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

External Examiner: Dr. NAME
Assistant Professor
Department of Computer Science
COMSATS Institute of Information Technology, 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)**

Permission is herewith granted to University of the Punjab to circulate and to have copied for non-commercial purposes, at its discretion, the above title, upon the request of individuals or institutions.

Signature of the Author

Dedicated to

Abstract

Text to speech synthesis system is system which takes raw text as input and converts it into speech signal. This is done by concatenation of small speech segments called phonetic strings of words or Statistical parametric speech synthesis which uses parameters to describe speech. In this technique, model is learned from speech data using Hidden Markov model (HMM) or Deep Neural Networks (DNN).

This paper describes development of Festival TTS system based Urdu text to speech system using Hidden Markov model (HMM). It describes Urdu text preprocessor system used to process numbers, dates and time text in input data and how Festvox voice package is generated for Urdu. In the end, evaluation of system is conducted using DRT, MRT and MOS tests to get performance of the system. .

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

Acknowledgements

Computational modeling is branch of computer science which deals with multiple disciplines. It assists other domains in understanding complex systems and phenomena by providing theory, tools and technology to model and simulate related systems and phenomena. In complex systems, behavior of an individual can have butterfly effect and can become root cause of an emergent phenomenon. Interaction of drivers with each other and surrounding environment forms the dynamics of traffic flow. Hence global effects of a traffic flow depend upon behavior of a single driver. In this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Speech Synthesis

Speech is most important medium of conveying opinions and expressing feeling and thoughts. Human convert their thought into speech by using words, phrases and sentences in order to communicate with each other [1]. Speech is produced when air is exhaled by the lungs and vibrations are produced by air, these vibrations got a proper waveform shape by glottal cords and vocal tract. Text to Speech synthesis is the process of conversion of raw text into speech signals. It works by concatenation of small segments of recorded speech called phonemes [2]. 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.

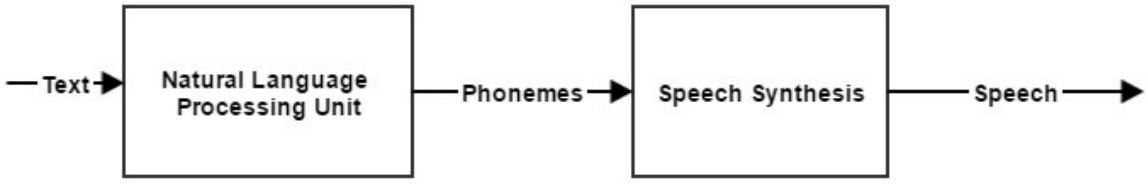

Figure 1.1: TTS Block Diagram

In NLP unit, text is first converted into string of letters and then word boundaries are marked by tokenizer. This is called normalization of text. Normalized data is then

converted into phonetic strings with the help of letter to sound rules after which Syllabifier marks syllable boundaries. Sound change rules are applied on the syllabified date. Language modeling techniques are also applied for finding context in which a specific word is used. As human has tendency to recognize basic rules for his native language, it is easy to judge context of a word in a sentence and what should be correct pronunciation of that word with respect to its context. For example, it can be guessed easily that "پل" in a sentence is used for پل (moment) or پل (bridge) for any native speaker of Urdu. Last stage of NLP is stress intonation marker which adds stress and intonation to the text. Speech Synthesis unit converts symbolic information received from NLP unit into audible speech with the help of different Digital Signal Processing Techniques. The quality of speech synthesis system is detected by naturalness and intelligibility of the speech.

Partially blinded or fully blinded people usually suffer while using computer technology when there is no assistant or computer is not enough interactive. Due to which text to speech systems are becoming necessity of modern life. These systems increase the degree to which blind people can interact with sighted people [3] and could boost up their hope to survive in this world gracefully [4]. 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 that convert text to speech are used [5].

In digital world there are some people who can read and understand different languages and some who can't understand languages except their own languages. 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 [6]. The Xitsonga is spoken in more than three African countries. TTS system of such languages will help lot of people of different literacy level [7]. Urdu is national language of Pakistan and it is spoken by more than 100 million people across the world [8]. A Text-to-Speech (TTS) system for Urdu will be very helping for visually impaired, handicapped and illiterate people.

For human, the task of speech synthesis is not difficult one as they have basic knowledge

of their language but for computer some other method has to be implemented for this task. When we talk about TTS systems speech types and procedure for synthesis, strategies or modules used to develop systems etc. are important to consider. Different types of speech exist such as isolated word (process single word at a time), connected words (isolated words but separated with least gap), continuous speech (permit client to talk while computer is processing content) and spontaneous speech (deals with variety of words that are used rarely) as well as two types of speaker model were presented independent and dependent of clients or speaker specifications. Vocabulary is also characterized according to size such as small vocabulary, medium vocabulary, large vocabulary, very large vocabulary and out-of-vocabulary. Below are the major speech generation techniques.

1.2 Types of Speech Synthesis

For the process of speech synthesis, three types of processes are used.

1.2.1 Formant Synthesis

In Formant Synthesis, simple model of speech production and set of rules are used to generate speech. But it is very difficult to accurately describe the process of speech generation in set of rules

1.2.2 Concatenative Synthesis

Concatenative Synthesis small units are selected from carrier sentences which then join to form speech of complete sentence. These small units are called phonemes. These are the units which collectively describe correct pronunciation of a word. This process is easy as compared to previous one as number of such phonemes is limited for any language. For English, there are 44 such phonemes. Similarly in Urdu, there are 44 consonants, 8 long vowels, 7 long nasal vowels, 3 short vowels and many diphthongs [9]. This reduce distortion but it can decrease the naturalness. That's why the derived synthetic speech may not resemble the donor speaker in training database [10].

1.2.3 Statistical Parametric Speech Synthesis

Statistical parametric speech synthesis is another approach which uses parameters to describe speech. In this technique, model is learned from speech data. This technique works better than concatenative technique [11].

1.3 Quality

The intelligibility and naturalness is the measure of quality of the synthesized speech [12]. There is 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 system, 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 understandings. 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 [4]. Multiple techniques like linear regression and neural networks were applied to get the results with improvements. Concatenation techniques are applied to get fully expressive and stylish messages for end users. By using concatenation technique users can customize and add styles and expression through provided Speech Synthesis Markup Language (SSML) [4]. Timing of events in speech is also important as timing of events in speech signals are affected by some contextual factors like phone identity factors. These factors make it difficult to control timing of events [13]. There are some approaches which have been proposed to control timing of events like linear regression [14] and tree regression [15]. A new technique is proposed in [13] in which timing of events is controlled by multi-dimensional Gaussian distribution based Hidden Markov model.

1.4 Architecture

Text to speech is a way of communication and transferring information using words and styles of speaking [4]. It has two processes which are text processing and speech generation. In text processing given input text is processed so that to get appropriate chain of phonemic units. Speech generator takes these units as inputs and convert them into synthetic speech by selection of a unit from large corpus TTS system for small database is easier to implement but not in good quality [16] [17] [18]. Different researchers and developers use different strategies to develop TTS system such as in [19], author interpret text to speech system as, it converts raw text into intelligible speech signals by following two sub processes called High-level synthesis and Low-level synthesis. High-level synthesis converts text into phonetic strings and Low-level synthesis converts these strings into speech signals [19]. In [20], TTS system is divided in three modules.

1. Natural language Processing
2. Text Parameterization
3. Speech Synthesis

Natural Language processing 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 1.2.

In [21], 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 inputted text is segmented into sentences and later on to words. These words are then categorized according to their syntactic and contextual meaning.

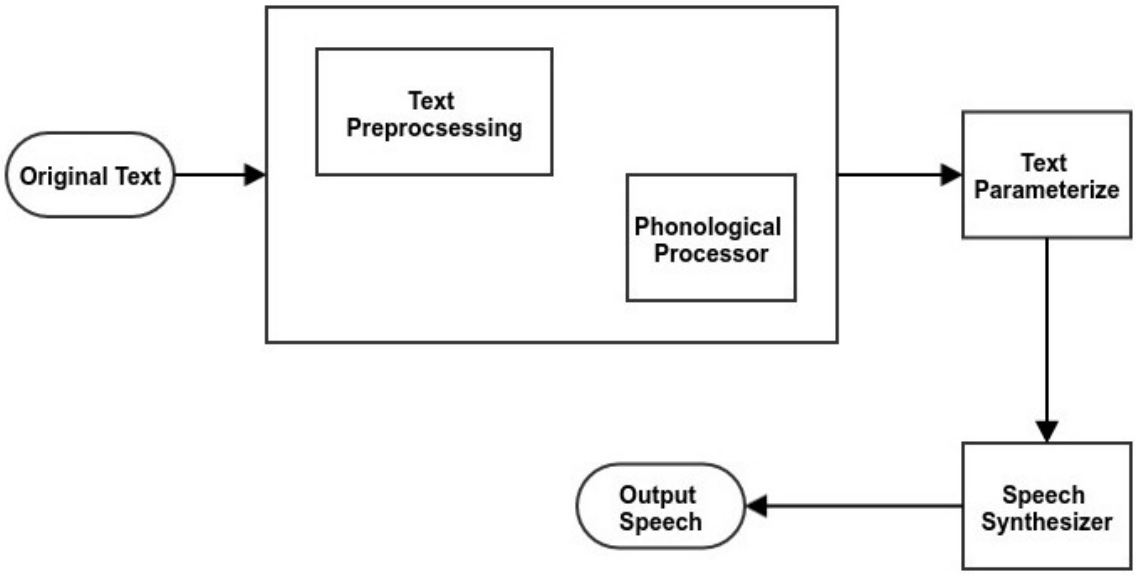

Figure 1.2: Architecture of TTS

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, 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 [22], text-to-Speech system is divided in two parts. One is called Natural Language Processing unit and other is called Speech Synthesis unit. Natural Language Processing 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

The TTS conversion is not a new field and people have been working on this field before electronic signal processing techniques. In beginning, people tried to build machines which were used to create human sound. After the development of computers, better systems were built using different techniques. In [12], basic speech synthesizing technique which works by concatenation of small recorded speech segments called phonemes to form complete speech are discussed. Each word is first divided into syllables and then pronunciation for each syllable is concatenated in order to get pronunciation for whole word. This concatenated word has some delay between pronunciations of each syllable which is removed and as a result, final pronunciation of that word is obtained. Problems like Text Preprocessing, Pronunciation and Prosody makes it difficult. In Text Preprocessing, digits and abbreviations are converted to full words. Other problem is guessing correct pronunciation of a word. For example, word "lives" has different pronunciation in "He lives in Lahore" and "He saved two lives". To create naturalness in sound, stress and intonation are applied to the input text which is also a very complex task.

A rule based technique is designed in [23]. The dataset is developed by extracting 50,000 words from standard Corpus, Corpus of Present-Day Edited American English i.e. Brown corpus [24]. The system gave accuracy of about 93%. A more improved system was proposed in [25] and [5].

Dictionary and rule based approach is used in [21] where 1000,000 words were used for training model. A TTS system with prosody and concatenative speech parameters that were extracted through use of probabilistic learning methods in [10]. A formant and Concate-

native synthesis is developed in [26] where small segments of phonemes were concatenated to form whole speech. A training database was used contains about 6,000 phonetically balanced sentences recorded in natural style. In [27], linear regression and unit selection based speech synthesis is designed using ATR Japanese database.

Statistical parametric speech synthesis is another approach which uses parameters to describe speech. In this technique, model is learned from speech data. This technique works better than concatenative technique. In [11], author discussed shortcomings of concatenative synthesis and why Hidden Markov model based technique is better than concatenative.

A Hidden Markov model based Statistical parametric speech synthesis system was developed in [28] where 450 sentences from ATR Japanese Database were used. A similar system is designed in [13] using Hidden Markov Model and evaluated it by taking input from Japanese database and by generating feature vector of those sentences and compared with original speech.

A Hidden Markov model and unit selection based system is proposed in [29] in which 524 sentences from CMU communicator database were taken. The sentences are used to develop system and two algorithms are compared. The result showed unit give high quality results but it uses large data, voice remain fixed whereas HTS gave smooth, stable, various voices but it gives buzzy speech. In [30], Hidden Markov model and rule based approach is 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 [4] in which manually produced dataset which is not publicly available was used. To evaluate the synthesized speech, the output is given for testing to 32 native English speakers. For bad news, good news and for yes/no the accuracy is 70.2%, 80.3% and 84% respectively

Tones and Break Indices ToBI are discussed on American English in [31]. Majorly bi-gram and tri-gram were used to predict the occurrences of particular word or letter. Analysis was performed by multiple techniques. The corpus was divided into following five yes-no questions, either-or questions, other questions (hereafter, “wh-questions”), exclamations, and other declarative sentences categories for the analysis purpose. Analysis by word frequency count produced not good results and the reason was that the data was of variant types. The results show that professional speakers produces better and informative prosodic events as compared to ordinary speakers.

A TTS system for Azerbhaijani language using concatenative synthesis in [32] where small recordings were concatenated to make speech waveform.

Hidden Markov model based approach is used in [7] to construct speech synthesizer for Xitsonga which is an African language. The system received acceptability of 92.3

TTS system for Fon language is designed in [6] using Multisyn algorithm 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. Back-off rules are applied when input text contains some characters that are not in us know characters. DSP module than choose required unit from database of units are concatenate them to form complete speech signals.

In [33], hybrid text to speech converter is developed by concatenating benefits of HMM based TTS system and waveform based TTS system. System is developed using Matlab, a library of phoneme and their sound is created and waveform of audio and audio itself is generated using these libraries.

A more advance technique in Neural Network based technique as it works better than Hidden Markov model based technique. Time domain neural networks with database containing sounds of words called phonemes is used in [34]. The basic flow of the system involves speech recording, speech labelling, voice coder and input processing using Time Delay Neural Network. The figure 2.1 shows the block diagram of system.

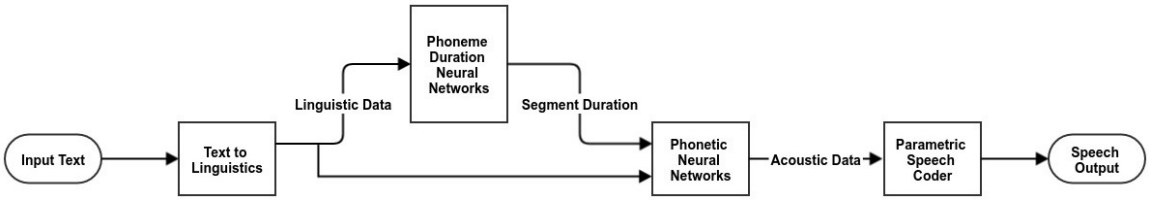

Figure 2.1: Time Domain Neural Network based TTS System

Deep Neural Networks is applied in place of Hidden Markov model in [35] as HMM based system cannot model complicated context dependencies. Deep Neural Networks (DNN) can cover limitations in HMM based system and can also outperform HMM based system.

Recurrent Neural Networks (RNN) is applied in [36] 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 female native speaker. Objective and subjective evaluation measures are used to find distortion between natural and synthesized speech and quality respectively. The preference results that hybrid system is better as hybrid gave 44%, 59% and 55% accuracy whereas neural network, HMM and DNN gave 29%, 22% and 20% accuracy respectively.

Recurrent Neural Network (RNN) is used in [37] for filter for synthesis. Apart from that a novel approach was also used for training of Classification and Regression Tree jointly instead of training all these independently.

Neural Networks is used in [38] with dataset consisting of 328 hours 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.

For Urdu Language, [9] and [22] described Natural Language Processing (NLP) unit for Urdu TTS. NLP unit is divided in two parts called as preprocessing and phonological processing unit. Pre-processing unit converts number, date and time into their respective literal strings. For example, 100 and 5-11-2002 will be converted into سو and پانچ نومبر دو ہزار دو respectively. Special symbols like $ and are also handled in pre-processing unit. Last stage of pre-processing unit is grapheme into phoneme convertor. Phonological processing unit contains syllable marker which marks syllable boundaries and stress and intonation markers mark stress and intonation.

In [9] consonantal and vocalic sounds for Urdu Language is discussed . Bi-Lingual Text to Speech Synthesis System for Urdu and Sindhi is designed in [39] using bilingual hybrid knowledge based approach by using concatenated synthesis method. (Hussain, S. 2005) discussed Phonological Processing unit for Urdu language in detail. This module applies letter to sound rules, syllabification to the normalized text. This is followed by stress and intonation marker. Statistical based part of speech tagger for Urdu language is discussed in [40]. The model is evaluated by comparison of unigram, bigram and backoff experiments with small and large tag sets. t-test, POS accuracy are used to measure performance.

Problems in Urdu segmentation are discussed for Urdu in [41]. Clause boundary identification is discussed in [42] using classifier and clause markers in Urdu language using conditional Random Field as a classifier. (Zeeshan. A, Joao P, 2014) and (Omer Nawaz, Dr. Tania Habib, 2014) designed a HMM based TTS system for Urdu language using HTS toolkit is designed in [43] and [44].

Chapter 3

Methodology

Text to Speech synthesis system designed in this paper is divided in two sub modules. One perform analysis and preprocessing of data and other transform processed data into sound signals. These modules are following

1. Text Processing Unit
2. Speech Synthesis System

3.1 Text Processing Unit

This unit is the unit which is responsible for processing of text before text is sent to speech synthesis system. This unit find 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 helps to understand context of a word but are not converted into sounds. We remove all

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.	۱۲ دسمبر

Table 3.1: Regular Expression for Semantic Tagger

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. It is because it makes it 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. ۱۲ دسمبر 2012
    c. ۱۲ 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.

Word	Converted Text
123	ایک سو تئیس
1231	ایک ہزار دو سو اکتیس
123.1234	ایک سو تئیس اعشاریہ ایک دو تین
12345	بارہ ہزار تین سو پینتالیس
1234567	بارہ لاکھ چونتیس ہزار پانچ سو ستاسٹھ
987654321	اٹھانوے کروڑ چھہتر لاکھ چون ہزار 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 number, dates and time from text with each one marked as date/time/number. Text generator part will take each word and generate 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.

The both integer and fractional part of floating point number are handled differently. The algorithm for integral number is described below.

```
def get_factors(number):
  factored_integer_list = [100000000000, 1000000000, 10000000, 100000,
↪  1000, 100]
  factored_number = []
  for factor in factored_integer_list:
      If factor > number and number != 0:
```

```
            factored_number.append(number/factor)
            factored_number.append(factor)
            number = number % factor
    If number != 0:
        factored_number.append(number)
    return factored_number
```

This algo will return list of factored number 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 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. These two parts are then join 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 date to text Converter 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.

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

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

The word tagged as date is first processed to get year, month and day of 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 (Nineteen hundred and eighty nine). This thing 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 word 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 converted. 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 resultant text will only contains 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. 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. This can be used to build voice for limited domain in specific language.

3.2.1.3 SPTK

SPTK stands for Speech Processing Toolkit. As name suggests, this tool is for processing speech signals in UNIX systems.

3.2.1.4 Festival

Festival is a speech synthesis system developed in Centre for Speech Technology Research (CSTR) which is a multi-platform framework for building speech synthesis system. This system is designed in such a way that it can be used for following purposes

1. Research purpose for improvement in speech synthesis system
2. For 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 is model based speech synthesis in which model is trained using training data. Training data consists of recorded speech and their corresponding labels. In this method, speech is elaborated with parameters which are defined by statistics. This is why it is called as statistical parametric speech synthesis.

The CLUSTERGEN statistical parametric speech synthesis is type of synthesis in which 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 [45] is used. This data consists of recordings of 708 phonetically rich sentence, 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 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 word used in this system is shown in table B.3.

3.2.2.5 Labeling Data

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, context dependent models are trained using Baum-Welch. This labeler works in 8 steps. Prompt files are extracted from utterance structure of Festival.

- From prompt files unique sequence of phones are extracted and stored in a list.
- List of wav files is collected for feature extraction by using prompt files.
- 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 string to be processed and features of that string.
- **Token**: Token mean each word in a sentence 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 mean 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 is 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 text to speech synthesis process.

Chapter 4

Experiments and Results

The purpose of a Text to Speech system is to build a system which is capable of generating voice as close to human voice as possible. 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 weill 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 are required. For our system, we only focussed 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)

This test is for Indicative and relative assessment of the understandability of single starting consonants. Test is conducted with words which are similar in sound but differ with each other in initial consonants [46]. User have 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 is to test demonstrative and relative assessment of the coherence of single last consonants.Tests is conducted using words which are similar in sound but differ with each other in last consonants [47]. User have 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 conducted to find out to which extent generated voice is close to human voice. The 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. The 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. The 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 Diagnostic Rhyme Test (DRT) and Modified Diagnostic Rhyme Test (M-DRT).

4.2.1 Methodology

An evaluation form is designed which have three sections.

- Diagnostic Rhyme Test (DRT)
- Modified Diagnostic Rhyme Test (M-DRT)
- Mean Opinion Score (MOS)

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	تئیس
24	چوبیس
25	پچیس
26	چھبیس
27	ستائیس
28	اٹھائیس
29	انتیس
30	تیس
31	اکتیس
32	بتیس
33	تینتیس
34	چونتیس
35	پینتیس
36	چھتیس
37	سینتیس
38	اٹھتیس
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	انھتر
70	ستر
71	اکھتر
72	بھتر
73	تھتر
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	ۓ	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	۔	Arabic Zero
१	۱	Arabic One
२	۲	Arabic Two
३	۳	Arabic Three
४	٤	Arabic Four
५	٥	Arabic Five
६	٦	Arabic Six
७	۷	Arabic Seven
८	۸	Arabic Eight
९	۹	Arabic Nine

Table B.3: Hindi to Urdu Character Mappings

Bibliography

[1] Benazir Mumtaz et al. "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] Miss Prachi Khilari and VP Bhope. "A Review On Speech To Text Conversion Methods". In: *International Journal of Advanced Research in Computer Engineering & Technology* 4.7 (2015).

[3] Dennis H Klatt. "Review of text-to-speech conversion for English". In: *The Journal of the Acoustical Society of America* 82.3 (1987), pp. 737–793.

[4] Ellen Eide et al. "A corpus-based approach to expressive speech synthesis". In: *Fifth ISCA Workshop on Speech Synthesis*. 2004.

[5] Dennis Klatt. "The Klattalk text-to-speech conversion system". In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.* Vol. 7. IEEE. 1982, pp. 1589–1592.

[6] Theophile K Dagba and Charbel Boco. "A text to speech system for Fon language using multisyn algorithm". In: *Procedia Computer Science* 35 (2014), pp. 447–455.

[7] Ntsako Baloyi. "A text-to-speech synthesis system for Xitsonga using hidden Markov models". PhD thesis. University of Limpopo (Turfloop Campus), 2012.

[8] *Top 30 Languages by Number of Native Speakers*. URL: http://www.vistawide.com/languages/top_30_languages.htm.

[9] ABDUL MANNAN Saleem et al. “Urdu consonantal and vocalic sounds”. In: *CRULP Annual Student Report* (2002).

[10] Xuedong Huang et al. “Whistler: A trainable text-to-speech system”. In: *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference On*. Vol. 4. IEEE. 1996, pp. 2387–2390.

[11] Thomas Merritt and Simon King. “Investigating the shortcomings of HMM synthesis”. In: *Eighth ISCA Workshop on Speech Synthesis*. 2013.

[12] N Swetha and K Anuradha. “Text-to-speech conversion”. In: *Int J Adv Trends Comput Sci Eng* 2.6 (2013), pp. 269–278.

[13] Keiichi Tokuda et al. “Speech parameter generation algorithms for HMM-based speech synthesis”. In: *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. Vol. 3. IEEE. 2000, pp. 1315–1318.

[14] Nobuyoshi Kaiki. “Linguistic properties in the control of segmental duration for speech synthesis”. In: *Talking Machines: Theories, Models, and Designs* (1992), pp. 255–263.

[15] Michael D Riley. “Tree-based modeling of segmental durations”. In: *Talking machines* (1992), pp. 265–273.

[16] Alan W Black, Heiga Zen, and Keiichi Tokuda. “Statistical parametric speech synthesis”. In: *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. Vol. 4. IEEE. 2007, pp. IV–1229.

[17] Heiga Zen et al. “The HMM-based speech synthesis system (HTS) version 2.0.” In: *SSW*. Citeseer. 2007, pp. 294–299.

[18] Anand Arokia Raj et al. “Text processing for text-to-speech systems in Indian languages.” In: *SSW*. 2007, pp. 188–193.

[19] Hasan Kabir et al. “Natural language processing for Urdu TTS system”. In: *Multi Topic Conference, 2002. Abstracts. INMIC 2002. International*. IEEE. 2002, pp. 58–58.

[20] Sarmad Hussain. "Phonological Processing for Urdu Text to Speech System". In: *Yadava, Y, Bhattarai, G, Lohani, RR, Prasain, B and Parajuli, K (eds.) Contemporary issues in Nepalese linguistics* (2005).

[21] Mark Y Liberman and Kenneth W Church. "Text analysis and word pronunciation in text-to-speech synthesis". In: *Advances in speech signal processing* (1992), pp. 791–831.

[22] 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.

[23] Honey S Elovitz et al. *Automatic translation of English text to phonetics by means of letter-to-sound rules*. Tech. rep. NAVAL RESEARCH LAB WASHINGTON DC, 1976.

[24] H Ku, WN Francis, et al. "Computational analysis of present-day American English". In: (1967).

[25] Rolf Carlson, B Granstrom, and Sheri Hunnicutt. "A multi-language text-to-speech module". In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.* Vol. 7. IEEE. 1982, pp. 1604–1607.

[26] Xuedong Huang et al. "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*. Vol. 2. IEEE. 1997, pp. 959–962.

[27] Andrew J Hunt and Alan 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*. Vol. 1. IEEE. 1996, pp. 373–376.

[28] Takayoshi Yoshimura et al. "Duration modeling for HMM-based speech synthesis". In: *Fifth International Conference on Spoken Language Processing*. 1998.

[29] Keiichi Tokuda, Heiga Zen, and Alan W Black. “An HMM-based speech synthesis system applied to English”. In: *IEEE Speech Synthesis Workshop*. 2002, pp. 227–230.

[30] Hideto D Harashima. “Review of ”VoiceText””. In: *Electronic Journal of Foreign Language Teaching* 3.1 (2006), pp. 131–135.

[31] John F Pitrelli. “ToBI prosodic analysis of a professional speaker of American English”. In: *Speech Prosody 2004, International Conference*. 2004.

[32] R Aida–Zade, C Ardil, and AM Sharifova. “The main principles of text-to-speech synthesis system”. In: *International Journal of Signal Processing* 6.1 (2010), pp. 13–19.

[33] Mohd Bilal Ganai and Er Jyoti Arora. “Text-to-Speech Conversion”. In: (2016).

[34] Orhan Karaali et al. “Text-to-speech conversion with neural networks: A recurrent TDNN approach”. In: *arXiv preprint cs/9811032* (1998).

[35] Heiga Ze, Andrew Senior, and Mike 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.

[36] Yuchen Fan et al. “TTS synthesis with bidirectional LSTM based recurrent neural networks”. In: *Fifteenth Annual Conference of the International Speech Communication Association*. 2014.

[37] Prasanna Kumar Muthukumar and Alan W Black. “Recurrent neural network postfilters for statistical parametric speech synthesis”. In: *arXiv preprint arXiv:1601.07215* (2016).

[38] Zhizheng Wu, Oliver Watts, and Simon King. “Merlin: An open source neural network speech synthesis system”. In: *Proc. SSW, Sunnyvale, USA* (2016).

[39] Azhar Ali Shah, Abdul Wahab Ansari, and Lachhman Das. “Bi-Lingual Text to Speech Synthesis System for Urdu and Sindhi”. In: *National Conf. on Emerging Technologies*. 2004, pp. 20126–130.

[40] Waqas Anwar et al. “A statistical based part of speech tagger for Urdu language”. In: *Machine Learning and Cybernetics, 2007 International Conference on*. Vol. 6. IEEE. 2007, pp. 3418–3424.

[41] Nadir Durrani and Sarmad 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.

[42] Daraksha Parveen, Ratna Sanyal, and Afreen Ansari. “Clause Boundary Identification using Classifier and Clause Markers in Urdu Language”. In: *Polibits* 43 (2011), pp. 61–65.

[43] Zeeshan Ahmed and João P Cabral. “HMM-Based Speech Synthesiser for the Urdu Language”. In: *Spoken Language Technologies for Under-Resourced Languages*. 2014.

[44] O Nawaz and T Habib. “Hidden Markov Model (HMM) based speech synthesis for Urdu language”. In: *Conference on Language & Technology (CLT)*. 2014.

[45] *Phonetically Rich Urdu Speech Corpus*. URL: http://www.cle.org.pk/software/ling_resources/phoneticallyrichurduspeechcorpus.htm.

[46] William D Voiers. “Diagnostic evaluation of speech intelligibility”. In: *Speech intelligibility and speaker recognition* (1977).

[47] Arthur S House et al. “Articulation-Testing Methods: Consonantal Differentiation with a Closed-Response Set”. In: *The Journal of the Acoustical Society of America* 37.1 (1965), pp. 158–166.

[48] Rolf Carlson and B Granstrom. “A text-to-speech system based entirely on rules”. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP’76*. Vol. 1. IEEE. 1976, pp. 686–688.