speech synthesis. Text preprocessor system identifies and processes special characters, numbers (Arabic numerals, floating point and whole numbers), time and dates in input data. Speech synthesis process takes processed input and converts it into corresponding voice. In this process, Lexicon and letter to sound rules of Hindi are used. Performance of the system is evaluated using Diagnostic Rhyme Test (DRT), Modified Diagnostic Rhyme Test (M-DRT), Naturalness Test, Intelligibility Test and Usability Test in the last step.

Keywords: Text to Speech, Urdu Text Preprocessor, Hidden Markov model, Festival, Festvox, Urdu Text to Speech

Acknowledgements

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speech is the most important medium of conveying opinions and expressing feelings and thoughts. Human convert their thoughts into speech by using words, phrases and sentences for communicating with each other [1]. Speech in human is generated by the vibrations in vocal cords with the air passage. Text to speech (TTS) synthesis is the process of transformation of textual data into voice output. In this process, small speech units called phonemes are joined in proper order to generate speech signals [2]. Speech data is obtained by first recording natural speech with the help of some type of recording systems and is converted to digital form. The digital data is sampled and stored in computer, after that passed back to analog signals and is converted back to speech [3].

TTS systems are becoming important as they can be used in machines to effectively transmit information to human using artificial speech as information exchange through computers has become the integral part of new era. Visually impaired people usually suffer while using computer technology when there is no assistant or computer is not enough interactive which makes TTS systems necessity of modern life. These systems increase the degree to which blind people can interact with sighted people [4] and could boost up their hope to survive in this world gracefully [5]. Many applications of speech synthesis are emerging such as machines that read for blinds, aids for handicaps, computers that interact with user through speech. For all these applications, a text to speech conversion system is used [6].

The TTS system comprises of two main stages. One is called Natural Language Processing (NLP) and other is called Speech Synthesis (SS). This is shown in figure 1.1.

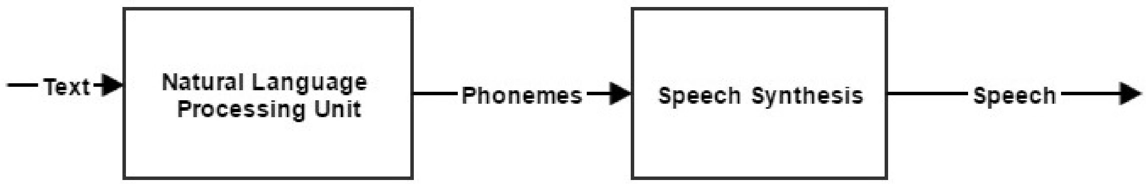

Figure 1.1: Main stages of TTS system

NLP unit is further divided in many sub processes. In start, word boundaries are marked by tokenizer which is called text normalization. This is followed by marking of syllable boundaries by using letter to sound rules. The syllabified data is processed to apply sound change rules. The data is also process to get context of each word in given sentence because for human, it is easy to guess context of each word in a sentence. For example, human can easily guess that if it is پل (moment) or پل (bridge) by reading a sentence. A computer program cannot guess that without context of that word. In the end, intonation and stress are added in the processed data. This processed data is then used in speech synthesizer to generate speech using different Digital Signal Processing techniques. The naturalness and intelligibility of the system is used to evaluate quality of the speech synthesizer.

In digital world, there are some people who can read and understand different languages and some who can't understand languages except their own language. Speech to text conversion system can also provide a facility to exchange information between people speaking different languages [2]. TTS systems are also needed to reduce the extinction of minority languages. As minority languages of the world are facing challenge of extinction considerable efforts are going on from last few years for their survival. Fon language is spoken in Republic of Benin and some other regions of Africa and it is also facing challenge of extinction [7]. Similarly, the Xitsonga is the language which is spoken by the people of excess of three African nations. TTS system of such languages will help lot of people of different literacy level [8]. Urdu is being spoken by more 70 million individuals on the planet [9] and it is national language of Pakistan and numerous states in India. A TTS system for Urdu will be extremely helping for partially or completely blind and illiterate people in these countries.

1.1 Types of speech synthesis

Different techniques exist for synthesizing artificial speech. Some of these techniques are discussed below.

1.1.1 Formant synthesis

In this technique, speech waveform is generated using concatenation of sine wave with the help of some algorithms to model a source of sound [10]. All speech parameters are changed periodically in order to get speech waveform. Some set of rules are also used to generate speech due to which this technique is also called rule based speech synthesis. As it is exceptionally hard to precisely portray speech generation process in set of rules due to which speech generated by this technique is not very natural but intelligible.

1.1.2 Concatenative synthesis

In concatenative synthesis, small units are selected from carrier sentences which are joined to form speech of complete sentence. These small units are called phonemes. The pronunciation of a word is described using collection of such units. As for any language, number of such units are few, therefor this procedure is simple when contrasted with the formant synthesis. English have 44 such phonemes. Urdu on the other hand have 44 consonants, 7 long nasal vowels, 3 short vowels, 8 long vowels, and many diphthongs [11]. This synthesized speech in this technique has less distortion but it is not very natural. That is the reason the produced speech may not look like the contributor speaker in training database [12].

1.1.3 Statistical parametric speech synthesis

Statistical parametric speech synthesis is another approach which is have become very popular in most recent couple of years and it works better than concatenative technique [13]. This technique is used with HMM or model which are firmly related to HMM e.g. Deep Neural Networks (DNN). HMM based statistical parametric speech synthesis has gained popularity because of its ability to produces high quality speech automatically with parametric flexibility, less data and resources [14].

1.2 Quality of speech synthesis system

Intelligibility and naturalness is the measure of quality of the synthesized speech [15]. There are lots of experimentation over naturalness of voice as a result of TTS systems. In today's world, different segments are recorded and then concatenated for completing a message. A collection of speech words is collected and maintained in database by using a reader who reads large series of text. In these kind of systems, to maintain the consistency the speaker speaks in a single style and keep in mind the distance from microphone and other factors to avoid the inconsistency. This type of TTS system is not required at all as the need is to have a system which can be expressive and convey message with proper expressions and styles. Work is performed to build a system that can convey the message according to the needs of the users. A single style of communication can lead towards wrong messages and can cause other problems of misunderstandings. For example, it is not appropriate to convey a good news and bad news in a same style and manner. Similarly, it is not acceptable to ask a question in neutral way of communication [16]. Numerous strategies like neural networks and linear regression were applied to get the enhanced outcomes. Concatenation techniques are applied to get fully expressive and stylish messages for end users. By using concatenation technique, users can customize, add styles and expression through provided Speech Synthesis Markup Language (SSML) [16]. In speech, Timing at which certain event occurs is also very important as in speech signals, it is affected by some contextual factors like phone identity factors. These factors make it difficult to control timing of events [17]. There are some approaches which have been proposed to control timing of events like linear regression [18] and tree regression [19]. A new technique is proposed in [17] where timing of events is controlled by multi-dimensional Gaussian distribution based Hidden Markov model.

1.3 Architecture

TTS system is a way of communication and transferring information using words and styles of speaking [16]. It has two processes which are text processing and speech generation. In text processing, the given input text is processed so that to get appropriate chain of phonemic units. Speech generator takes these units as input and converts them into

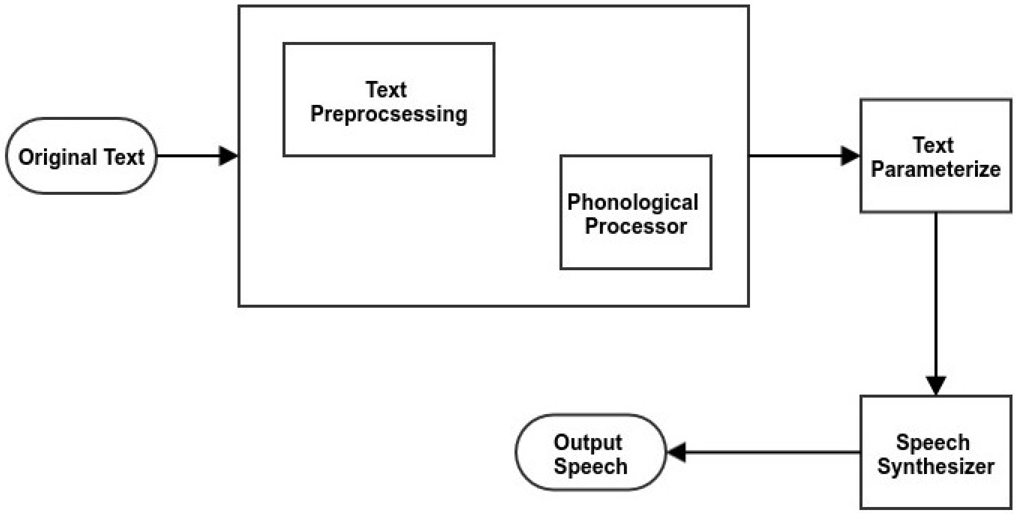

Figure 1.2: Architecture of TTS

synthetic speech by selection of a unit from large corpus TTS system for small database is easier to implement but not in good quality [14], [20], [21]. Different researchers and developers used different strategies and architecture to develop TTS system. In [22], raw text is converted into intelligible speech signals by following two sub processes called High-level and Low-level synthesis. High-level synthesis converts textual data into phonetic strings and Low-level synthesis converts these strings into speech signals [22]. In [23], TTS system is divided into three modules.

1. Natural language processing (NLP)
2. Text parameterization
3. Speech synthesis

NLP unit converts text into phonetic strings. The second and third stages use these phonetic strings and convert them into speech signals. This is shown in figure **??**.

In [24], TTS system is implemented by following four modules in sequence.

1. Text analysis
2. Word pronunciation
3. Phonetic interpretation
4. Speech signal generation

In text analysis, the input text is segmented into sentences and then dividied into words. These words are then categorized according to their syntactic and contextual meaning. The numbers and abbreviations are also processed in this step. In word pronunciation process, words are represented by respective phonetic notations by using word pronunciation dictionary. In phonetic interpretation the duration of phonetic segments, pitch and accents are assigned. Signal generation component of TTS system takes output from all above processes and generate a signal of speech using a function. In [25], TTS system is divided in two parts. One is called NLP unit and other is called Speech Synthesis unit. NLP unit preprocess text and converts it into phonetic strings. These phonetic strings are then marked by stress marker and passed to speech synthesis unit which converts it into speech signals.

Chapter 2

Related Work

Speech synthesis system mostly use different electronic signal processing techniques to process speech signals. But people have been working on speech synthesis before electronic signal processing techniques. In beginning, people tried to build machines with mechanical devices which were used to create human sound. After the development of computers, better systems were built using different techniques. In [15], basic speech synthesizing technique are discussed which works by concatenation of small recorded speech segments called phonemes to form complete speech. Words are first separated into syllables which is the unit that collectively depicts correct pronunciation of each word. Recorded speech of each syllable is then joined to get complete pronunciation of the word. The resulting voice need some post processing as it has some delay due to concatenation of small units. These delays are removed in post processing which gives final pronunciation of the word. Preprocessing of the input text, guessing correct pronunciation of each word and prosody are the problems which make this system difficult. In preprocessing of text, all abbreviations and any type of numbers, time or dates are replaced with their corresponding full words. Other issue is the speculating pronunciation of a word with respect to its context. For example, pronunciation of "lives" will be different in "He lives in Lahore" and "He saved two lives". Stress and intonation is also important and complex part of speech synthesis system in order to get naturalness in synthesized speech. A text to speech (TTS) system for Azerbaijani language is developed using concatenative synthesis in [5] where small recordings were concatenated to make speech waveform.

Two sub parts of TTS system text analysis and word pronunciation were discussed in

[24]. TTS system is divided in 4 sub parts: text analysis, word pronunciation, phonetic interpretation and signal generation. Text analysis step includes division of text into sentences, words, phrases and expansion of abbreviations etc. Text analysis is required to get correct context and pronunciation of each word in a sentence. In text analysis, sentence division and parts of speech tagging is done by using heuristic solutions [26] and dynamic programming respectively. For word pronunciation, dictionary based approach was used for about 99.9% words and for 0.1% words, letter to sound rules were followed.

A technique based on letter to sound rules is also developed in [27]. Total 329 letter to sound rules have been created. These rules take text as input and translate it into phonetic alphabet which in turn converted to synthetic sound. This system produces about 97% correct pronunciation of phonemes. The paper also describes software and hardware requirements and overall performance statistics of this system. The dataset is developed by extracting 50,000 words from Brown corpus [28]. The system gave accuracy of about 93%. A more improved system was proposed in [6], [29]. In [6], a TTS system is developed for English and simple numerical and algebraic expressions. The system is rule based system having 500 letter to sound rules. However, it can use pronunciation dictionary of 1500 words for exceptions. The system interface provides facility of selection of voice types (Male or Female). The word boundaries and probable position of phrases and clauses is analyzed by syntactic analyzer. The phonemes of word are then passed to synthetic speech synthesizer that converts phoneme to sound by following extensive set of rules and rules for consonant-vowel transitions. The whole process is divided in two sub processes, text analysis which is conversion of text to corresponding linguistic representation that comprises of phoneme, stress and boundaries and positions of respective vowel or consonant and durational phenomena such as pauses, interaction between segments [30] and speech synthesis in which sequence of phonemes are converted to speech sound with the help of some set of rules. The abbreviations in input data are converted to their respective text and if dictionary does not have their respective words, the abbreviation is pronounced as a word. After preprocessing, the words are exposed to letter to sound rules. If an unstressed function appears in text and there is no rule for it then it is passed to pronunciation dictionary. The results show that about 95% rules were successful when executed. The syntactic analyzer then determines the structure of sentence according to its pauses and boundaries. The phoneme to speech rules are divided into two components, phonological that provide information of stress and rhythm and duration of words in a sentence. All these outputs are then passed to synthesizer that produces synthetic sound. Synthesizer is simple version of synthesizer proposed in [31].

In [29], TTS system is presented which require some hardware resources as well with the enhancements in microprocessor, memory and signal processor technology. This TTS system can be put into portable form and can be used anywhere with different systems. A higher level language is developed in laboratory that can be easily used for linguistic processes. These enhancements and developments in hardware and software level made transforming of text to speech at 250 wpm rate. This combination of hardware and software is tested against several applications used for handicaps. The fundamental reason for this framework is to make a system that will be utilized for transformation of any language from text to speech.

In [12], Whistler which is a TTS engine is developed by using prosody and concatenative speech parameters that were extracted using probabilistic learning methods. The resulting voice of this TTS system appears to be very much real. This system can also help to build TTS system for other languages.

A formant and concatenative synthesis is developed in [32] where small segments of phonemes were concatenated to form whole speech. A recorded speech database of about 6,000 phonetically balanced sentences is used for training of the system. The technique which has been used in Whistler [12] can significantly encourage the way toward making bland TTS system for new speech style. This system supports Microsoft Speech API [33] and requires under 3 MB memory.

In [34], linear regression and unit selection based speech synthesis is designed using ATR Japanese database. In this algorithm, raw text is converted to phonetic strings and against each phoneme, best candidate unit from a huge database of speech units is selected with *Viterbi* search. By concatenating these units, target waveform is generated. These units can be contemplated as state transition network where each unit represents a different state. The cost of the system depends on target cost and cost of the concatenation of units. Each phoneme and unit is denoted by a multi-dimensional feature vector. Weighted difference of target and candidate feature vector is taken in order to measure the target cost. Similarly, the cost of concatenation is also measured by weighted sum of sub-cost of concatenation. Cost function can be trained in two different ways. In Weight Space Search method, units are searched with *Viterbi* and distance between constructed waveform and natural waveform is minimized. In Regression Training, linear regression is used to choose best unit from the list of all possible units for a given phoneme.

Statistical parametric speech synthesis is another approach which is gaining popularity in recent couple of years [35]. This approach works better than concatenative technique on smaller data. On larger data, concatenative synthesis can produce better quality speech. In this technique, HMM or model firmly related to HMM are used for training model over given data. HMM based statistical parametric speech synthesis has picked up notoriety as a result of its capacity to produces top notch speech automatically with parametric flexibility, less data and resources [14].

A multi-dimensional Gaussian distribution based HMM based statistical parametric speech synthesis system was developed in [36]. In this approach, duration models clustering is performed using decision tree based context clustering. The contextual factors are also considered with phone identity factors. Mel-cepstral coefficients are calculated and model is trained by these coefficients. Context clustering technique which is based on decision tree is used for clustering of the context dependent HMMs [37]. In state duration modeling, multi-dimensional Gaussian distributions are used to model HMM. The clustering of the duration models is performed after estimation using clustering technique based on decision tree. By traversing decision tree, all contexts can be searched. Contextual factors which effects timing of events in speech are also taken into account and resultant speech shows that it has good quality and natural timing. For testing of the system, 450 sentences of Japanese are used for training of system. Sampling of speech signal is done at 16 kHz. Feature vector is composed of 25 mel-cepstral coefficients. There were 3030 states and 2984 distributions in output of the system. The listening tests show that synthesized speech has good quality.

A similar system is designed in [17] using HMM and evaluated by taking input from Japanese database. The parameters in this system are generated with HMM. The state sequence fully or partially is hidden due to which iterations are performed for parameter generation and forward-backward algorithm is used for the situation where state sequence is provided. This algorithm, from multi-mixture HMMs, can generate clear formant structure.

A HMM and unit selection based system is proposed in [38] where model is trained with speech database after which excitation and spectral parameters are calculated. These extracted parameters are modeled by context dependent HMMs. Decision tree based context clustering technique is used in order to get correct model parameters which are then used in speech synthesizer for generating speech signals. The speech characteristics can be controlled using these parameters. In [39], HMM and rule based approaches are applied on voices taken from e-learning courses and online lessons for dataset creation and tested by generating voices and given as input to students to interpret it.

Corpus based approach for Expressive Prosody Modeling is applied in [16] where manually produced dataset was used. To evaluate the synthesized speech and expression, the output is given for testing to 32 native English speakers. Test is performed with different types of sentences like bad news, good news and for yes/no and the accuracies we get are 70.2%, 80.3% and 84% respectively.

HMM based approach is used in [8] to construct a speech synthesizer for Xitsonga which is an African language. The dataset used here consists of phone set of consonants and vowels. These sets utilized to set up a set of letter to sound rules to be used in TTS system. The main tool used for speech synthesis is HTS toolkit [40] with other software that support to setup complete environment for speech synthesis. HMM based approach is used in this study because HMM based statistical parametric speech synthesis can be used to synthesize speech waveform without requiring huge dataset for training. The system received acceptability of 92.3%.

TTS system is designed in [7] for Fon language using Multisyn algorithm [41] which consists of Natural Language Processing (NLP) and Digital Signal Processing (DSP) modules. NLP consists of segmentation, Letter-to-Sound conversion and back-off rules module. When a character is not found in known characters' list, back-off rules are applied. DSP module than choose required unit from database of units are concatenate them to form complete speech signals.

In [42], hybrid TTS converter is developed by concatenating benefits of HMM and waveform based TTS system. For developing it, an audio phoneme library is used. The main edge of developed system over other is that it produced more human like voice/speech. The experiments were taking in Matlab and a phoneme library is developed that consists of audio files and dictionary of words with their phoneme. Sentence is taken as input then model parsed it into words. The system analyzes each word, gets its phoneme and combines all phonemes and plays the sound. Waveform for each sound is also presented. This sub-ban speech synthesized approach is obtained by this combination of models that improved the quality of synthesized speech. The quality of speech is not good enough, in future system will be improved to get better controls.

A speech synthesis system is introduced in [43] which uses context-dependent HMM for defining set of subphone units. This system uses context-dependent HMM for defining set of subphone units. These subphone units are then used in concatenation synthesizer. The training data is one hour recorded speech which is used for getting required parameters. TD-PSOLA waveform concatenation synthesizer is then used to generate pronunciation using these parameters. The synthesized speech is very natural and intelligible. This system uses automatic statistical processes to extract segments of speech from large speech carpus. Desired sentence is produced by concatenation of small segments of speech. HMM is trained and used for segmentation of speech database into HMM-state-sized units. A decision tree is constructed by using phonetic context labels which is used for clustering of the training speech into acoustically self-comparable grouped states. This process helps to find most important context effects. The string to be converted into speech is first converted into sequence of phonetic strings which then using decision tree is changed over to speech segments which are used to generate final speech signals. Modified Rhyme Tests [44] were used to compare system with other. Six listeners were used with each give an answer sheet, and they have to mark word from list of provided words which is played during test. The MRT error rate for test was 5.0% and standard error rate was 0.47%. HMM is used for training of the model. The dataset used for training of model is recorded speech. Four datasets are used in which are termed as M2, M3, F1 and F2 where M stands for male and F stands for female. Six listeners evaluate output produced by model. The MRT error rate and standard error rate for test was 5.0% and 0.47% respectively. In future, segment selection algorithm used in the system can be improved where segments in each state would be available in speech synthesis process. The process of searching optimal segment sequence can be improved by using dynamic programming.

[45] present a TTS system which is based on HMM which comprises dynamic features. Speaker adaptation technique [46] and speaker interpolation technique [47] can be used to modify voice characteristics of speech in statistical parametric speech synthesis system. The HMM based statistical parametric TTS system can model speech parameters like spectrum or excitation with the help of context-dependent HMM and construct speech signals. Version

2.0 of already build HMM based TTS system (HTS) toolkit is presented in [20]. HMM based speech synthesis system can build speech synthesis system even with small dataset for training [48] but the quality of that speech will not be equal to recorded speech.

A more advance technique is Neural Networks based technique as it works better than HMM based technique. Time domain Neural Networks with database containing sounds of words called phonemes is used in [49]. The basic flow of the system involves speech recording, speech labeling, voice coder and input processing using Time Delay Neural Network. The figure 2.1 shows the block diagram of system.

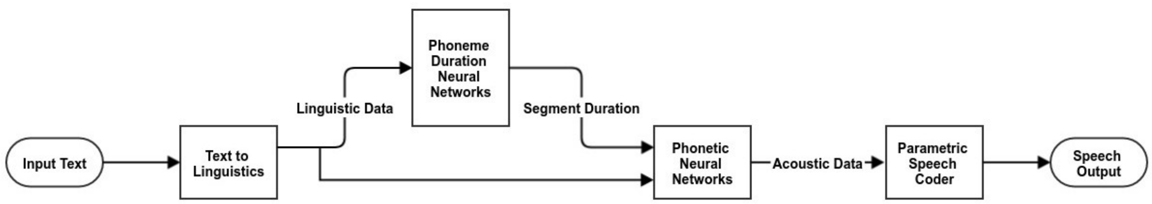

Figure 2.1: Time Domain Neural Networks based TTS System

Neural Networks based techniques are used to learn features automatically during training along with the combination of various techniques like linear regression and Neural Networks in [50]. Dataset of all previous Blizard Challenges [51] and afterwards up to 2013 were used. Model was evaluated using 5-fold cross validation. The given model gave 0.11% and 0.17% error for LR+LR and LR+NN respectively. Neural Networks is also used in [52] with dataset consisting of 328 hours which was collected in voice of native 1506 speakers. The model is tested by giving 20 sets of randomly selected from evaluation set and asked them to rate output of each set between 0 and 100.

Deep Neural Networks is applied in place of HMM in [53] as HMM based system cannot model complicated context dependencies. There are impediments in HMM based system that Deep Neural Networks (DNN) can cover and also likewise beat HMM based system.

Recurrent Neural Networks (RNN) is applied in [54] by using the Bidirectional Long Short Term Memory (BLSTM) with dataset consisting of 5000 training utterances and 200 utterances for testing the system. Whole recording was done in voice of the female native speaker. Objective and subjective evaluation measures are used to find distortion between natural and synthesized speech and quality respectively which shows that hybrid system is better as it gave 44%, 59% and 55% accuracy whereas the Neural Networks, HMM and DNN gave 29%, 22% and 20% accuracy respectively. In [55] Recurrent Neural Networks (RNN) postfilters are used for speech synthesis.

Researchers have also been working on TTS system for Urdu language for many years. A bi-lingual TTS synthesis system for Urdu and Sindhi is designed in [56] using bilingual hybrid knowledge based approach by using concatenated synthesis method which is capable of providing high quality Urdu and Sindhi speech. This system can be further expanded to include sensitive and visual text-to-speech (VTTS) policies in future.

In [57], an HMM based speech synthesis system is developed for Urdu. The speech corpus is created by recording 1 hour and 15 minutes of speech containing 989 sentences in total. In training phase, context level and prosody level parameters are extracted from recorded sentences e.g. counts, position, distances, stress and phone utterance information. F0 excitation parameter and mel-cepstral coefficients are calculated using RAPT [58]. The F0 is modeled using frequency distributions discrete for unvoiced and continuous for voiced regions. These HMM models are then clustered using decision trees. In text analysis, numerals and abbreviations in input text are preprocessed and converted to full textual forms. The date/time and numeric notations are processed using regular expressions (rule based component) and abbreviation are converted to text by finding their corresponding words from dictionary. This stage is followed by diacritic restoration stage which used dictionary develop by CRULP [59] to restore diacritics. After this, G2P converter which follows guidelines of [60] is used to convert grapheme to phoneme. In synthesis module, input text is labeled and then by speech synthesis algorithm, it generates speech features which are passed to filter and obtain speech signal. The model uses 36 consonants and 10 vowels. All speech recordings in a corpus are converted to their phoneme representation and saved in pronunciation dictionary after checking by author. After this, STRAIGHT vocoder is used to estimate speech parameters and generation of speech waveform. 680 questions were gathered by using spectrum and context features, speech parameter were generated using maximum likelihood criteria. For evaluation of system, the author listened to synthetic speech himself and found that it is not intelligible but can be improved in future work.

In [61], a TTS system for Urdu is developed by using HTS toolkit and Urdu Qaida of grade 2 and 4. This system consists of two processes, text analysis and synthesis of speech. Here feature extraction and calculation of mel-cepstral coefficient is done using technique mentioned in [62], text processing is done using [22] and process of synthesis using process described in [17]. The HTS toolkit is available for English, Japanese and Portuguese languages. For Urdu, certain modifications are needed which involve creation of context level labels and questions file for Urdu phoneme set. Frequently speaking Urdu words were identified by using greedy algorithm and question files are made to deal with the issue of data sparsity as in a model, only a certain amount of examples can be handled during training phase. If we look on to the contextual level, it is observed that multiple contextual occurrences exist for a single phoneme. To deal with this problem, a clustering method is used to cluster similar acoustic words. The whole training set is placed into single cluster and on the basis of each question, cluster is split. The cluster is chosen on the basis of the question which minimizes the objective function. In the evaluation process, experiment is performed by using 200 frequent Urdu words and native Urdu speakers. Testing of the system shows that the system gives output which is intelligible but not very natural. The reason behind this is data used in training phase consists of full sentences rather than words. Performance with respect to naturalness can be improved by using words instead of sentences because of clarity and length of word. It is found that 92.5% words are correctly identified. The system has taken 66 phonemes but for better performance at least 270 examples should provide the system during training phases.

Natural Language Processing unit is very importing unit in speech synthesis system as it handles all language related issues. This unit performs a list of steps for its complete operation which contains tokenization, semantic tagging, string generation, syllabification, stress and intonation marking etc. [11], [25] discuss such unit for Urdu language. This unit is divided in two parts called as preprocessing and phonological processing unit. Preprocessing unit process all numbers, dates and time in input data and converts them into their respective words. For example, 1000 and 6-10-2012 will be changed into ہزار and چھے اکتوبر دو ہزار بارہ respectively. All special characters like $ and are also processed in this unit. In last step of preprocessing unit, data is passed through grapheme to phoneme converter. In phonological processing unit, syllable boundaries, stress and intonation are marked by their respective markers.

[11] discussed consonantal and vocalic sounds for Urdu Language in detail. In [23], phonological processing unit for Urdu language is discussed in detail. In this module, word

boundaries are marked by tokenizer which is called text normalization. This is followed by marking of syllable boundaries by using letter to sound rules. The syllabified data is processed to apply sound change rules. This is followed by stress and intonation marker. In [63] a statistical based part of speech tagger for Urdu language is discussed which works by calculating probability of each word given a particular tag. Unigram model assign tag for each token that has the maximum probability. Conditional probability for given tags against each word using maximization principle as used in [64], [65]. The model is evaluated by comparing Unigram, Bigram and Backoff experiments with different size of tag sets. t-test, POS accuracy are used to measure performance. Bigram model considered maximum likelihood principle keeping an eye on the context of text. Backoff model was used to blow away sparse problems. Problems in Urdu segmentation are discussed for Urdu in [66]. Clause boundary identification is discussed in [67] using clause markers and classifier in Urdu language using conditional random field as a classifier.

2.1 Discussion

Raw text can be converted into speech by concatenation of small units of speech from a huge single-speaker speech database. Huge database makes it possible to produce more natural sound. TTS system development can be based on rules for generation of speech but this method can take intensive labor and rules are difficult to be general so that they can be used for other languages as well. In prosody modeling, linguistic rules are used [4], [68] but speech produce by this approach felt to be robotic. So in order to increase quality of voice, large units are used. This strategy enhanced naturalness as well as diminished expected time to create new voice and furthermore made the manufactured speech like unique benefactor speaker.

The best approach for speech synthesis until now is considered to selection synthesis but it has certain limitation that is it need large database of recording which is very expensive and not feasible for certain languages [34], [69], [70]. Statistical parametric speech synthesis is becoming popular and being used for number of languages like English [38], Chinese [71], Arabic [72], Croatian [73] and Urdu [57]. The advantage of parametric over selection is that it does not require saving original signal for synthesis due to which database is small for this approach [74]. Basic TTS system focus over conversion of text to voice using multiple techniques [13]. Different synthesis model has been developed but HMM is becoming popular from last few years [53]. There are multiple tools for TTS but freely available tools mostly use 2 techniques i.e.

1. HMM based speech synthesis called SPSS

2. Simple waveform concatenation.

SPSS technique is attractive although its results are comparatively not amazing. In recent years, the use of statistical modeling in speech recognition system has increased a lot and most of these systems are using HMM for acoustic modeling of the system [75]. These systems enable us to construct models with large amount of data that is difficult to analyze manually. This technique can be applied on the process of speech synthesis. This type of system can be used to run on different data, voices and languages [75]. There are many TTS systems which are capable of generating high quality speech, but they cannot generate speech with different speaking style and voice because speech data required by these systems in order to get these characteristics is very large. HMM based speech synthesis system [38] is capable of doing this without needing large speech database. A speech synthesis system can be developed using statistical learning techniques. These systems can be trained and voice characteristics of original speaker can be produced in synthesized speech. This type of system can be built with HMM and its performance can be improved by techniques like context-dependent modeling and environment adaptation techniques [17]. It has many advantages like ability to change voice characteristics and robustness which will be very difficult in concatenative speech synthesis. But it has some limitations like inefficiency in handling complicated context ascendance [53].

Another famous speech synthesis approach is unit selection in which small units of recorded speech are concatenated in order to synthesize speech waveform. This technique can generate high quality speech signals but for getting various characteristics of synthesized speech, a huge database is required. On the other hand, this can be achieved using HMM based statistical parametric TTS system without needing large speech database [20]. A considerable measure of research chip away HMM approach but the output voices produced by these kinds of systems look unnatural sometimes. It is surprising that by the time this should be improved a lot but there are still existing problems and drawbacks that decrease the performance of TTS system, as compared to other simpler concatenation based TTS systems. Through literature review it is easy to say that HMM system do over smoothing which cause unnaturalness for TTS System output. There is no proper study which can prove this hypothesis so [13] present the reasons for this unnatural behavior [13].

Many TTS systems have been purposed and each have its own pros and cons. For example, waveform based model is good enough to produce human like sound but it requires large database. In rule based techniques, most of the time rules updating is required and novelty is too much difficult with traditional rules. Similarly, with concatenation of phonemes, it is also difficult to bring novelty and handle new and unseen words [49], [76]. Neural Networks can be used to improve results of speech synthesis system [55]. From recent years, Neural Networks are being used as acoustic models [77], [78]. There exists a wide research over the correlation between acoustic modeling and linguistic features in late 90s [79]. Now more focus is not Neural Networks based techniques. Neural Networks easily map linguistic features to acoustic models using feed forward approach [53], [80]–[82].

Chapter 3

Methodology

Text to speech (TTS) synthesis is the process of transformation of data from textual form into voice output. This process is divided in two sub modules. One module performs analysis and preprocessing of data and other transforms processed data into sound signals. This is shown in 3.1

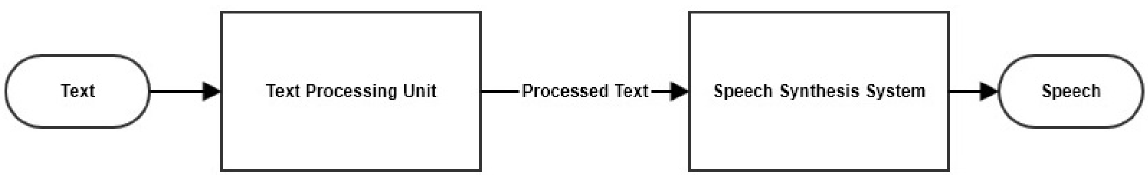

Figure 3.1: TTS Sub Modules

3.1 Text Processing Unit

Text processing unit is responsible for processing of text before it is sent to speech synthesis system. It finds numbers, dates and time in input data and converts it into format acceptable by speech synthesizer. This module consists of following sub modules.

1. Special Character Processor
2. Semantic Tagger
3. Text Generator
4. Text Formatter

3.1.1 Special Character Processor

Raw text may contain special characters such as punctuation marks. These characters help to understand context of a word but are not converted into sounds. We remove all such characters from text before further processing. Another processing which is done is conversion of all arabic numerals like ۱, ۲ and ۳ into their corresponding characters like 1, 2 and 3 because it is easy to further process text after it has been converted into same type of numerals.

3.1.2 Semantic Tagger

The purpose of Semantic Tagger is to identify numbers, dates and time from input data and give them proper tags. All numbers in input data are converted into arabic numerals of type 1, 2 and 3 in previous step. There can be multiple form of numbers, dates and time. These forms are explained below.

1. Dates in following format
   a. 12/11/2018 or 12/11/18 with different separaters like "/" or "-" or "."
   b. 12 دسمبر 2012
   c. 12 دسمبر
2. Number in following format
   a. Whole Numbers such as 123
   b. Floating point numbers such as 12.3
3. Time in following format
   a. 12:12
   b. 12:12:12

In table 3.1, regex used for identification of these numbers, dates and time are shown.

Regex	Type	Example
(\d+(?:\.\d+)?)	Integer or Floating point number	123 or 12.312
\d{1,2}:\d{1,2}(?::\d{1,2})?	Time with or without seconds	5:12 or 5:12:10
\d{1,4}[./-]\d{1,4}[./-]\d{1,4}	Date with separator like "/" or "-" or "."	12-10-2018 or 12/10/18
%s \d{4}	Dates with month name in Urdu. This is checked by replacing %s with each month name separately.	12 دسمبر

Table 3.1: Regular Expression for Semantic Tagger

Word	Converted Text
123	ایک سو تئیس
1231	ایک ہزار دو سو اکتیس
123.1234	ایک سو تئیس اعشاریہ ایک دو تین
12345	بارہ ھزار تین سو پینتالیس
1234567	بارہ لاکھ چونتیس ھزار پانچ سو ستاسٹھ
987654321	اٹھانوے کروڑ چھہتر لاکھ چون ھزار تین سو اکیس
143.159874	ایک سو تینتالیس اعشاریہ ایک پانچ نو آٹھ سات چار

Table 3.2: Number Conversion example

3.1.3 Text Generator

Semantic tagger will return all numbers, dates and time from text with each one marked as date/time/number. Text generator part will take each word and generates Urdu text according to its tagging. Each tagged number is handled by specific text converter. These converters are listed below.

1. Number to text converter
2. Date to text converter
3. Time to text converter

3.1.3.1 Number to Text Converter

This unit will deal with whole numbers, fractional numbers and decimal numbers. In table 3.2, example conversions are shown.

Different process is performed in order to handle fractional and integer parts of floating point numbers. The algorithm for integral number is described below.

This algo will return list of factored numbers and factors. For 1213, the result of this algorithm will be [1, 1000, 2, 100, 13]. The integer to Urdu mapping for number 0 to 100 and number like 1000, 100000 etc are stored in CSV file. The factored number list will then

Date	Converted Text
12/10/15	دس دسمبر دو ھزار پندرہ
12.10.15	دس دسمبر دو ھزار پندرہ
12-10-15	دس دسمبر دو ھزار پندرہ
12.10.1989	دس دسمبر انیس سو نواسی

Table 3.3: Example Date Conversions

be converted into text by using integer to Urdu mapping. In table B.1, Urdu mapping of each possible factor is listed. Each number in fractional part of floating pointing number is replaced by their respective mapping from above list. Both the mappings of integral and fractional part of number are then joined to get complete text of input number. This module is very important module as it is also used in date and time conversion.

3.1.3.2 Date to Text Converter

The unit deals with date in following formats

- 12/12/2012
- 12/12/12
- 12.12.2012
- 12.12.12
- 12-12-2012
- 12-12-12
- 12 دسمبر 2012

All dates will be converted into common format e.g. بارہ دسمبر دو ہزار بارہ . Some example conversions are shown in table 3.3.

The word tagged as date is first processed to get year, month and day of the month. Day and year are then passed to number to text converter and month is converted to its corresponding mapping. This mapping is saved in CSV file. This is shown in table B.2. In Urdu, in dates, we have different notation for year e.g. 1980 in number is spoken as ایک ھزار نو سو نواسی (One thousand nine hundred and eighty nine) but in dates, it is spoken as انیس سو نواسی

Time	Converted Text
1:12:15	ایک بج کر بارہ منٹ اور پندرہ سیکنڈ
7:45	سات بج کر پینتالیس منٹ

Table 3.4: Example Time Conversions

(Nineteen hundred and eighty nine). This is also handled during the process of date to text conversion.

3.1.3.3 Time to Text Converter

Time can occur with seconds or without seconds in text. It is written in 1:11:12 or 1:12 format. All words tagged as time will be converted into text by separating hour, minutes and seconds from time. Each value will be converted into Urdu text by using number to text converter. All these values are combined to make complete time text. Table 3.4 shows example conversions.

3.1.4 Text Formatter

The purpose of formatter is to replace all number, dates and time with their corresponding Urdu text returned by Text generator. This process is performed in following order

1. Word tagged as dates are replaced by their corresponding text
2. Word tagged as time are replaced by their corresponding text
3. Word tagged as number are replaced by their corresponding text.

The output of all above processes will be the text which will only contain Urdu text which can now pass to the speech synthesizer which will convert it to speech signals.

3.2 Speech Synthesis System

3.2.1 Tools

Below are the tools used for the process of Speech Synthesis of Urdu.

1. Speech Tools Library of Edinburg

2. Festvox

3. Speech Signal Processing Toolkit (SPTK)

4. Festival

3.2.1.1 Speech Tools Library of Edinburg

The Edinburgh Speech tools is collection of utilities used for speech processing. These utilities cover major tasks such that reading and writing speech waveforms, parameter files(F0 and LPC etc). The speech tools also include executable programs which can be used in user defined programs.

3.2.1.2 Festvox

Festvox is a tool which can be used to build synthetic voices. This includes scripts for building voice in other languages.

3.2.1.3 Speech Signal Processing Toolkit (SPTK)

As name suggests, this tool is for processing speech signals in UNIX systems.

3.2.1.4 Festival

Festival [83] is a speech synthesis system which is developed in Centre for Speech Technology Research (CSTR) [84]. It is a multi-platform framework for building speech synthesis system. This system is outlined so that it tends to be utilized for following purposes

1. Improvement in speech synthesis system

2. Developing speech synthesis applications

One of the main thing that makes Festival very useful is scripting language which is based upon Scheme programming language. This can be used to manage parameters and flow of control in Festival.

3.2.2 Process

The process of speech synthesis is based on statistical parametric speech synthesis. The statistical parametric speech synthesis system is based on models like HMM or DNN in which recorded speech is used to train model. In this method, speech is elaborated with parameters which are defined by statistics. This is why it is called as statistical parametric speech synthesis.

In CLUSTERGEN statistical parametric speech synthesis, model is trained and used for synthesis in Festival Speech Synthesis system.

3.2.2.1 Preparing Data

For training purpose, Phonetically Rich Urdu Speech Corpus [85] is used. This data consists of recordings of 708 phonetically rich sentences, 10,101 tokens with 5,656 unique words. Total duration of recording is 70 minutes.

3.2.2.2 Data Labeling

Data is labeled in specific format which is required by FestVox for training purpose. The is labeled in following format.

(c1 "نیلم نے سالگرہ پر ہیڈ سیسموگراف اسود قریشی کے ماتھے پر اینٹھن اور غم کی آتشیں رو محسوس کی")

Where c1 is the name of recording file and text between quotation marks is corresponding label of that recording.

3.2.2.3 Training Data

The whole data is further divided in 10:1 ratio in training and test set respectively.

3.2.2.4 Urdu to Hindi Transliteration

The underlying system of Urdu TTS is Hindi TTS system. So all the alphabets are mapped in their corresponding Hindi alphabets. In this way, text is first converted into corresponding Hindi text using that mapping and then it is converted into sound. Mapping of each Urdu word with Hindi in this system is shown in table B.3.

3.2.2.5 Data Labeling

The first stage of training is to label speech database using HMM labeler. We are using EHMM labeler which is provided in FestVox. In this process, Baum-Welch is used to train context dependent models. This labeler works in 8 steps. Prompt files are extracted from utterance structure of Festival.

- Unique sequence of phones are extracted and stored.
- List of wav files is collected for feature extraction.
- From wav files, cepstral coefficients (LPCCs and MFCCs) are extracted.
- From cepstral coefficients, deltas and delta-delta features are generated.
- By using generated features and wav files list, features vectors are modified.
- Phones list generated in step 2 and wav file list is used to modify prompt list.
- Hidden Markov model is trained using Baum-Welsh algorithm till difference in the average likelihood is less than 0.001.
- Labels are generated according to training data.
- Integer indices of labels are converted into phone names

3.2.2.6 Building Utterance Structure

Utterance is the essential building unit of Festival. It shows relation between bunch of items where each item relates to word, syllable or segment etc. Below are the some of the relations used in building utterance structure.

- **Text**: It consists of strings to be processed and features of that string.
- **Token**: Token means each word in a sentence is separated by some language specific separator.
- **Word**: A small unit of speech which can be pronounced with the help of letter to sound rules of a language.
- **Phrase**: Phrase means group of words forming a part of a sentence.

- **Syllable**: Syllables are units which when combined with vowels form complete pronunciation of a word.

- **Segment**: Segment consists of list of phones.

- **SylStructure**: This is a tree structure which is formed with word, syllable and segment.

- **IntEvent**: These are array of syllable related intonation events.

- **Intonation**: Intonation means rise and fall in speech signals.

3.2.2.7 Coefficient Extraction

Coefficient extraction is the process of extracting parameters like F0, mcep and voicing coefficients using SPTK. This is done by generating F0 and mcep coefficient. These parameters are then combined to make final parameter files. This is a lengthy process which can take lot of time depending on size of training data.

3.2.2.8 Building the Model

All the data generated above is used to train and build HMM-state duration model. This process works in following steps.

1. Statenames Generation

2. Parametric Model generation

3. Duration model generation for statenames

This resulting model can be used to perform TTS synthesis process.

Chapter 4

Experiments and Results

The reason for a Text to Speech system is to manufacture a system which is equipped for producing voice as near human voice as could reasonably be expected. The generated voice should be intelligible so that people can easily understand generated voice. To find quality of generated sound, every speech synthesis system is evaluated. The evaluation process can be subjective as well as objective. In subjective evaluation, system is evaluated using human users while in objective evaluation, different algorithms are used. For the process of evaluation, native speaker of specific language is required. For our system, we only focused on subjective testing.

4.1 Subjective Testing

There are many type of subjective tests. Some of them are listed below.

- Diagnostic Rhyme Test (DRT)
- Modified Diagnostic Rhyme Test (M-DRT)
- Naturalness Test
- Intelligibility Test
- Usability Test

4.1.1 Diagnostic Rhyme Test (DRT)

DRT is performed in order to do indicative and relative assessment of the understandability of single starting consonants. This is conducted with words which are similar in sound but differ with each other in initial consonants [86]. User has to listen speech generated by system of a specific word and identify that spoken word from list of words. The result of this test is the percentage of words correctly identified.

4.1.2 Modified Diagnostic Rhyme Test (M-DRT)

This test is to check demonstrative and relative assessment of the coherence of single last consonants. This is conducted using words which are similar in sound but differ with each other in last consonants [44]. User has to listen speech generated by system of a specific word and identify that spoken word from list of words. The result of this test is the percentage of words correctly identified.

4.1.3 Naturalness Test

This test is directed to discover to which degree created voice is near human voice. This is conducted by rating generated voice from 1 to 5. User will play some voice and will give synthesized speech some value from 1 to 5 according to his understanding of the speech.

4.1.4 Intelligibility Test

This test is conducted to find out to which extent generated voice is understandable. This is conducted by rating generated voice from 1 to 5. User will play some voice and will give synthesized speech some value from 1 to 5 according to his understanding of the speech.

4.1.5 Usability Test

This test is conducted to find out to which extent generated voice can be used for blind or non-blind people. This is conducted by rating generated voice from 1 to 5. User will play some voice and will give synthesized speech some value from 1 to 5 according to his understanding of the speech.

4.2 Evaluation

For the process of evaluation, we selected list of 64 words and 8 sentences. Words are selected on the sound and first and last words in order to use in DRT and M-DRT.

4.2.1 Methodology

An evaluation form is designed which have three sections.

4.2.1.1 DRT Section

This section has 8 questions. In each question, user will play recording of some words which are converted to sound using our TTS system. These words are tested through following carrier sentence.

نیچے دیے گئے الفاظ میں سے حرکت پے نشان لگائیں

User will have to select that word from list of words which have same sound but different first character.

4.2.1.2 M-DRT Section

This section has 8 questions. In each question, user will play recording of some words which are converted to sound using our TTS system. These words are tested through following carrier sentence.

نیچے دیے گئے الفاظ میں سے حرکت پے نشان لگائیں

User will have to select that word from list of words which have same sound but different last character.

4.2.1.3 Mean Opinion Score (MOS) Section

In this section, user will have to play a sentence and user will rate converted text from 1 to 5. User will have to rate them on the basis of following properties.

- **Naturalness**: How much converted sound is near sound delivered by a human?
- **Intelligibility**: How conveniently the word was perceived?

Test	Score
Diagnostic Rhyme Test (DRT)	0.95
Modified Diagnostic Rhyme Test (M-DRT) Section	0.88
Naturalness	3.24
Intelligibility	3.42
Usability	3.49

Table 4.1: Evaluation Result

- **Overall**: How do you rate this sound overall? Is this system is usable to use for blind people?

System is evaluated by 47 (33 males and 14 female) native Urdu speakers who carefully listened and evaluated system. Each listener evaluated each question separately after listening it.

4.3 Results

The results of evaluation were very satisfying as it is observed that most of the words which have same sound are easily recognizable. Almost 90% of the such words are correctly identified by users. The result shows that output of the system is recognizable and intelligible but not very natural. Table 4.1 shows the complete result of evaluation.

Chapter 5

Conclusion

A speech synthesizer system is built for Urdu using Festival speech synthesis system [83] and and Festvox [87]. We used recorded speech of 70 minutes which consists of over 700 sentences. It contains almost 10000 tokens and over 5500 unique words. This data is divided in testing and training data in 10:90 proportion and used for training of the model to be used in synthesize speech. We used lexicon of Hindi for Urdu in order to build this system. A text preprocessing unit is also developed which process any number, dates or time string in input text.

The resulting system is tested by 47 native Urdu speakers and found to be intelligible but not very natural. The synthesized words having same sound but having different first or last character are also recognizable. The resulting voice is intelligible and usable but it not very close to natural voice. It is because system is developed using Hindi lexicon and letter to sound rules. Some character in Hindi are spoken differently as compared to Urdu. For example, غ in Hindi is spoken as گ. Apart from this, the size of training data also effects quality of synthesized speech. The quality of synthesized speech also depends on size of speech corpus used. In current development of the system, we did not consider context of word in input data which also effects pronunciation of the word. In future, we can use parts of speech tagger in order get correct pronunciation of each word.

Appendix A

Figures

Appendix B

Tables

Number	Mapping
.	اعشاریہ
0	زیرو
1	ایک
2	دو
3	تین
4	چار
5	پانچ
6	چھ
7	سات
8	آٹھ
9	نو
10	دس
11	گیارہ
12	بارہ
13	تیرہ
14	چودہ
15	پندرہ
16	سولہ
17	سترہ

Number	Mapping
18	اٹھارہ
19	انیس
20	بیس
21	اکیس
22	بائیس
23	تیئیس
24	چوبیس
25	پچیس
26	چھبیس
27	ستائیس
28	اٹھائیس
29	انیتس
30	تیس
31	اکتیس
32	بتیس
33	تینتیس
34	چونیتس
35	پینتیس
36	چھتیس
37	سینتیس
38	اڑتیس
39	انتالیس
40	چالیس
41	اکتالیس
42	بیالیس
43	تینتالیس
44	چوالیس
45	پینتالیس
46	چھیالیس
47	سینتالیس

Number	Mapping
48	اڑتالیس
49	انچاس
50	پچاس
51	اکیاون
52	باون
53	ترپن
54	چون
55	پچپن
56	چھپن
57	ستاون
58	اٹھاون
59	انسٹھ
60	ساٹھ
61	اکسٹھ
62	باسٹھ
63	تریسٹھ
64	چونسٹھ
65	پینسٹھ
66	چھیاسٹھ
67	ستاسٹھ
68	اٹھاسٹھ
69	انہتر
70	ستر
71	اکہتر
72	بہتر
73	تہتر
74	چوہتر
75	پچھتر
76	چھہتر
77	ستتر

Number	Mapping
78	اٹھتر
79	اناسی
80	اسی
81	اکاسی
82	بیاسی
83	تراسی
84	چوراسی
85	پچاسی
86	چھیاسی
87	ستاسی
88	اٹھاسی
89	نواسی
90	نوے
91	اکانوے
92	بانوے
93	ترانوے
94	چورانوے
95	پچانوے
96	چھیانوے
97	ستانوے
98	اٹھانوے
99	ننانوے
100	سو
1000	ھزار
100000	لاکھ
10000000	کروڑ
1000000000	ارب
100000000000	کھرب

Table B.1: Number Mappings

Number	Mapping
january	جنوری
february	فروری
march	مارچ
april	اپریل
may	مئی
june	جون
july	جولائی
august	اگست
september	ستمبر
october	اکتوبر
november	نومبر
december	دسمبر
1	جنوری
2	فروری
3	مارچ
4	اپریل
5	مئی
6	جون
7	جولائی
8	اگست
9	ستمبر
10	اکتوبر
11	نومبر
12	دسمبر

Table B.2: Month Mapping

Hindi Character	Urdu Mapping	Character Detail
◌ँ	ں	Noon Ghunna
ऄ	َ	Arabic Zabar or Fatha
न	ً	Arabic Fathatan

Hindi Character	Urdu Mapping	Character Detail
अ	ا	Alif
ओ	ا	Alif
अ	ء	Hamza
अ	ٔ	Hamza Above
अ	ع	Ain
आ	آ	Alif Madda
इ	ِ	Arabic Kasra or Zair
ई	ی	Yeh
उ	ُ	Arabic Damma or Paish
◌ू	ؤ	Waw with hamza above
ऊ	ؤ	Waw with hamza above
ऋ	رِ	Reh with Zair
ए	ے	Baree Yeh
ऐ	آے	Aaey
ओ	ؤ	Waw with hamza above
औ	آو	Aao
क़	ق	Qaf
क	ک	Kaaf
ख	کھ	Khay
ख़	خ	Khay
ग	گ	Gaaf
घ	گھ	Ghaa
घ	چ	Chay
छ	چھ	Chhay
ज	ج	Jeem
झ	جھ	Jhay
ञ	یاں	Yaan
ट	ٹ	Tay
ठ	ٹھ	Thay
ड	ڈ	Daal

Hindi Character	Urdu Mapping	Character Detail
ढ	ڈھ	Dhaal
ण	ڈاں	Daan
त	ت	Tay
त	ط	Toain
थ	تھ	Thay
द	د	Dal
ध	دھ	Dhal
न	ن	Noon
प	پ	Pay
फ	پھ	Phay
ब	ب	Bay
ब	بھ	Bhay
म	م	Meem
र	ر	Ray
ल	ل	Laam
ळ	ّ	Arabic Shadda
व	و	Wow
श	ش	Sheen
स	ث	Say
स	س	Seen
स	ص	Saad
ह	ح	Hay
ह	ہ	Gol Heh
ह	ھ	Heh
ग़	غ	Ghain
ज़	ذ	Zaal
ज़	ز	Zay
ज़	ظ	Zoain
ज़	ژ	Zay
ज़	ض	Zaad

Hindi Character	Urdu Mapping	Character Detail
ड़	ڑ	Rhay
ढ़	ڑھ	Rhay
फ़	ف	Fay
य़	ئ	Hamza Choti Yeh
०	0	Zero
१	1	One
२	2	Two
३	3	Three
४	4	Four
५	5	Five
६	6	Six
७	7	Seven
८	8	Eight
९	9	Nine
०	٠	Arabic Zero
१	١	Arabic One
२	٢	Arabic Two
३	٣	Arabic Three
४	٤	Arabic Four
५	٥	Arabic Five
६	٦	Arabic Six
७	٧	Arabic Seven
८	٨	Arabic Eight
९	٩	Arabic Nine

Table B.3: Hindi to Urdu Character Mappings

Bibliography

[1] B. Mumtaz, S. Urooj, S. Hussain, and E. U. Haq, "Break index (bi) annotated speech corpus for urdu tts," in *Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), 2016 Conference of The Oriental Chapter of International Committee for*, IEEE, 2016, pp. 22–27.

[2] P. Khilari and V. Bhope, "A review on speech to text conversion methods," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 4, no. 7, 2015.

[3] B. G. Greene, J. S. Logan, and D. B. Pisoni, "Perception of synthetic speech produced automatically by rule: Intelligibility of eight text-to-speech systems," *Behavior Research Methods, Instruments, & Computers*, vol. 18, no. 2, pp. 100–107, 1986.

[4] D. H. Klatt, "Review of text-to-speech conversion for english," *The Journal of the Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, 1987.

[5] . Aida–Zade, C. Ardil, and A. Sharifova, "The main principles of text-to-speech synthesis system," *International Journal of Signal Processing*, vol. 6, no. 1, pp. 13–19, 2010.

[6] D. Klatt, "The klattalk text-to-speech conversion system," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.*, IEEE, vol. 7, 1982, pp. 1589–1592.

[7] T. K. Dagba and C. Boco, "A text to speech system for fon language using multisyn algorithm," *Procedia Computer Science*, vol. 35, pp. 447–455, 2014.

[8] N. Baloyi, "A text-to-speech synthesis system for xitsonga using hidden markov models," PhD thesis, University of Limpopo (Turfloop Campus), 2012.

[9] (). Top 30 languages by number of native speakers, [Online]. Available: http://www.vistawide.com/languages/top_30_languages.htm.

[10] (). Speech synthesis - formant synthesis, [Online]. Available: https://en.wikipedia.org/wiki/Speech_synthesis#Formant_synthesis.

[11] A. M. Saleem, H. Kabir, M. K. Riaz, M. M. Rafique, N. Khalid, and S. R. Shahid, "Urdu consonantal and vocalic sounds," *CRULP Annual Student Report*, 2002.

[12] X. Huang, A. Acero, J. Adcock, H.-W. Hon, J. Goldsmith, J. Liu, and M. Plumpe, "Whistler: A trainable text-to-speech system," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference On*, IEEE, vol. 4, 1996, pp. 2387–2390.

[13] T. Merritt and S. King, "Investigating the shortcomings of hmm synthesis," in *Eighth ISCA Workshop on Speech Synthesis*, 2013.

[14] A. W. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, IEEE, vol. 4, 2007, pp. IV–1229.

[15] N. Swetha and K. Anuradha, "Text-to-speech conversion," *Int J Adv Trends Comput Sci Eng*, vol. 2, no. 6, pp. 269–278, 2013.

[16] E. Eide, A. Aaron, R. Bakis, W. Hamza, M. Picheny, and J. Pitrelli, "A corpus-based approach to expressive speech synthesis," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[17] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, IEEE, vol. 3, 2000, pp. 1315–1318.

[18] N. Kaiki, "Linguistic properties in the control of segmental duration for speech synthesis," *Talking Machines: Theories, Models, and Designs*, pp. 255–263, 1992.

[19] M. D. Riley, "Tree-based modeling of segmental durations," *Talking machines*, pp. 265–273, 1992.

[20] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The hmm-based speech synthesis system (hts) version 2.0.," in *SSW*, Citeseer, 2007, pp. 294–299.

[21] A. A. Raj, T. Sarkar, S. C. Pammi, S. Yuvaraj, M. Bansal, K. Prahallad, and A. W. Black, "Text processing for text-to-speech systems in indian languages.," in *SSW*, 2007, pp. 188–193.

[22] H. Kabir, S. R. Shahid, A. M. Saleem, and S. Hussain, "Natural language processing for urdu tts system," in *Multi Topic Conference, 2002. Abstracts. INMIC 2002. International*, IEEE, 2002, pp. 58–58.

[23] S. Hussain, "Phonological processing for urdu text to speech system," *Yadava, Y, Bhattarai, G, Lohani, RR, Prasain, B and Parajuli, K (eds.) Contemporary issues in Nepalese linguistics*, 2005.

[24] M. Y. Liberman and K. W. Church, "Text analysis and word pronunciation in text-to-speech synthesis," *Advances in speech signal processing*, pp. 791–831, 1992.

[25] H. R. Basit and S. Hussain, *Text processing for urdu tts system*, Poster presentation in Conference on Language and Technology 2014 (CLT 14), Karachi, Pakistan, 2014.

[26] M. D. Riley, "Some applications of tree-based modelling to speech and language," in *Proceedings of the workshop on Speech and Natural Language*, Association for Computational Linguistics, 1989, pp. 339–352.

[27] H. S. Elovitz, R. W. Johnson, A. McHugh, and J. E. Shore, "Automatic translation of english text to phonetics by means of letter-to-sound rules," NAVAL RESEARCH LAB WASHINGTON DC, Tech. Rep., 1976.

[28] H. Ku, W. Francis, *et al.*, "Computational analysis of present-day american english," 1967.

[29] R. Carlson, B. Granstrom, and S. Hunnicutt, "A multi-language text-to-speech module," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.*, IEEE, vol. 7, 1982, pp. 1604–1607.

[30] D. H. Klatt, "Synthesis by rule of segmental durations in english sentences," *Frontiers of Speech Comm. Res.*, pp. 287–299, 1979.

[31] ——, "Software for a cascade/parallel formant synthesizer," *the Journal of the Acoustical Society of America*, vol. 67, no. 3, pp. 971–995, 1980.

[32] X. Huang, A. Acero, H. Hon, Y. Ju, J. Liu, S. Meredith, and M. Plumpe, "Recent improvements on microsoft's trainable text-to-speech system-whistler," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, IEEE, vol. 2, 1997, pp. 959–962.

[33] (). Microsoft research's speech technology group web page, [Online]. Available: https://www.microsoft.com/en-us/research/?from=http%3A%2F%2Fresearch.microsoft.com%2Fresearch%2Fsrd.

[34] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, IEEE, vol. 1, 1996, pp. 373–376.

[35] S. King, "A beginners' guide to statistical parametric speech synthesis," *The Centre for Speech Technology Research, University of Edinburgh, UK*, 2010.

[36] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for hmm-based speech synthesis," in *Fifth International Conference on Spoken Language Processing*, 1998.

[37] J. J. Odell, “The use of context in large vocabulary speech recognition,” PhD thesis, University of Cambridge, Mar. 1995.

[38] K. Tokuda, H. Zen, and A. W. Black, “An hmm-based speech synthesis system applied to english,” in *IEEE Speech Synthesis Workshop*, 2002, pp. 227–230.

[39] H. D. Harashima, “Review of ”voicetext”,” *Electronic Journal of Foreign Language Teaching*, vol. 3, no. 1, pp. 131–135, 2006.

[40] (2002). Hmm based synthesis system version 2.2, [Online]. Available: http://hts.sp.nitech.ac.jp/.

[41] R. A. Clark, K. Richmond, and S. King, “Multisyn: Open-domain unit selection for the festival speech synthesis system,” *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[42] M. B. Ganai and E. J. Arora, “Text-to-speech conversion,” 2016.

[43] R. E. Donovan and P. C. Woodland, “Improvements in an hmm-based speech synthesiser,” in *Fourth European Conference on Speech Communication and Technology*, 1995.

[44] A. S. House, C. E. Williams, M. H. Hecker, and K. D. Kryter, “Articulation-testing methods: Consonantal differentiation with a closed-response set,” *The Journal of the Acoustical Society of America*, vol. 37, no. 1, pp. 158–166, 1965.

[45] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, “Speech synthesis using hmms with dynamic features,” in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, IEEE, vol. 1, 1996, pp. 389–392.

[46] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, “Speaker adaptation for hmm-based speech synthesis system using mllr,” in *the third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998.

[47] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Speaker interpolation for hmm-based speech synthesis system,” *Acoustical Science and Technology*, vol. 21, no. 4, pp. 199–206, 2001.

[48] X. Huang, A. Acero, H.-W. Hon, and R. Reddy, *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR Upper Saddle River, 2001, vol. 1.

[49] O. Karaali, G. Corrigan, I. Gerson, and N. Massey, "Text-to-speech conversion with neural networks: A recurrent tdnn approach," *arXiv preprint cs/9811032*, 1998.

[50] T. Yoshimura, G. E. Henter, O. Watts, M. Wester, J. Yamagishi, and K. Tokuda, "A hierarchical predictor of synthetic speech naturalness using neural networks.," in *INTERSPEECH*, 2016, pp. 342–346.

[51] K. T. Alan W Black Simon King. (Jan. 2009). The blizzard challenge 2009, [Online]. Available: https://synsig.org/images/archive/9/94/20090121161941!Blizzard_2009_full.pdf.

[52] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," *Proc. SSW, Sunnyvale, USA*, 2016.

[53] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 7962–7966.

[54] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[55] P. K. Muthukumar and A. W. Black, "Recurrent neural network postfilters for statistical parametric speech synthesis," *arXiv preprint arXiv:1601.07215*, 2016.

[56] A. A. Shah, A. W. Ansari, and L. Das, "Bi-lingual text to speech synthesis system for urdu and sindhi," in *National Conf. on Emerging Technologies*, 2004, pp. 20 126–130.

[57] Z. Ahmed and J. P. Cabral, "Hmm-based speech synthesiser for the urdu language," in *Spoken Language Technologies for Under-Resourced Languages*, 2014.

[58] W. B. Kleijn and K. K. Paliwal, *Speech coding and synthesis*. Elsevier Science Inc., 1995.

[59] (). Center for research in urdu language processing, [Online]. Available: http://www.cle.org.pk/software/ling_resources.htm.

[60] S. Hussain, "Letter-to-sound conversion for urdu text-to-speech system," in *Proceedings of the workshop on computational approaches to Arabic script-based languages*, Association for Computational Linguistics, 2004, pp. 74–79.

[61] O. Nawaz and T. Habib, "Hidden markov model (hmm) based speech synthesis for urdu language," in *Conference on Language & Technology (CLT)*, 2014.

[62] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, IEEE, vol. 1, 1992, pp. 137–140.

[63] W. Anwar, X. Wang, L. Li, and X.-L. Wang, "A statistical based part of speech tagger for urdu language," in *Machine Learning and Cybernetics, 2007 International Conference on*, IEEE, vol. 6, 2007, pp. 3418–3424.

[64] S. Bird, E. Klein, and E. Loper, "Introduction to natural language processing," *University of Pennsylvania*, 2007.

[65] J. Carlberger and V. Kann, "Implementing an efficient part-of-speech tagger," *Software: Practice and Experience*, vol. 29, no. 9, pp. 815–832, 1999.

[66] N. Durrani and S. Hussain, "Urdu word segmentation," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2010, pp. 528–536.

[67] D. Parveen, R. Sanyal, and A. Ansari, "Clause boundary identification using classifier and clause markers in urdu language," *Polibits*, no. 43, pp. 61–65, 2011.

[68] J. Pierrehumbert, "Synthesizing intonation," *The Journal of the Acoustical Society of America*, vol. 70, no. 4, pp. 985–995, 1981.

[69] A. W. Black and P. Taylor, "Chatr: A generic speech synthesis system," in *Proceedings of the 15th conference on Computational linguistics-Volume 2*, Association for Computational Linguistics, 1994, pp. 983–986.

[70] A. W. Black, "Unit selection and emotional speech," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[71] Y. Qian, F. Soong, Y. Chen, and M. Chu, "An hmm-based mandarin chinese text-to-speech system," in *Chinese Spoken Language Processing*, Springer, 2006, pp. 223–232.

[72] O. Abdel-Hamid, S. M. Abdou, and M. Rashwan, "Improving arabic hmm based speech synthesis quality," in *Ninth International Conference on Spoken Language Processing*, 2006.

[73] S. Martincic-Ipsic and I. Ipsic, "Croatian hmm based speech synthesis," in *Information Technology Interfaces, 2006. 28th International Conference on*, IEEE, pp. 251–256.

[74] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[75] R. E. Donovan and P. C. Woodland, "A hidden markov-model-based trainable speech synthesizer," *Computer speech & language*, vol. 13, no. 3, pp. 223–241, 1999.

[76] J. F. Pitrelli, "Tobi prosodic analysis of a professional speaker of american english," in *Speech Prosody 2004, International Conference*, 2004.

[77] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[78] H. Zen, "Acoustic modeling in statistical parametric speech synthesis–from hmm to lstm-rnn," *Proc. MLSLP*, 2015.

[79] G. Cawley and P. Noakes, "Lsp speech synthesis using backpropagation networks," in *Artificial Neural Networks, 1993., Third International Conference on*, IET, 1993, pp. 291–294.

[80] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," in *Eighth ISCA Workshop on Speech Synthesis*, 2013.

[81] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural network (dnn) for parametric tts synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, IEEE, 2014, pp. 3829–3833.

[82] L.-H. Chen, T. Raitio, C. Valentini-Botinhao, Z.-H. Ling, and J. Yamagishi, "A deep generative architecture for postfiltering in statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 11, pp. 2003–2014, 2015.

[83] (). The festival speech synthesis system, [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival/.

[84] (). Centre for speech technology research, [Online]. Available: http://www.cstr.ed.ac.uk.

[85] (). Phonetically rich urdu speech corpus, [Online]. Available: http://www.cle.org.pk/software/ling_resources/phoneticallyrichurduspeechcorpus.htm.

[86] W. D. Voiers, "Diagnostic evaluation of speech intelligibility," *Speech intelligibility and speaker recognition*, 1977.

[87] (). Festvox, [Online]. Available: http://festvox.org.