A TEXT-TO-SPEECH SYNTHESIS SYSTEM FOR XITSONGA USING HIDDEN MARKOV MODELS

by

Ntsako Baloyi

Mini-Dissertation
Submitted in partial fulfilment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

in the

FACULTY OF SCIENCE AND AGRICULTURE
(SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES)

at the

UNIVERSITY OF LIMPOPO

SUPERVISOR: MR M. J. D. MANAMELA
CO-SUPERVISOR: DR N. GASELA

June 2012

DECLARATION

I declare that the mini-dissertation hereby submitted to the University of Limpopo, for the degree of Master of Science in Computer Science has not previously been submitted by me for a degree at this or any other university; that it is my work in design and in execution, and that all material contained herein has been duly acknowledged.

_____ **11/06/2012**

Baloyi N. (Mr) **Date**

[redacted]

Acknowledgements

I would like to thank the Almighty God for the grace to embark on and finish this project. Telkom South Africa and the National Research Foundation (NRF) have rallied behind research by providing financial support for students at the University of Limpopo (Turfloop Campus)'s Centre of Excellence for Speech Technology and at many other institutions as well; and that is to be applauded. I am truly grateful for the personal financial support offered by Telkom for my studies. I also extend my sincerest salutations to my supervisors, Mr M.J.D. Manamela and Dr N. Gasela. Dr N.E. Nxumalo of the African Language Department at the University of Limpopo (Tufloop Campus) has made very insightful linguistic inputs to this research project and deserves my salutations. Last but not least, I would to thank God for my family and friends. You are all very dear to me and I love you all.

Abstract

This research study focuses on building a general-purpose working Xitsonga speech synthesis system that is as far as can be possible reasonably intelligible, natural sounding, and flexible. The system built has to be able to model some of the desirable speaker characteristics and speaking styles. This research project forms part of the broader national speech technology project that aims at developing spoken language systems for human-machine interaction using the eleven official languages of South Africa (SA). Speech synthesis is the reverse of automatic speech recognition (which receives speech as input and converts it to text) in that it receives text as input and produces synthesized speech as output. It is generally accepted that most people find listening to spoken utterances better that reading the equivalent of such utterances.

The Xitsonga speech synthesis system has been developed using a hidden Markov model (HMM) speech synthesis method. The HMM-based speech synthesis (HTS) system synthesizes speech that is intelligible, and natural sounding. This method can synthesize speech on a footprint of only a few megabytes of training speech data. The HTS toolkit is applied as a patch to the HTK toolkit which is a hidden Markov model toolkit primarily designed for use in speech recognition to build and manipulate hidden Markov models.

Table of Contents

List of Figures

List of Tables

Abbreviations and Acronyms

CoE – Centre of Excellence
CSIR – Council for Scientific and Industrial Research
CSMAPLR – Constrained Structural Maximum A Posteriori Linear Regression
DB – Database
dB – Decibels
EM – Expectation-Maximization
FSA-PG – Faculty of Science and Agriculture Post-Graduate Research Day
GUI – Graphical User Interface
HLT – Human Language Technology
HMM – Hidden Markov Model
HTK – Hidden Markov Model Toolkit
HSMM – Hidden Semi-Markov Model
HTS – Hidden Markov Model Speech Synthesis Toolkit
ICT – Information and Communications Technology
KHz – KiloHertz
MAP – Maximum A Posteriori
MB – Megabytes
MDL – Minimum Distance Length
MLSA – Mel Log Spectrum Approximation
MOS – Mean Opinion Score
ms – Microseconds
MSD – Multi-Space Probability Distribution
NRF – National Research Foundation
PANSALB – Pan South African Language Board
PAT – Parametric Artificial Talker
PDF – Probability Density Function
RIFF – Resource Interchange File Format
SA – South Africa
SATNAC – Southern Africa Telecommunication Networks and Applications Conference
SMS – Simple Message Service

SPS – Statistical Parametric Synthesis

STRAIGHT – Speech Transformation and Representation using Adaptive Interpolation weiGHTed spectrum

TTS – Text-to-Speech

UK – United Kingdom

UL – University of Limpopo

URL – Universal Resource Locator (URL)

VOCODER – Voice Coder

VODER – Voice Operating Demonstrator

WER – Word Error Rate

Chapter 1

1 Introduction

1.1 Background

A text-to-speech (TTS) synthesis system is a computational/computer-based system that synthetically audibly reads out input text given in a particular natural language of interest. Such a system is fed with text as its input and the corresponding waveform rendition becomes its output. The input text can be in various forms, such as, keyboard input, text/word documents, emails, web pages, blogs, simple message service (SMS), chat room conversations' scripts, etc. The TTS synthesis systems are sometimes also referred to as speech synthesis systems. Although there has been significant improvements on the naturalness of speech synthesis systems over the years, there is still a huge gap between natural and synthesized speech [1].

The development of a TTS synthesis system for one of the resource-scarce, indigenous languages of South Africa, is aimed at improving human-computer interface issues of addressing the digital divide that exists between the under-privileged/illiterate citizens and the corporate, privileged or literate few. While there are a wide variety of assistive technologies for the physically challenged that use English, the same is not true for some of the under-resourced languages like Xitsonga. We live in an information age; and thus, there is a serious need to make available all kinds of information to even the most disadvantaged and illiterate of our people. The development of a general purpose TTS system using one of the indigenous languages of South Africa is also a step towards making both information and technology accessible and easy to use by all individuals at different literacy levels.

Thus far, there has not been much research on speech technology systems for Xitsonga language. Furthermore, there is currently no TTS synthesis system for Xitsonga language at the University of Limpopo speech technology research

programme – the Telkom Centre of Excellence (CoE) for Speech Technology. Xitsonga is a resource-scarce language and is lagging behind most of South Africa's languages when it comes to developments in the speech technology arena. This research project is an effort to make resources available for Xitsonga language speakers in addition to creating and expanding a pool of speech technology resources for South African languages. Speech synthesis systems can also be used by visually challenged people to access stored information or to create documents in a much easier way. There are many other uses of speech synthesis systems such as email readers, teaching assistants, eye-free computer interaction, etc. [2]

A speech synthesis method based on hidden Markov models was selected for use in this research project. The HMM-based speech synthesis can also be referred to as statistical speech synthesis (SPS). The HTS toolkit has been used for experimentation purposes. The choice of this speech synthesis method based on HMMs, over other speech synthesis methods was inspired by its ability to synthesize intelligible and natural sounding speech without requiring a huge training corpus [3, 4]. This method achieves this task by statistically modelling speech parameters using HMMs. Furthermore, the runtime synthesis engine of HTS – the toolkit used for HMM-based speech synthesis – is considerably small, spanning only a few megabytes (MBs), when excluding the text analysis component. Low memory requirements, flexibility, and ease of adaptability to speaker's voice characteristics and speaking styles using the HTS toolkit are some of the factors that favoured the choice of this method of speech synthesis over other methods. It is, therefore, easy to implement a system built using HTS on multiple platforms, including those associated with handheld devices.

1.2 Objectives

This research project was developed with the following objectives in mind:

- A general purpose Xitsonga TTS synthesis system that will be easy to use and can be easily integrated into various voice-enabled software application systems. This will encourage use by a broader audience and can assist

computer/techno-phobic people to relate and/or interact well with technology in a language that they understand.

- Developing a Xitsonga baseline TTS synthesis system that will be flexible for making future improvements to the system and be a good framework for developing TTS systems for other under-resourced languages.
- Making the use of indigenous languages interesting and appealing even to the younger generation through technology, thereby significantly contributing to the preservation of the heritage of these languages.

1.3 Significance/Relevance of Study

This research project is in harmony with the mission of the national Department of Arts and Culture to promote the use of indigenous languages even within the information and communications technology (ICT) arena [5]. A text-to-speech system makes it possible for people who cannot read, to be able to listen to near natural sounding utterances of written text in a language that they understand. It is the responsibility of the national, provincial and local government to make government information available to all citizens and this task can be made easy with the advent of efficiently working and user-friendly speech synthesis tools for illiterate people. Such a mode of information dissemination is also essential for the private sector or corporate world. People wishing to learn a new language can use a TTS system to learn a language by listening to how text in the language of their interest is pronounced. The TTS synthesis systems can also be integrated to work with systems that recognise text from scanned documents and those that recognise a person's handwriting digitally.

The visually challenged can benefit enormously from using speech synthesis systems as hearing tools (reading out text conversations) for communication and/or as learning assistants. However, the TTS synthesis systems do not only benefit visually-challenged people; professionals, too, can use a TTS to quickly browse through their electronic mails. This will not only save people time and make life easier, but it can also, to some extent, reduce the need to constantly focus on computer monitors. The TTS synthesis systems can contribute to policy

developments with regard to information access issues. The LWAZI project run by the Council for Scientific and Industrial Research (CSIR) addresses community needs using spoken language systems [12]. New computer-based products can also be developed such that they integrate TTS synthesis systems and/or other spoken language systems. The Microsoft Windows operating systems, for instance, includes an English TTS synthesis system among other spoken language processing systems.

1.4 Text-to-Speech Synthesis System Components (TTS Structure)

The basic architecture of a TTS system includes text analysis, phonetic analysis, prosodic analysis, and synthesis components [3, 7]. The text analysis component normalizes the input text (raw or tagged) into the form appropriate for conversion into speech. The processed text is then converted into its corresponding phonetic sequence by the phonetic analysis component. During prosodic analysis, pitch and duration information are attached to the prosodic sequence. Lastly, during synthesis, a rendition of the corresponding voice output takes place [7]. Figure 1–1 is a pictorial representation of these phases.

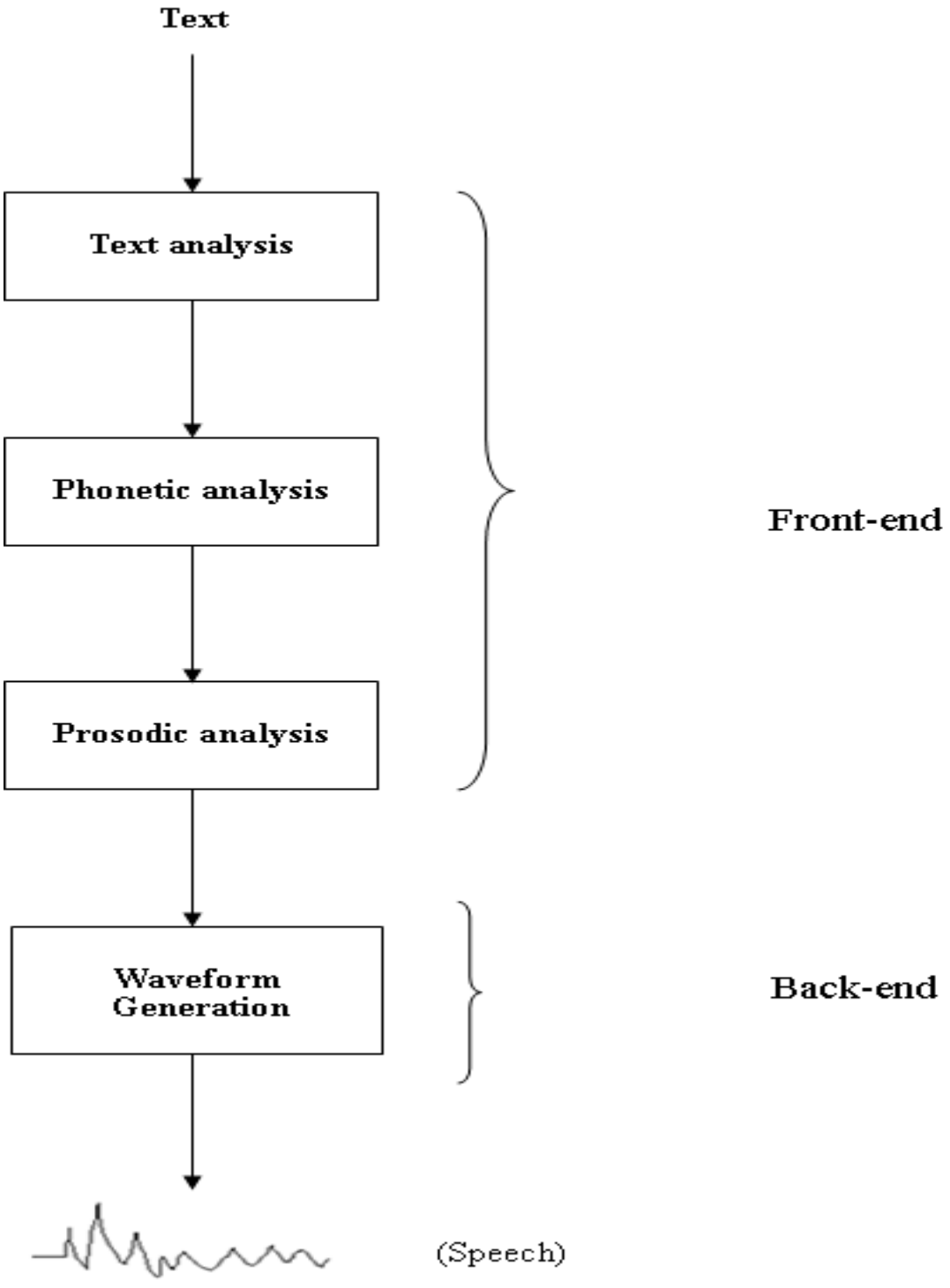

Figure 1–1. A TTS synthesis system block diagram

1.5 The Mini-Dissertation Outline

In chapter 1, a brief description of what a TTS synthesis system is and what it does are discussed. A brief overview of the methodology selected for this study is given. The significance of this research is discussed and the objectives outlined. The current state of TTS synthesis systems' research for Xitsonga at the University of Limpopo is also outlined. This chapter concludes with the discussion that focuses on the different applications of TTS synthesis systems and the components that make up a TTS synthesis system.

Chapter 2 elaborates on and reviews the literature of TTS synthesis systems. It takes both a historical and current view on the progress made in the TTS synthesis systems' developments. Different historical and recent systems are also looked into in some detail. The different methods used in the development of the TTS synthesis systems are explained. Several concepts critical to the understanding of the TTS synthesis system terminology are also described.

Chapter 3 looks into the history and orthographic structure that underpins Xitsonga as a natural language. It is in this chapter that details on the initial reduction of Xitsonga speech into writing are briefly overviewed. A top view of the phonetics and phonology that underlies the origin of the orthography of the language is briefly undertaken.

Chapter 4 takes a detailed view into the research methodology chosen for the development of a Xitsonga speech synthesis system. A hidden Markov model approach for speech synthesis, which is the method chosen in this research project is discussed. The reasons underlying the choice of this method are advanced together with the different phases involved in the development of a TTS system using this method. Two mandatory phases and one optional phase which are training, synthesis and adaptation respectively are also elaborated on.

In chapter 5, details that pertain to the design of the TTS synthesis system are discussed. This chapter is *the workhorse of the entire research project* as it discusses details ranging from data collection to implementation. Sentence selection, speech recording, file/data preparation, and software installation are some of the issues that are detailed in this chapter.

The evaluation of the system is discussed in chapter 6. This chapter starts by discussing the methods used for evaluating the system. It then looks into details of the evaluation, such as the number of respondents, the background of evaluators and the number of sentences used for evaluation. The results of the evaluation are then used to provide an analysis of the results acquired from the respondents.

Chapter 7 uses the results and the analysis given in chapter 6 to outline summative remarks or the conclusion of the entire research project. The recommended future improvements to the system comprise the last portion of the chapter.

1.6 Chapter Summary

This chapter started with a definition of TTS systems. It then looked into the different uses of TTS systems. The methodology adopted for this research was briefly discussed together with some of the factors that informed its choice. The set objectives for this research project and the general TTS structure have also been detailed in this chapter.

Chapter 2

2 Literature Review

This chapter discusses the reviewed literature relating to TTS synthesis systems. It starts by discussing the origins of TTS synthesis systems and the subsequent early developments in trying to synthesize speech. It also looks into some recent developments of TTS synthesis systems both locally and internationally. An overview of some key concepts in speech synthesis is also presented. More advanced TTS synthesis systems are also briefly discussed. The different methods that are available for TTS development are overviewed in this chapter as well.

2.1 History of TTS Systems

This section focuses on the history of TTS synthesis systems, dating from the early days of attempting to develop speaking machines to the recent state-of-the-art TTS developments. This section is significant in that it gives an appreciation of the improvements that TTS systems have undergone over the years starting with the early conception of the ideas to develop such systems. It also invokes the need for the much needed improvements in the quality of synthesized speech.

2.1.1 Early systems

Christian G. Kratzenstein was the first person to attempt synthesizing human speech. At least that is one part of the origin of speech synthesis history that several authors seem to agree on. The other part has to do with the year in which the work was done. One view holds that this work was done in 1773 [8, 1] while the other states the year 1779 [9, 10]. This Russian professor in Physiology explained and demonstrated the physiological differences of *five vowels* (a, e, i, o, u) by making the apparatus for their artificial production [9]. His apparatus included acoustic

resonators to mimic the human vocal tract system. Vibrating reeds were used to control air flow similar to the use of vocal cords [9].

In 1791, the Hungarian, Wolfgang von Klempelen tried to construct a speaking machine by simulating the human speech production system. His system went as far as being able to produce short sentences. He is recognized as the first experimental phonetician. Von Klempelen documented the details of his invention in order to make it easy for others to redevelop or maybe better the system [8]. He also developed a speaking chess-playing machine, whose mechanical details were not documented and made public as had been done with his speaking machine [9, 10].

The machine was constructed in such a way that the lungs were resembled by a pressure chamber (or bellows), the vocal cords were simulated by a shaking reed, and the vocal tract action imitated by a leather tube [9, 10]. As reported in von Kempelen's work, *vowel* sounds could be produced by varying the resonance properties of the mouth (rubber). For the production of *consonants*, the system was equipped with four separate constricted passages (or holes), which could be actuated with the controller's fingers. Nasals could be produced by leaving the nostrils open and that was the only time they could be kept open when producing speech sounds. Levers attached to the machine were used to produce fricatives [S, ʃ, Z, ʒ] and rattling [R] sounds. Although the shape of the mouth could not be changed, the length of the reed could be varied, except during speech production. Contrary to the first version of von Kempelen's machine, the last version allowed for the length of the reed to be varied for pitch control during speech production, by means of a handle [8].

In 1835, Charles Wheatstone reconstructed von Kempelen's speaking machine by making a few changes to it. Wheatstones's version of the speaking machine, however, included a flexible oral cavity (or mouth) and an active voicing control, thereby, simulating a more natural articulation or articulatory system. While the final version of Kempelen's machine was capable of pitch control by varying the length of the reed during speech production, Wheatstone's machine did not have this capability [8]. All vowels are voiced [1]. Voiced sounds were produced by exerting pressure on the bellows and forcing the air through the shaking reed [10]. Unvoiced

sounds, which are consonants were produced when the reed was off and the corresponding passage open [9, 10]. Vowels were produced with the reed on and all passages closed [9, 10]. Vowel resonances were effected by squeezing the leather tube [9]. Figure 2–1 gives a pictorial representation of Wheatstone's machine.

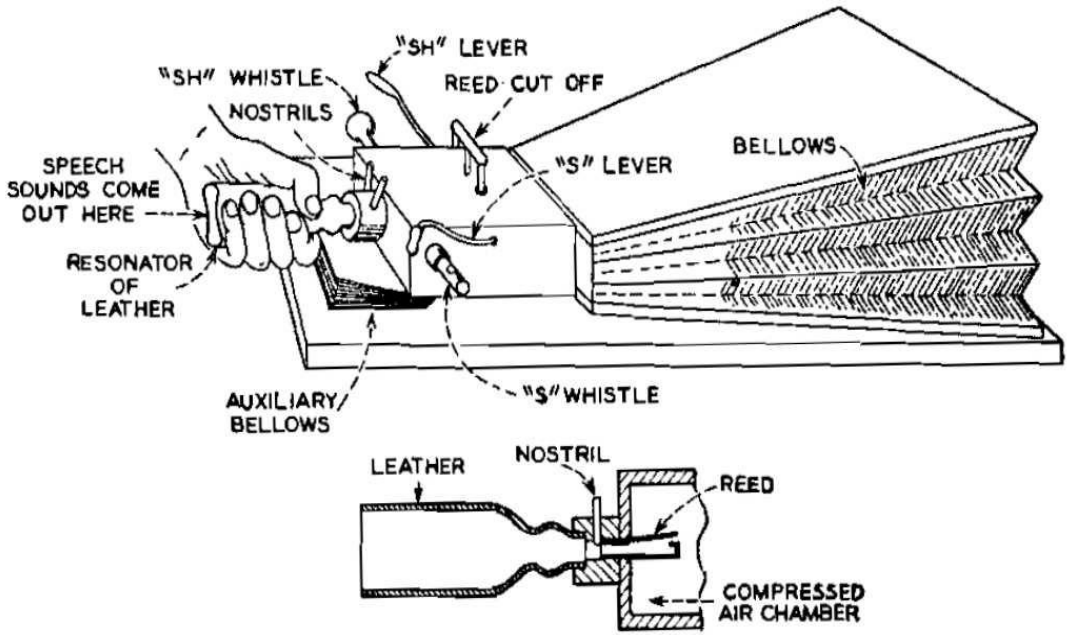

Figure 2–1. Charles Wheatstone's reconstruction of the von Kempelen's speaking machine [10]

In the late 1800, Alexandra Graham Bell inspired by Kempelen's machine and with support from his father and brother, developed his own speaking machine [10]. He put together equipment that tried to simulate the human vocal organs such as lips, cheeks, tongue, larynx, windpipe, etc. [10]. His speaking machine was also controlled using a keyboard and could synthesize *vowels*, nasals and simple *sentences* [10]. Bell went as far as attempting to teach his pet (dog) to speak, by manipulating the dog's vocal tract by hand as it growled continuously [9, 10]. Although the dog could do nothing more than to growl by itself, Bell managed to manipulate it to say *"How are you Grandmamma"* [10].

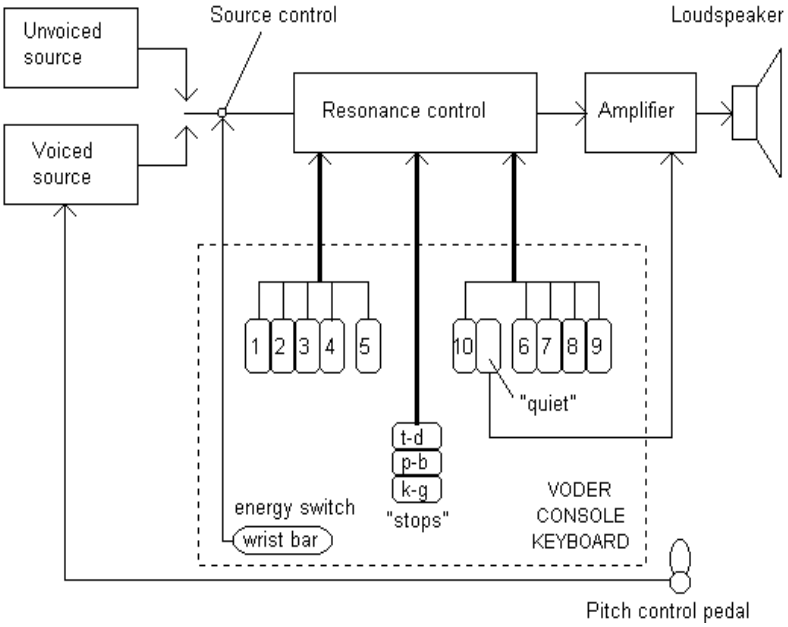

Figure 2–2. The VODER [9]

In 1939, Homer Dudley developed what could be said to be the *first speech synthesizer*, called, VODER (Voice Operating Demonstrator) based on the VOCODER (Voice Coder). The basic idea behind the VOCODER was to analyze speech into acoustic parameters which would then be used to try and approximate the construction of the original signal. The VODER had a resonance control box with 10 band filters connected in parallel and spanning the speech frequency range. It also had three more keys for the stop-consonants sounds as can be seen in Figure 2–2. The device was operated like a keyboard and required a skilful/trained person to operate the keyboard for playing a sentence [9]. As can be seen in Figure 2–3, the device was equipped with a wrist bar for selecting the excitation (voicing and noise) source and a foot pedal for controlling the pitch (fundamental frequency) [9, 10]. Training the device required a considerable amount of time ranging from a year to years [10]. Although VODER could not produce intelligible and good quality speech, it demonstrated a capability to produce artificial speech [9].

Figure 2–3. World Fair VODER demonstration 1939 [8]

In 1953, Walter Lawrence developed what is said to be the *first formant synthesizer* – the Parametric Artificial Talker (PAT) [9]. PAT had three parallel formant resonators and painted patterns which could be converted into three formant frequencies, voicing and noise (excitation) amplitude, and fundamental frequency. The PAT could produce natural sounding consonants [8].

The year 1970 saw the introduction of computers in the development of speech synthesizers [8]. This breakthrough meant that speech synthesizers could now be trained much faster, although the speech quality still suffered in terms of naturalness.

2.1.2 Recent TTS synthesis systems

According to Semi Lemmetty [9], by the year 1977, the first Finnish speech synthesis system called SYNTEC2 was developed. SYNTEC2, boasts being the first portable TTS synthesis system in the whole world and was microprocessor-based. Several TTS synthesis systems for Finnish have since been developed. In 2008, Tuomo Raitio developed a TTS synthesis system for Finnish using the glottal inverse filtering method embedded in the *Hidden Markov Model (HMM)* framework [7]. The glottal inverse filtering method was used as a way to improve the naturalness of synthetic speech for speakers with various speaking styles, characteristics and emotions.

Producing natural sounding speech is still a challenge although there has been great progress over the years [7]. Speech technology research focusing on investigating methods that can produce more natural sounding speech is still very active. There are also similar research endeavours underway at the Human Language Technology (HLT) division of Meraka Institute of the CSIR to improve the naturalness of synthetic speech by focusing on language intonation.

According to [11], Aby Louw together with his colleagues built a general-purpose IsiZulu TTS synthesis system based on Festival's "Multisyn" unit-selection approach. Diphones – which are comprised of the latter half of one phone with the first half of the other – were chosen as speech units for the training of the IsiZulu speech synthesizer. The system was trained using 210 recorded sentences which included frequently occurring *English loan words, numbers and dates*. The IsiZulu TTS Synthesizer was evaluated by individuals ranging from first language speakers to speakers of other languages who spoke and/or understood Zulu and people with low literacy to high literacy levels. On average, the system was found to synthesize speech that is both understandable and natural sounding.

A diphone-based concatenative synthesis system and a cluster unit selection synthesis system were developed for Northern Sotho [12] at the University of Limpopo (Turfloop Campus). Nonsense words were used for the extraction of diphones through a diphone index file in the diphone-based system. As for the unit selection system, 400 sentences were recorded, with only 43 sentences manually correctly labeled. Both systems were trained using speech files at 16 KHz mono. Festival and Festvox were used for the development of the two systems. To evaluate the systems, eleven different sentences in Northern Sotho were used. Fourteen individuals ranging between the ages of 21 and 30 were used to do the listening tests and evaluate the systems. Five of the fourteen respondents were female, with the other nine being males. Respondents were not only Northern Sotho speakers but different dialects of Northern Sotho were represented in the evaluation process. The understandability of both systems was found to be almost the same, with the naturalness of the cluster unit selection system perceived as being better than that of the diphone system. The overall impression of both systems was perceived to be

almost the same, with the diphone system slightly better than the unit selection system.

In 2008 Pieter Scholtz et al. [13] developed an SPS system for entry into the Blizzard challenge. It was a first time entry for the South African university (Stellenbosch University) to participate in this world class challenge called the Blizzard challenge. Two United Kingdom (UK) English voices and one Mandarin voice were used as entries to the challenge. The system scored well on intelligibility, but suffered on the categories of naturalness and similarity to original speaker. Poor intonation modelling of the Mandarin voice resulted in poor voice quality compared to the two UK English voices.

2.1.3 Multilingual and hybrid TTS systems

Multilingual TTS synthesis systems have also appeared; such systems can be trained and used for various languages. A multilingual speech synthesis system is one that is trained using data from different languages in order to be able to synthesize speech from any given text in any of those languages for which the system has been trained. **Speect** is an example of a multilingual TTS synthesis system developed at the Meraka Institute's HLT division for the eleven South African official languages [14]. Speect can also be used to create multilingual TTS synthesis systems for other languages similar to the way the Festival system has been and is used to develop TTS systems for individual languages. Festival is also a multilingual speech synthesis system [16].

In 2006 Lehlohonolo Mohasi developed a hybrid TTS synthesis system for Sesotho [1]. A hybrid TTS synthesis system is a system developed by combining two different TTS synthesis synthesis methods. Mohasi's system was based on both the limited domain and open vocabulary methods, thereby epitomizing on advantages of both methods to produce an even better system. Limited domain systems are known for their high levels of naturalness but can only synthesize words within the domain of interest, i.e., only words in the database (DB). On the other hand, open vocabulary systems suffer from naturalness but are more flexible in that they can synthesize

even words not in the database. The result of the two systems is a system that is both flexible (synthesizing even words not in the DB) and natural sounding (human sounding), but the synthesized speech was not perceived as being fluent. To address the problem of fluent speech, Mohasi implemented intonation modelling techniques which produced better results as per subjective evaluations.

2.2 Text-to-Speech Synthesis System Overview

The purpose of this section is to outline concepts that are key to the understanding of TTS synthesis systems. This section lays a good reference guide and foundation for users who are new to the development of TTS synthesis systems and other interested readers.

2.2.1 Front-end and back-end

A speech synthesis system is an electronic system that receives typed or stored text as input and produces or synthesizes the corresponding speech waveform as output. A TTS synthesis system trained for a particular language can be used to synthesize arbitrary text in that language. It is common practice to view TTS synthesis systems as consisting of a front-end and a back-end. The former is responsible for text analysis, whereby tokenization converts ambiguous symbols like dates to their equivalent word format, and then these words are also assigned their corresponding phonetic transcriptions by a process called grapheme-to-phoneme conversion [7]. The latter is responsible for converting the output (phonetic transcriptions and prosodic information) of the front-end to the corresponding waveform rendition.

2.2.2 Naturalness and intelligibility

There are two key properties which are expected of any TTS synthesis system. One such property is naturalness – which is the degree to which the synthesized speech sounds close to speech uttered by humans [12]. The other is intelligibility – which is the degree of ease with which people understand the synthesized speech. Understandability is sometimes used in the place of intelligibility [12]. These

properties can also be complemented by three other concepts called flexibility, pleasantness, and similarity to original speaker. Flexibility has to do with how well the system handles symbols which need translation' for example, time phrases, out-of-vocabulary words and others. Pleasantness on the other hand, deals with the desirability and pleasure that one associates with listening to the synthesized voice sound. Similarity to original speaker deals with how close the synthesized voice compares to that of the original speaker.

2.2.3 Speaker-dependent and speaker-independent systems

Speech synthesis systems can be trained for speaker-independent, speaker-dependent or adaptive platforms. A speaker-dependent system is one that is trained on data from one particular speaker. A speaker-independent system on the other hand is trained on data from several speakers and can be used to synthesize text using any of the trained voices. An adaptive TTS synthesis system is one that allows a new speaker to be adapted based on trained data of a speaker-independent system hopefully using only minimal data from the target speaker.

2.2.4 Limited domain and open vocabulary

The TTS synthesis systems can be developed for either limited domain or open vocabulary platforms. Limited domain speech synthesis systems are those trained using data from a particular domain (e.g., medicine) to be used only for purposes relating to that domain [1]. Such systems have been proven to exhibit high performance, naturalness, intelligibility, and low word error rates (WERs). Although limited domain systems do not require a huge database, they cannot synthesize words not in their lowetcase database. An open vocabulary TTS synthesis system is one that is trained on general purpose data (though it may be optimized for a particular domain) from a particular natural language to be used for general purpose applications [1]. Unlike limited domain systems, open vocabulary systems are flexible in that they can synthesize even words not in their database. Open vocabulary systems, however, often require a huge database, more training data,

produce less natural speech than that produced by limited domain systems, and their WERs are often higher than those of limited domain systems.

2.3 Speech Synthesis Methods

This section discusses the different methods of TTS synthesis system design together with their associated pros and cons. This discussion is essential to the understanding of the selection of the method chosen for this research over its competitors.

2.3.1 Rule-driven and corpus-based methods

There are different kinds of synthesis methods that can be used when building a TTS synthesis system. Some of these methods require a *set of rules* to drive the synthesizer whereas others depend on *parameters* excised from the recorded speech corpus [3]. These classifications are called rule-driven and data-driven (or corpus-based) synthesis respectively. Examples of rule-driven synthesis include articulatory synthesis and formant synthesis. Examples of data-driven synthesis on the other hand include concatenative synthesis and HMM-based synthesis.

2.3.2 Articulatory synthesis

Articulatory synthesis simulates the human speech production system in its approach to speech synthesis [3]. It is motivated by how the human articulators such as vocal tract, nasal tract, lungs and larynx generate speech. Theoretically, this method is the best synthesis method as it focuses on the human speech production system. It is however the most difficult to implement and it is computationally very expensive. This method is not being given much research attention at the moment, but is more of a historical treasure. Currently its speech quality is not as good as compared to other synthesis methods, although Huang et al. speculate that it may actually be the best design method for the future [3].

2.3.3 Formant synthesis

Formant synthesis uses a source-filter model which varies parameters such as formant frequency, amplitudes and noise to produce speech [3, 11]. It is an acoustic speech synthesis method which uses the source-filter theory of speech production. This method varies the parameters of fundamental frequency, voicing, and noise levels over time to create an artificial speech waveform. It is very flexible in that it can produce an infinite number of sounds though with robotic qualities, i.e., monotone speech that is not very natural sounding [4].

2.3.4 Concatenative/unit selection synthesis

Concatenative synthesis generates speech by concatenating/joining pre-recorded speech segments from the speech database [4]. Units of different lengths ranging from phones to phrases may be concatenated [4, 3]. Each concatenation unit of choice has its own advantages and disadvantages. This method is known to produce the most natural sounding synthetic speech as compared to other methods. Its greatest drawback is that it requires a very large speech corpus for training the system, – which is very hard to collect and thus very restrictive – in order to achieve excellent results. Louw [11] used concatenated diphones when developing the IsiZulu TTS synthesis system.

2.3.5 Statistical parametric synthesis

The HMM-based speech synthesis is a statistical parametric model that extracts speech parameters from the speech database, trains them and produces the sound equivalent of the input text. This method has the advantage of being able to synthesize speech with various speaker characteristics, speaking styles, emotions, and still produce reasonably natural sounding synthetic speech. Although this method can produce natural sounding speech, the speech it produces does not beat the naturalness of the speech produced by the best unit selection synthesis methods. This method, however, makes it very easy to adapt new speakers and has

very little memory requirements. HTS is the toolkit that is used to develop HMM-based synthesis systems.

2.4 Chapter Outline

This chapter discussed historical developments in the design of TTS synthesis systems dating from the 4th century. It also looked at the different methods that exist for TTS synthesis system development. A discussion of the core speech synthesis concepts which could be viewed as a reference guide for the uninitiated reader in this field of speech technology was also presented.

Chapter 3

3 Language Structure

The purpose of this chapter is to give a brief overview of both the historical, current and grammatical structure of Xitsonga language, which is the language of focus in this project. The statistical estimates and biographic location of Xitsonga speakers are discussed in this chapter. The history of Tsonga orthography and the transformation of both the language and its orthography have gone through are also discussed. The grammatical structure of the language, which is the part of the language interest, is also reflected upon.

3.1 Brief Language History

Xitsonga is standard form of a language that is largely spoken in the Limpopo Province of the Republic of South Africa. The same language is known as Xichangana in both Mozambique and Botswana. The Xitsonga language speakers are known and referred to as Vatsonga/Machangana. Xitsonga is one of the eleven official languages of South Africa as per the country's constitution. The second largest population of Xitsonga speakers is found in Mozambique, with others found in Swaziland and Zimbabwe amongst some of the Southern African countries. It will of course be utter ignorance for anyone to limit one's view to only the fore-mentioned countries as the spread of Xitsonga speakers can possibly extend to almost the whole world - considering the current global state of migration and globalisation. The same may be true for any other natural language.

The 2001 census estimated the number of Xitsonga South African mother tongue speakers to be about 2 million (4.44%) of the estimated 41million South African population [17]. At the moment, the South African population is estimated at just over 50 million [18]. The Xitsonga mother tongue speakers are most likely over 2 million in number. In South Africa, Xitsonga mother tongue speakers are mainly situated around Limpopo (mostly in the Giyani region), Mpumalanga (Bushbackridge region)

and Gauteng Province. A map depicting these areas where in Xitsonga language is dominant is given in Figure 3–1.

Figure 3–1. Xitsonga language dominant areas in SA adopted from African Languages [19]

Xitsonga, like most South African languages is a tone (or tonal) language. What this means is that similarly written words have different meanings based on the tone (high or low) with which they are pronounced. Xitsonga orthography was first reduced into writing and published in 1883 by Georges Bridel and Cie in a book called “BUKU YA TSIKWEMBO TSAIWENA TINSIMU TA NHLENGELETANO” [20]. Just like in most other South African indigeneous languages, work to reduce Xitsonga into writing was done by missionaries. According to Bridel et al [20] the Sotho or Sesotho orthography contributed to the initial development of Xitsonga orthography as its orthography was already developed. This resulted in the initial Xitsonga orthography being closely linked to Sesotho orthography. A simple example could be drawn from the Sotho language word “bana” for “children”, which is written as “vana” in Xitsonga, but was originally written as “bana”, just like in Sotho. The first

Xitsonga mother tongue speaker to write a novelette called “Sasavona” in Xitsonga vernacular was Prof Rev C. T. D. Marivate in 1938 [20].

Xitsonga has got several dialects. Even before it was initially committed to writing there were already several dialects of the language. As can be seen in Figure 3–2 Xitsonga had three subgroups with seven dialects, from which the Nkuna dialect of the Tsonga subgroup was chosen to represent the written form of Xitsonga [21].

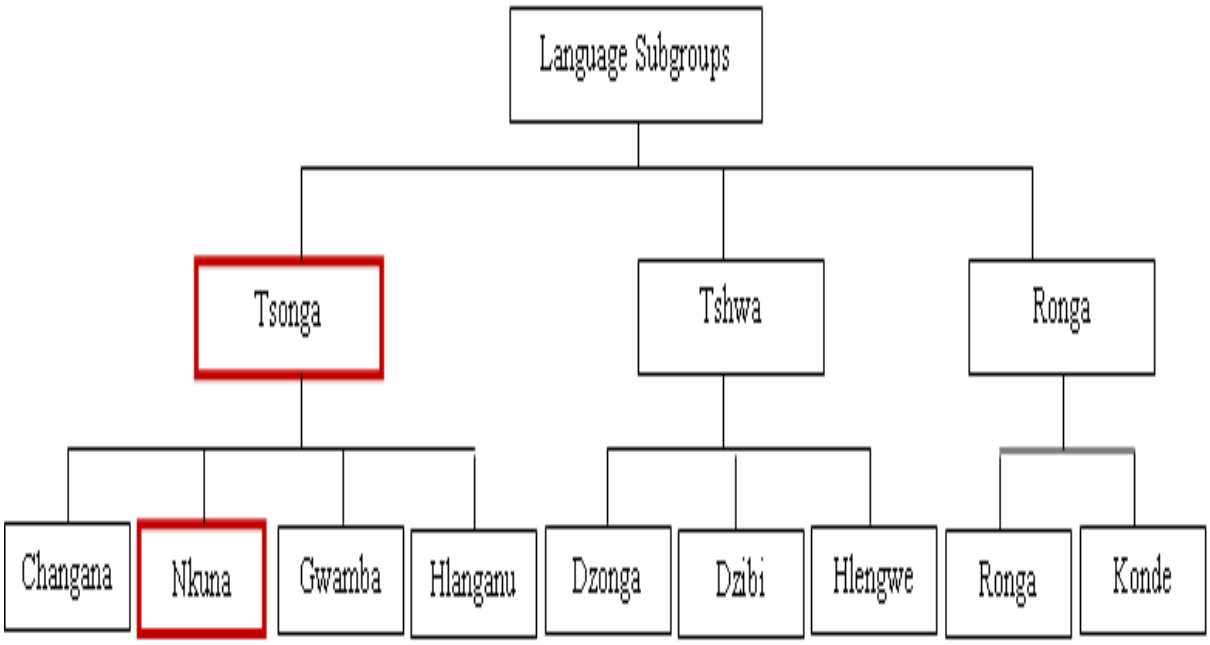

Figure 3–2. Xitsonga language groups and their dialects [21]

Initially, the Mission Council of Switzerland was the chief governing body on matters concerning language. Later on, different language boards and committees came on board; these includes the Tsonga language Board in 1954, the Autonomous Language Board in 1977, and now the Pan South African Language Board (PANSALB) amongst others [20, 22]. These bodies worked hard as the language evolved, having pioneered the unification of the Tsonga, Ronga, and Tshwa dialects in 1966 and deciding on making the Nkuna dialect the official written format of the Tsonga language. In a conference in 1898 it was said that the orthography of a language should be based on the way people speak “today” and not as they did in time past [20].

3.2 Language Structure

This section discusses the language structure of Xitsonga, with special emphasis on both the vowel and consonant structures. The importance of tone in Xitsonga is highlighted. Factors that differentiate the various phonemes (vowels and consonants) in Xitsonga are outlined.

3.2.1 Phonetic background

Xitsonga is made up of a complete set of 41 phonemes [22]. These phonemes include 5 vowels and 36 consonants. Mohasi defines a complete phoneme set in a natural language to be the minimum number of letters required to describe all possible word combinations in that language. Vowels are always voiced [1, 23, 22], whereas consonants can either be voiced or unvoiced [23]. Although Xitsonga is a tonal language, tone is not represented in day-to-day writings of the language. Words are rather differentiated based on the context of the sentence within which they are used. Only high and low tones are allowed in cases where in tone should be explicitly marked or represented [22]. The significance of tone can easily be seen with homographs, which are words that are written the same but pronounced differently [24].

3.2.2 Vowel structure

Vowels are the source of tone in Xitsonga. The five vowels used in Xitsonga are – *a, e, i, o,* and *u*. The five vowels can be further expanded by two extra vowels which represent the low tones of *e* (represented by Ɛ) and *o* (represented by Ɔ), as can be seen in Figure 3–3. According to [23]; Xitsonga vowels can be classified and described based on the raising/lowering of the tongue within the oral cavity, the stretching or relaxation of articulators, and the shape/positioning of the mouth when uttering a vowel. Figure 3–4 maps the structure of vowels given in Figure 3–3 with a diagram which includes a structure of the physical articulatory organs.

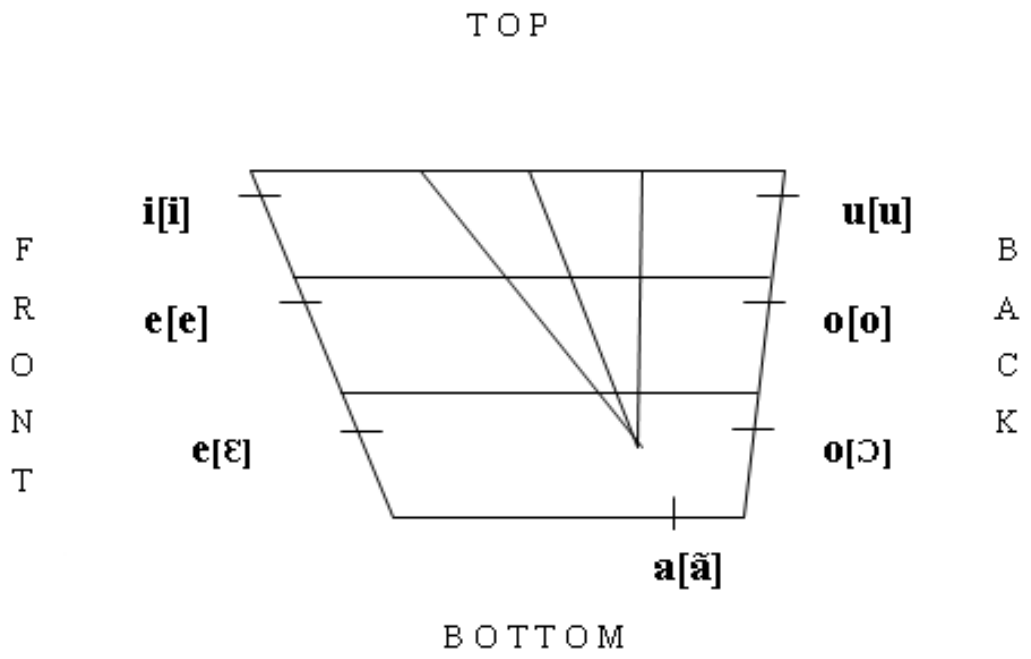

Figure 3–3. Xitsonga vowel structure [23]

The classification of vowels as given in [23] is described as follows:

- a[a] – is a back vowel. It is articulated with the articulators being stretched; the mouth half-closed and round, e.g., "het***a***"
- e[e] – is a mid-front vowel. It is articulated with the articulators being slightly stretched; the mouth half-closed and flat, e.g., "vukh***e***ta"
- e[ɛ] – is a half-open, mid-front vowel. It is articulated with the articulators relaxed and the mouth moving sideways, e.g., "v***ɛ***le"
- i[i] – is closed, top-front vowel. It is articulated with the articulators very stretched and the mouth being flat, e.g., "x***i***nkwa"
- o[o] – is a half-closed, mid-back vowel. It is articulated with the mouth being round and the articulators stretched, e.g., "***o***lela"
- o[ɔ] – is a half-open, mid-back vowel. It is articulated with the mouth being almost round and the articulators a little stretched, e.g., "nt***ɔ***ma"
- u[u] – is a closed, back vowel. It is articulated with the articulators being very stretched and the mouth extremely round, e.g., "namb***u***"

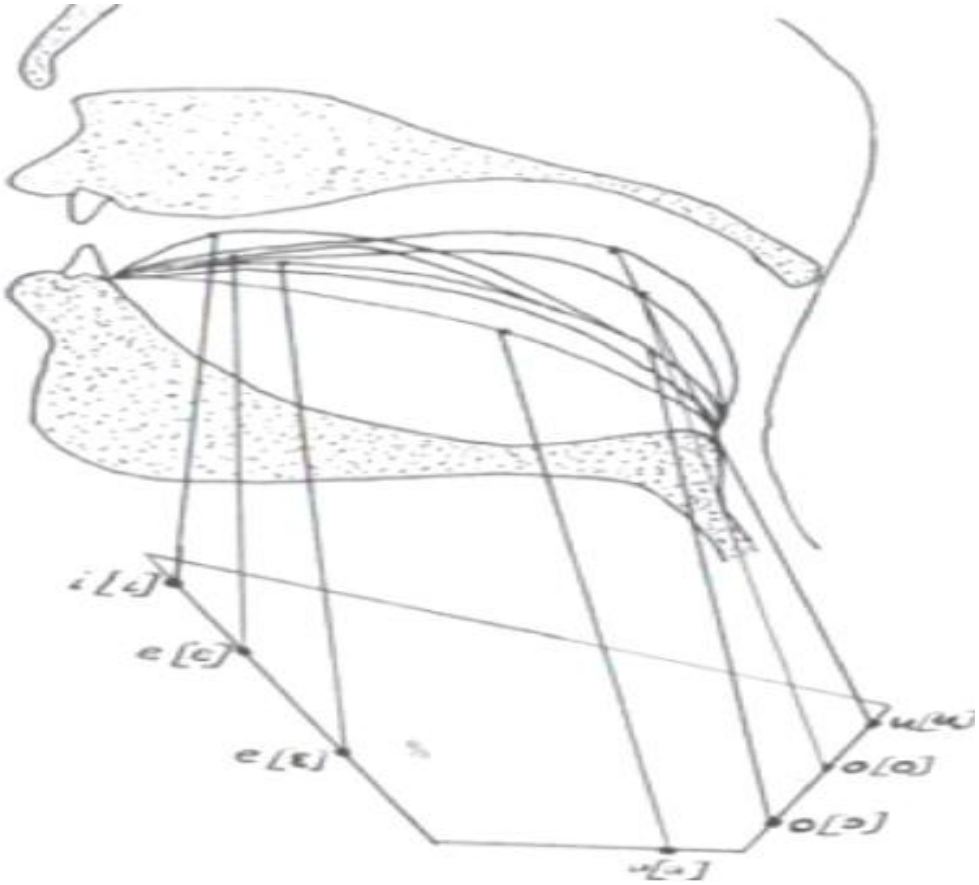

Figure 3–4. A mapping of the vowel structure and the oral cavity [23]

3.2.3 Consonant structure

There is no predefined standard for pronouncing or uttering consonants in Xitsonga. Vowels are responsible for shaping the way consonant are pronounced and used in Xitsonga. In English and some other languages, consonants can be pronounced without being complimented by vowels. An example could be drawn from consonant ‘k’ which is pronounced ‘khey’ in English. Consonants are classified based on the manner and place of articulation. Table 3–1 gives *Tshwa* consonants which is another dialect of Tsonga. The 36 consonants that are found in Xitsonga are briefly outlined below [22]:

{b, bv, c, d, dhl, dl, dz, f, g, gq, h, hl, j, k, l, m, n, n’, ny, p, pf, py, q, r, s, sw, t, tl, ts, v, vh, w, x, xj, y, z}.

A brief description of the consonants’ *place of articulation*:

- Bilabial – involves the coming together of both the upper and lower lips during the articulation of a consonant (e.g., b).
- Labiodental – involves the contact of the lower lip and the upper teeth, with the lip bent towards the oral cavity in the articulation process (e.g., f).
- Alveolar – involves the tongue tip coming into contact with the alveolar (teeth ridge) during the articulation of a consonant (e.g., t).
- Bilabio-alveolar – involves an articulation whereby both the upper and lower lips come into contact with the tongue tip hitting the alveolar usually producing a whistling sound (e.g., sw).
- Prepalatal – involves an articulation of consonants towards the front of the palate (e.g., x).
- Velar – involves the articulation of consonants with the back of the tongue touching the back of the soft palate (e.g., n').
- Glottal – involves the articulation of consonant sounds by closing the vocal folds (e.g., nhw).

A brief description of the consonants' *manner of articulation*:

- Explosives – involves consonants that are articulated by releasing an explosive sound (e.g., p).
- Implosives – involves the articulation of a consonant by releasing air from the lungs and moving the glottis downward (e.g., b).
- Laterals – involves the articulation of a sound by allowing air to only pass through the sides of a tongue while the other sides are blocked (e.g., hl).
- Fricatives – involves the articulation of consonants with air being forced through a small passage (e.g., f).
- Affricates – involves the articulation of consonant sounds that effect a plosive and a fricative concurrently (e.g., ts).
- Nasals – involves articulation with the soft palate closing air-flow through the mouth while allowing air to pass through the nasal cavity (e.g., n).
- Semivowels – involves the articulation of consonants by partially blocking the air through the teeth and tongue or the mouth (e.g., j).

TYPES OF CONSONANTS	VOICING	PLACE OF ARTICULATION								MANNER OF ARTICULATION		
		Bilabial	Labiodental	Alveolar	Bilabio-alveoar	Prepalata	Palatal	Velar	Glottal			
OBSTTRUENTS	VL	P		T						FEATURES	PLOSIVES	Explosives/
	VD			ɗ								Implosives
	VL											Intermittent
	VD			r				k g				Explosive
	VL			tl								Lateral
	VD			dl								Explosive
	VL			/								Implosive
	VD	ɓ		g/								
	VL			s						SECONDARY	(FRICATIVES)	(Central)
	VD		f v	z	ʂ □	ʃ ʒ			ɦ			Fricative
	VL			ɬ								Lateral
	VD			ɮ								Fricative
	VL		□f	ts	p□	tʃ						Affricate
	VD		□v	dz	dʐ	dʒ						
SONORANTS	VL VD	m	(ɱ)	N				ɲ	ŋ			Nasal
	VL VD			L								Lateral
	VL VD	w					J					Semivowel

Table 3–1. Xitsonga consonant structure (adopted from Tshwa [21])

3.3 Chapter Summary

This chapter discussed the history of the Xitsonga language orthography and its transformation. It also looked into the phonetic structure of the language. The building blocks of the Xitsonga language phonemes have been discussed. The relevance of the phonetic structure of Xitsonga to the development of a Xitsonga TTS synthesis system has been outlined.

Chapter 4

4 Hidden Markov Model Speech Synthesis Toolkit

An HMM-based synthesis method has been selected for use in this research project. Its flexibility in the ease of adaptability to speaker's voice characteristics and speaking styles, less memory requirements for the runtime engine and less speech data required for training the system influenced its choice and preference. The focus of this research project has been to produce acceptably natural sounding speech for the TTS system developed using an HMM-based approach. The HMM-based speech synthesis system (HTS) is made up of two main parts – the training and synthesis part – and the optional part, adaptation. The HTS toolkit was used for experimentation and the phases involved are discussed below.

4.1 Training Phase

In the first phase, spectrum (mel-cepstral coefficients) and excitation (log fundamental frequency or log F0) parameters were extracted from raw files in the speech database and their dynamic features (delta and delta-delta coefficients) calculated [2, 25]. The extracted parameters model speaker characteristics and speaking styles and they are used to train (or model) the context-dependent phoneme HMMs [26]. Spectrum parameters are modelled by multivariate Gaussian distributions, whereas excitation parameters are modelled by multi-space probability distribution hidden Markov models (MSD-HMMs) [2].

Mel-cepstral, log fundamental frequency (log F0), and state durations are simultaneously modelled in a unified framework but clustered in isolation using a decision tree based clustering technique called minimum distance length (MDL) [25, 26]. The MDL technique ties contextual factors (i.e., phoneme identity, stress-related and locational contexts) that are almost similar. This is done because it is both impractical and impossible to prepare a speech database that can model all combinations of contextual factors. A re-estimation of the clustered context-

dependent phoneme sequence will then be performed using the expectation-maximization (EM) algorithm [26]. Clustering is also used to generate excitation and spectrum parameters for newly observed vectors, i.e., observation vectors not included in the training corpus [7].

State durations are modelled by context-dependent n-dimensional Gaussian distributions which are then clustered by a decision tree. State densities capture/model the temporal structure of speech [25, 27]. Mel-cepstral coefficients, log F0 and state durations are modelled simultaneously in a unified framework of HMM [25, 27]. Context-dependent multi-space probability distribution (MSD) hidden semi-Markov models (HSMMs) are used to model feature vectors of both continuous and discrete HMMs of the F0 part and continuous HMMs of the spectrum part.

4.2 Adaptation Phase

A speaker adaptation technique is used to adapt a target speaker using the trained HMM from the training phase. MSD-HSMM also simultaneously models and adapts duration parameters for excitation and spectral parameters. Tikashi Masuko in [2] indicates that the adaptation technique requires only a small amount of speaker adaptation speech data from the target speaker. The adaptation process may be done using an individual speaker's speech data or by averaging several speakers' speech data [2]. The average voice model is adapted to the target speaker using constrained structural maximum a posteriori linear regression (CSMAPLR) and maximum a posteriori (MAP) adaptation techniques. Speaker voice characteristics, styles or even emotions can be modified/updated by transforming HMM parameters using adaptation or other methods (such as interpolation and eigenvoices) [26].

Adaptation can either be supervised or unsupervised. In supervised adaptation the word transcriptions of target speaker are known, whereas, in unsupervised adaptation the word transcriptions of target speaker are not known. The adaptation method used for this project is supervised adaptation [3].

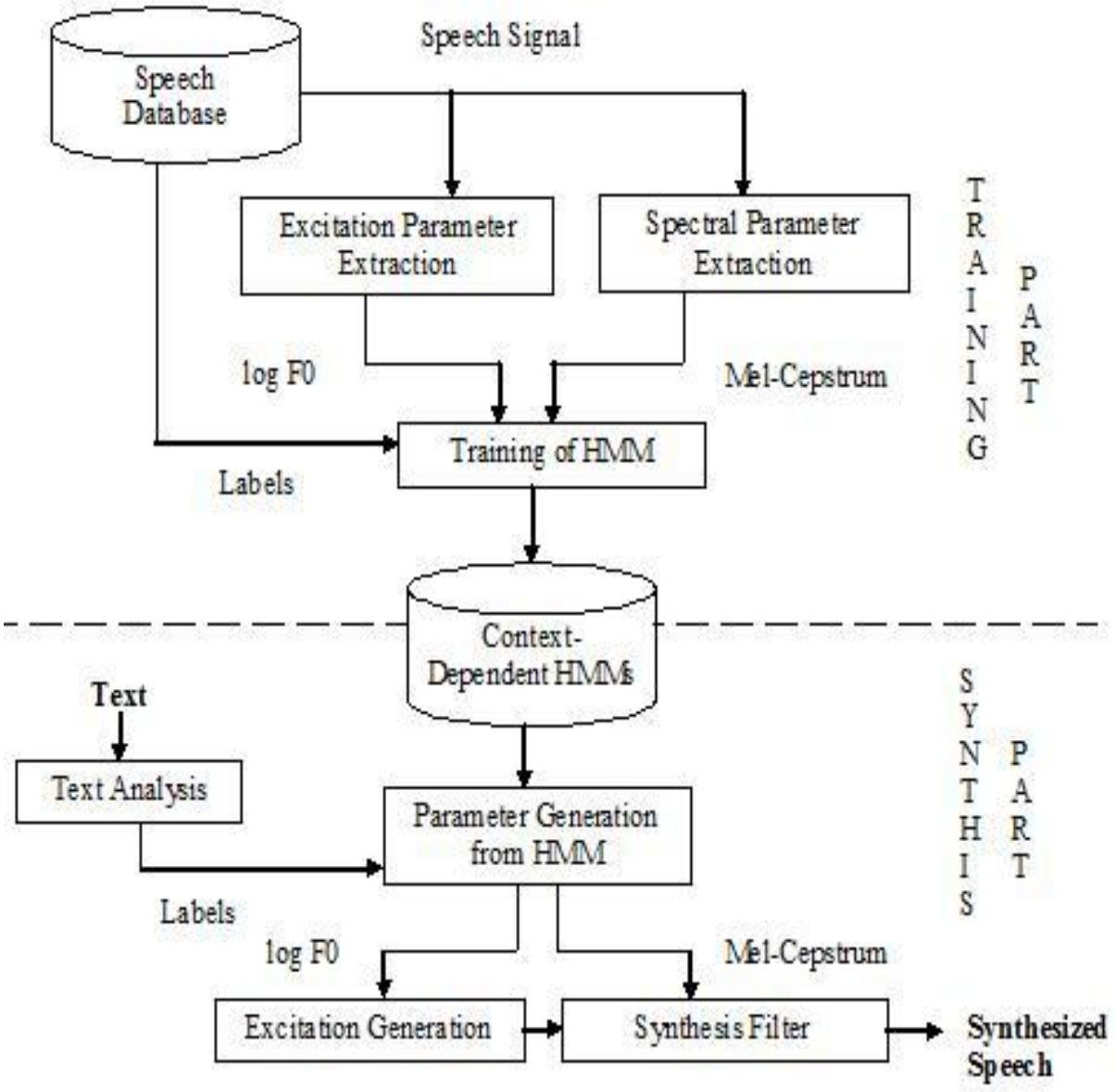

Figure 4–1. An HMM-based synthesis system [2]

4.3 Synthesis Phase

In the synthesis part, text is input, analysed and a context-dependent label sequence is produced. The context-dependent label sequence is used for concatenating the (adapted) context-dependent HMMs (from the adaptation phase) to produce a sentence HMM. State durations of the MSD-HSMM are automatically generated by probability density functions (PDFs) and they maximize their probabilities. State duration distributions are then used to determine state durations of the label sequence [26]. A speech parameter generation algorithm is then used to generate

mel-cepstrum, log F0, and aperiodicity-measure sequences from the context-dependent phoneme HMMs. The synthesis filter is used to synthesize a speech waveform from the generated excitation and the spectral parameters [25, 26].

The HTS toolkit does not, however, include a text analyzer [27]. For text analysis, there is a variety of text analyzers that could be used such as Festival, Flite, etc. Festvox together with festival are used for the text analysis part of the TTS synthesis system and their language specific portions are appropriately modified to conform to the Xitsonga language.

4.4 Chapter Summary

This chapter gave a presentation of the hidden Markov model speech synthesis. This is the method chosen for the development of the speech synthesis system for Xitsonga. The different components that make up an HMM-based speech synthesizer have also been discussed. The training phase is at the heart of this method, generating HMMs which are essential for synthesis. The adaptation phase, which is an optional component of this method, is also discussed. Finally, the synthesis phase – which outputs a speech waveform – is outlined.

Chapter 5

5 System Design

This chapter details the process undertaken to develop a TTS synthesis system for Xitsonga using hidden Markov models. It explains the whole process starting with the selection of sentences for recording and the recording process. It then looks into the preparation of the files and data required for training the system. It further lists the software required for experimentation and also discusses the step-by-step implementation phase of the research project.

5.1 Data Collection and Preparation

This section discusses the preparation details of the data and files necessary for successful experimentations conducted in this research project.

5.1.1 Sentence selection

Phonetically balanced sentences in Xitsonga were collected. These sentences were collected from different sources, such as PANSALB's "MILAWU YA MAPELETELO NA MATSALELO YA XITSONGA" [22] and C.P.N. Nkondo's "Xiletelo xa Xitsonga" [23]. Others were composed by the author based on everyday spoken language. The collected sentences were then numbered and put into a festvox file called *data.txt.done* which is used for creating prompts. An example of the file *data.txt.done* can be found in Appendix A, and the path to the file is *ul_tso_bn/etc*. The selection of sentences was done bearing in mind the general-purpose nature of the desired TTS synthesis system. a manual process was used for the selection of phonetically balanced sentences. The maximum number of recorded sentences was 453. These sentences also catered for a little redundancy for some of the phonemes.

5.1.2 Speech recording

The recording process took place in an office environment with minimal noise. Praat is the software that was used to record the speech corpus. A regular microphone (Sennheiser) and a normal office computer made up the hardware equipment used to record the speech corpus. The Xitsonga phonetically balanced sentences were used for recording. Each speaker read out a certain number of sentences (practiced and/or unpracticed). The speech sample was recorded at 44.1 KHz stereo and the files stored in waveform format (.wav). The waveform files were normalized and changed to conform to 16 KHz, 16 bit, RIFF format as required by the festvox system and to make it easier to create raw files of small sizes. The command used to achieve this normalization is *bin/get_wavs recording/*.wav*. The new wave files were then converted to little endian raw files using **Praat** – see Appendix B.

The recorded speech by five speakers – two males and three females – went through the *.wav to .raw* file conversion process. The speech corpus of the other speakers from the five speakers that was already available was also recorded under the same office conditions and using the same equipment. This corpus was used to compensate the newly recorded data both for training the system and adapting the voice of one female speaker. The recording process occurred over a number of days for all the speakers in order to capture the variability in the speakers' speaking characteristics. The number of recorded sentences for the different recording subjects varied from one subject to another. A detailed numerical specification of these variations is given in Table 5–1.

The recorded speech sounded rather low (in loudness) as compared to the sound acquired when recording using a built-in laptop microphone. The low sound of recordings is attributed to the quality of the microphone used and it is expected that microphones of better quality could have produced better results.

5.1.3 Labels and utterances

From the sentences listed in *data.txt.done*, festvox generates prompts using the festvox command *./bin/do_build build_prompts*. This command returns among other things the initial set of labels and utterances in the directories */prompt-lab* and */prompt-utt* respectively. This initial set of labels and utterances is not well aligned/transcribed as can be seen in Figure 5–1 and the labels are not context dependent; (see examples of the initial labels and utterances in Appendix B). More accurate labels and utterances are then generated using the festvox commands *./bin/do_build label* and *./bin/do_build build_utts* respectively run consecutively. The newly generated labels will be used to generate utterances which will, in turn, generate context-dependent labels. These labels are automatically aligned/transcribed using an **EHMM** labeler and are found on the path *ul_tso_bn/la*, (see Figure 5–2 for a diagrammatic representation). The utterances are also generated automatically based on the label files in *ul_tso_bn/lab* and are themselves found on the path *ul_tso_bn/festival/utts*. Examples of these newly generated utterances and the context-dependent labels derived from these utterances are given in Appendix B as well.

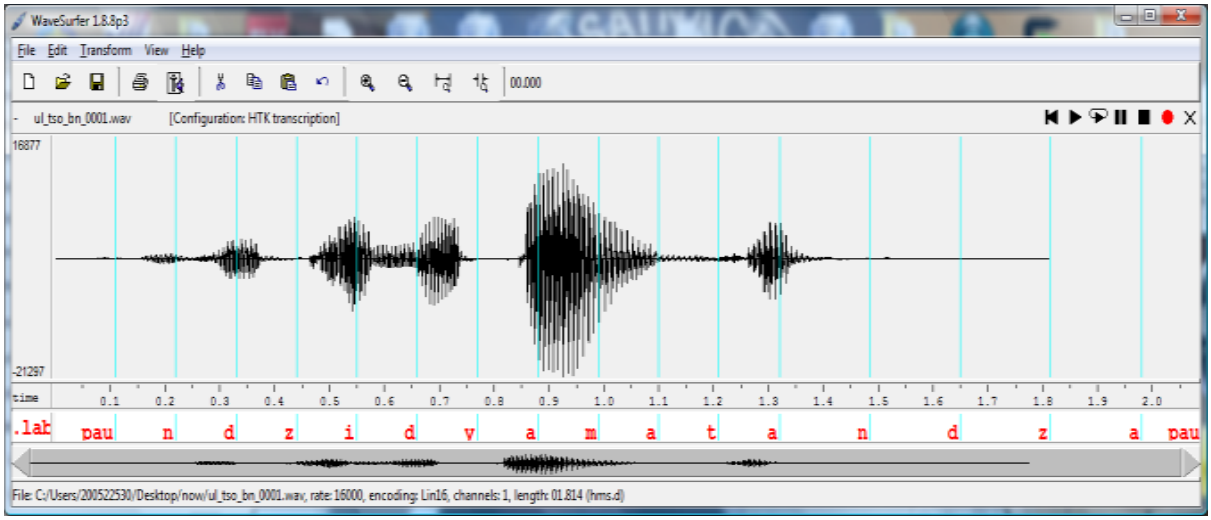

Figure 5–1. Initial festvox labelling of the sentence "ndzi dya matandza" /I [am] eat[ing] eggs/

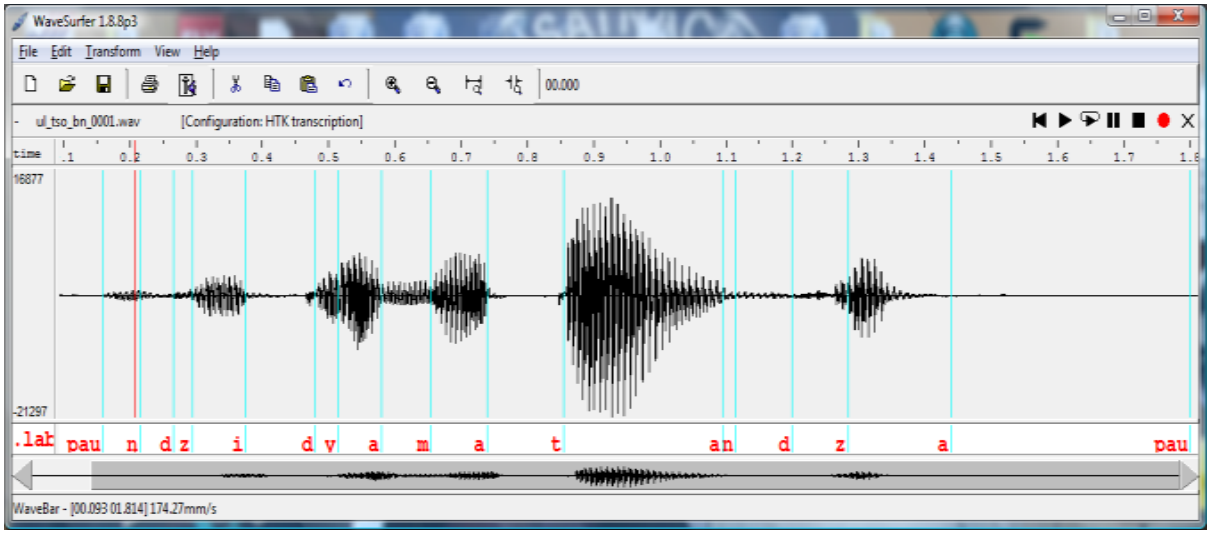

Figure 5–2. EHMM labeller applied to the sentence "ndzi dya matandza" /I [am] eat[ing] eggs/

There are several algorithms that can be used for automatic labelling in addition to EHMM labeler. An alternative to using automatic labelling can be to hand-label the speech corpus. Hand labelling is a very tiresome task and it requires superb expertise. The Northern Sotho TTS system only had 43 of its 400-sentence corpus hand labelled [12]. For our research project, festvox's EHMM labeller was considered sufficient for its automatic labelling. Hand labelling can produce even better results especially when applied to improve on an already automatically labelled label. Hand labelling was not used for this project due to time constraints and its need for special skilled human resources.

It is important to note that the word *utterance* is used in two different contexts in this document. The first is an utterance (.utt) file from which labels can be derived. The second refers to a speech rendition synonymous to a person uttering a word or a machine rendering (uttering) a speech waveform.

5.1.4 Question set and phone set radio

The properties of both consonants and vowels are significant to the design of both the phone set radio and the question set files. Both these files capture almost the same information, but the difference lies in the representation of the information. It is in these files that the classifications of vowels and consonants such as voicing, place of articulation, manner of articulation, lip rounding, etc., are captured.

A question set required for the HTS-demo was created for Xitsonga language using the English HTS-demo format for creating the question set. An example of the question set is included in Appendix C. The question set is found on the HTS-demo under the path */data/questions* and the file is named *questions_qst001.hed*. The question set uses a method of grouping phonemes with the same properties together (Appendix C). The question set was created based on the language structure for Xitsonga and more details about the question set will be explained on the next chapter.

The Festival system installation includes a file named r*adio_phones.scm* in its library (lib) folder. This file needs to be changed to conform to the rules of the language being used in order to correctly train and synthesize the text entered for synthesis. This file is required to use the same language units (phonemes – vowels and consonants) as used in the question set file. The phone set radio (*radio_phones.scm*) was also changed to conform to Xitsonga language and a portion of this file is included in Appendix D. This file captures the details of the entire phoneme set using predefined special characters, e.g, +, -, 0, 1, 2, 3, s, l, d, etc., to represent certain properties.

5.1.5 Tokenization

There are a lot of symbols in every language that often require translation to their equivalent word format before being spoken or synthesized. Such symbols include *numbers, dates, time, money, acronyms, etc.* To achieve this task, a template file named *ul_tso_bn_tokenizer.scm* included in festvox has to be modified to cater for these needs. Indeed a general-purpose system should be able to convert and synthesize such symbols without difficulty. However, a finite set of such symbols is difficult (if not impossible) to collect, especially for acronyms and abbreviations, even more, so in an instant where one has not created a pronunciation dictionary, like in this study. Such a file was also modified to include common symbols in Xitsonga and a snapshot of this file can be seen in Appendix E.

5.1.6 Other processes

For the HTS-demo, the context-dependent structure of Xitsonga was not changed since it is the same as that of English, i.e., *phoneme -- syllable -- word -- phrase -- utterance*. The context-dependent labels are generated from utterances using festival's *dumpfeats* function when using the HTS-demo. The speech signal was windowed at 25 ms Blackman window with a 5 ms shift. Five-state, left-to-right context-dependent multi-stream multi-space probability distribution hidden Markov models (MSD-HMMs) were used. Different instances of the same phoneme HMM utterances may be stored in the model file(s) in order to represent different contexts.

5.2 Software Packages

This section lists the software packages required for installation in order to set up the working environment for the research project. It also discusses the experimental environment of the project. A brief descriptive discussion of the software packages is also given.

5.2.1 Software package listing

A few software installations were required in order to create a working environment for launching the present research project. The project was set up on a normal office desktop computer with 2GB RAM, 1.80GHz Intel (R) Core (TM) CPU 4300. The computer was running on an *Ubuntu 11.04* operating system. Several software packages necessary to the success of this research project were acquired and/or downloaded and installed. Appendix F details the unpacking and installation of these software systems. The support for most of the packages can be found at their respective sites' mailing lists as one might face some challenges in using some of the software. For the majority of the software packages, even their recent versions would work well in the design of a similar project. The software components listed below were downloaded for installation as they were key to the success of the research project and they can be downloaded from their respective sites.

Key software packages:

- speech_tools-2.1-release.tar.gz [15]
- festival-2.1.tar.gz [16]
- festlex_CMU.tar.gz [16]
- festlex_POSLEX.tar.gz [16]
- festvox_kallpc16k.tar.gz [16]
- festvox_cmu_us_awb_arctic_hts.tar.tar [16]
- HTK-3.4.1.tar.gz [28]
- Hdecode-3.4.1.tar.gz [29]
- HTS-2.2_for_HTK-3.4.1.tar.tar [30]
- SPTK-3.4.1.tar.gz [31]
- hts_engine_API-1.06.tar.gz [32]
- ActiveTcl8.4.19.4.292682-linux-ix86.tar.gz [33]
- STRAIGHTtrial.zip [34]
- MATLAB R2011b [35]
- sox-14.3.2.tar.gz [36]
- HTS-demo_CMU-ARCTIC-SLT.tar.tar or HTS-demo_CMU-ARCTIC-SLT_STRAIGHT.tar.tar [30]
- HTS-demo_CMU-ARCTIC-ADAPT.tar.tar or HTS-demo_CMU-ARCTIC-ADAPT_STRAIGHT.tar.tar [30]
- festvox- 2.1-release.tar.gz [37]

Extra software packages:

- Praat [38]
- Wavesurfer [39]

5.2.2 Software package description

Speech tools – A library of C++ functions for the speech processing of related speech objects. It is free software developed and maintained at the University of Edinburgh's Centre for Speech Technology Research. It is used for reading, writing, converting and supporting speech processing objects such as fundamental frequency, waveform, labels, etc. [15].

Festival – It is a baseline system that provides a platform for developing TTS systems. It is a multilingual TTS system which is also a package that allows for the development of new systems with ease [16]. Similar to speech tools, festival is free software with a licence that allows for unrestricted usage. Two lexicons (festlex_CMUand festlex_POSLEX) and voices (festvox_kallpc16k and festvox_cmu_us_awb_arctic_hts) were included in order to allow festival to speak. There are, however, many voices and lexicons available for download that can be used with festival.

HTK – It is a toolkit that was primarily designed for use in speech recognition research for building and manipulating hidden Markov models [28]. HTK was originally developed at Cambridge University and can be used in many different applications including speech synthesis. HTK can be downloaded for free, but it requires one to first register with a valid email address and agree to its licence.

HDecode – HDecode is a package that is an add-on to HTK; it requires one to have registered as an HTK user, and also agree to its licence in order to download it [29].

HTS – HTS is an HMM-based Speech Synthesis System toolkit that is designed to be patched to HTK [30]. HTS is not a standalone software system, as a result it is required that once patched to HTK one should also agree to and obey the HTK licence.

SPTK – It is a software package that comprises speech signal processing tools [31]. It is freely downloadable and is released under the Modified BSD licence.

hts_engine – It is software that is used to synthesize speech waveform from HMMs output by HTS [32]. It is freely downloadable and is released under the Modified BSD licence.

ActiveTcl – TCL stands for Tool Command Language. It is a portable interpreter which is freely downloadable [33].

STRAIGHT – It stands for Speech Transformation and Representation using Adaptive Interpolation weiGHTed spectrum; and it is a speech vocoding method. There is a trial version [34] of the software which can be used for the HTS-demo, but the software itself is not free, although one can ask for the source codes for research purposes from Prof. Kiwahara.

Matlab – It is the language of technical computing that is used for computationally intensive tasks [35]. It has components that support signal processing, data analysis, data visualisation and many other capabilities. Matlab is not a free software package, so for one to access and use it one has to purchase it.

Sox – It is a command line utility that is famous for the conversion of different audio file formats [36]. It can also be used to record, play and manipulate audio files. Sox is a free software.

HTS-demo – It is software used to train and synthesize a TTS system using HTS [30]. Most of the software listed and discussed in this chapter are meant for use by this package. It is also freely downloadable.

Festvox – It is a project that is based on festival and is aimed at making the task of building a new voice easy [37]. All the software components that it uses are free and unrestricted.

Praat – It is free software for sound manipulation, phonetic and acoustic analysis [38].

Wavesurfer – It is a toolkit used for sound visualisation and manipulation [39]. It is very convenient in displaying waveforms, transcriptions, etc. It can also be used to convert speech files from one form to another. It is open source software.

5.3 Implementation

This section takes one through the process taken to develop our TTS synthesis system by assuming that the installations, as outlined in Appendix F, of the packages discussed in the previous section have already been done. A detailed approach ranging from the actual data preparation to the commands used to run the system is given.

5.3.1 Initial stages

Now that all the necessary software packages have been installed, a step-by-step narrative approach is taken into the development of the TTS synthesis system based on hidden Markov models. Firstly, *a set of 545 phonetically balanced sentences* in Xitsonga was compiled. These sentences were collected mainly from the general domain looking into everyday use of the language and considering some of the frequently used sentences. This list was divided into two sets making up the training and testing sets. The initial set would be used later in the construction of a prompt list file called *txt.done.data*. The sentences were recorded using a software package called Praat. These recordings were done at 44.1 KHz, in a normal office environment using an ordinary desktop computer and a head mounted microphone.

A phone set radio file was then created according to the language structure of Xitsonga. This file was named *radio_phones.scm* and copied to the path */festival/lib* where it overwrote its counterpart. It could be a good idea to first rename the file *radio_phones.scm* under festival's lib directory before copying the newly created file therein. The old phone set radio file could also offer some light on how to create a phone set radio file for a different language.

5.3.2 Festvox

Festvox is actually based on festival. One can look at festvox as festival made easy in the creation of new voices for any natural language. All the processes defined in

this section are documented in a manual called “Building Synthetic Voices”, by Black and Lenzo [40].

Every time one uses festvox, it is a requirement to first set the two environmental variables FESTVOXDIR and ESTDIR. These variables point to the paths where festvox and speech tools are located. In our case the following paths were used to initialize the variables.

```
export ESTDIR=/home/ntsako/speech_tools
export FESTVOXDIR=/home/ntsako/festvox
```

In order to start a new project, a new folder was created. The created folder is required to follow a particular standard, whereby, the folder name should start with the name of the institution, followed by the language, and concluded by the surname and name of the subject/owner in abbreviations. In this case the name of the institution is University of Limpopo (**ul**), the language is Xitsonga (**tso**), and the surname and first name of the primary subject are Baloyi Ntsako (**bn**). The newly created folder was then navigated into. The following commands perform two operations which are to create a new folder and navigate into it respectively.

```
mkdir ul_tso_bn
cd ul_tso_bn
```

A template or skeleton of directories and files were then created. The template files and directories were automatically created by running the command.

```
$FESTVOXDIR/src/clustergen/setup_cg ul tso bn
```

After creating the template files, it is important to then transform them to suit the present project’s specific and special needs. The collected phonetically balanced list of sentences was used, in order to create a file called *txt.done.data* stored under */ul_tso_bn/etc*. A snapshot of the file can be found in Appendix A. An example of the structure used in creating the file is given by the following:

(ul_tso_bn_0001 "Swa khenseka hinkwaswo")
(ul_tso_bn_0002 "inkomu")

The above structure starts with an opening bracket, followed by the festvox project folder name and sentence number separated by an underscore (_), the sentence is included as a string delimited by double quotes at the beginning and end, and concluded by a closing bracket. Another file called *ul_tso_bn_tokenizer.scm* found under */ul_tso_bn/festvox* was created. A description of this file is given in section 5.1.5 and a snapshot of the file is given in Appendix E. Other files which were modified include *ul_tso_bn_lexicon.scm* and *ul_tso_bn_phoneset.scm* found under the same folder as the file *ul_tso_bn_tokenizer.scm*. A pronunciation dictionary was not created for this project and the system depended solely on grapheme-based rules defined in the *ul_tso_bn_tokenizer.scm* file.

The recorded files were then placed under the folder called */Recordings* which is under */ul_tso_bn*. The recorded wave files were then normalized and down sampled to conform to a *16 KHz, 16 bit, Resource Interchange File Format (RIFF) format*. The resultant wave files are then stored on the path */ul_tso_bn/wav*. The command used to achieve this is given below.

bin/get_wavs recording/*.wav

Next, the prompt files were created, whose output was saved in the folders starting with the keyword *prompt* under */ul_tso/bn.* This command used the tokenizer, the phone set radio, the sentence list (*txt.done.data*) and other files to create the prompts. The initial label and utterance files are a result of the following command:

./bin/do_build build_prompt

In order to have context-dependent labels, another command was executed/run; this command gave the final set of labels which would then be used to create utterances from which the desired labels would be obtained. These labels are automatically generated using the EHMM labeler. The resulting labels are stored on the path /ul_tso_bn/lab. These labels are used to create utterances by using a different

command. The commands used for creating labels and utterances are given below in their sequence.

./bin/do_build label
./bin/do_build build_utts

The utterance files just created are in the format that can be used by the **HTS-demo** and those utterances can be found on the path */ul_tso_bn/festival/utts*. The HTS-demo also requires raw (*.raw*) files. The raw files that are generated by using the newly generated wav files in */wav* of */ul_tso_bn* are much smaller in size as compared to those generated from the wav files in */recording* of */ul_tso_bn.* The raw files generated from the */wav* directory worked fine on HTS-2.1.1, whereas the ones generated from the */recording* directory gave problems with forced alignment due to their bigger size. Conversely, HTS-2.2, seemed to prefer raw files generated from the wav files in */recording* and returned an error/warning when using those generated using wav files in */wav*. The reason for this is, HTS-2.1.1 used the sampling rate of 16 kilohertz (KHz), whereas HTS-2.1.1 is defaulted to use the sampling rate of 48 KHz. The original wave files recorded at 44.1 KHz were loaded into Praat and converted into raw format as depicted in Figure 5–3. The preliminary results used in the research paper presented at the Southern Africa Telecommunication Networks and Applications Conference (SATNAC) by Baloyi [41], were based on raw files obtained from the 16 bit, 16 KHz, RIFF wav files which were down sampled from the 44.1KHz recordings using festvox's command (*bin/get_wavs recording/*.wav*).

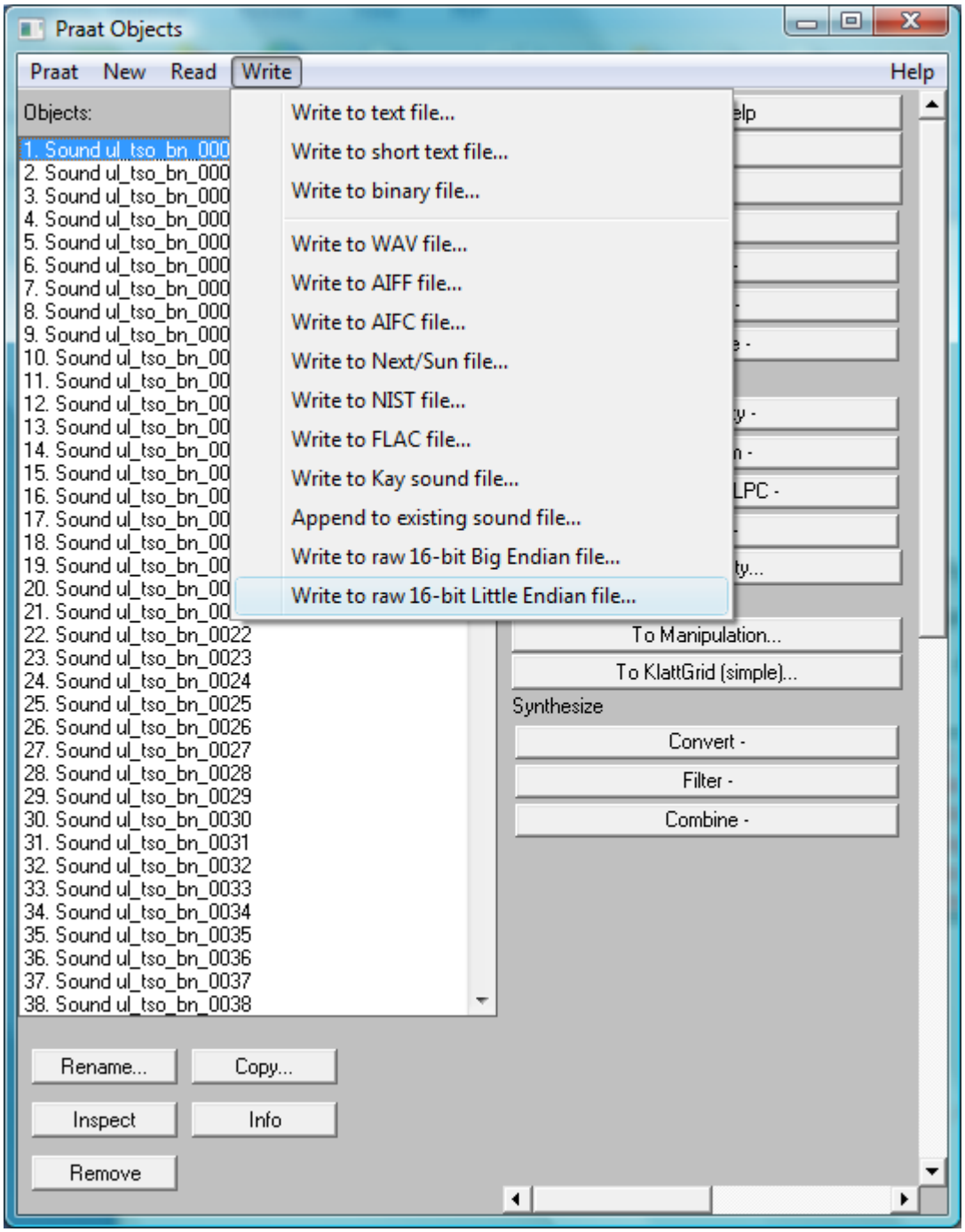

Figure 5–3. Wave to raw conversion using Praat

5.3.3 HTS-demo

With most of the files already prepared the HTS-demo was unpacked using the ***tar*** command. The question set was then created, as described in chapter 4 and in section 5.1.4, as well as depicted in Appendix C. The question set named *questions_qst001.hed* was replaced with the question set relevant for the present

project and having a name similar to the one of the deleted file, on the path */data/questions* of the HTS-demo. All files on the paths */data/labels/gen*, */data/utts*, and */data/raw* were deleted. Test labels, utterance files, and raw files specific/relevant to the present project were then copied into their respective emptied folders. The path */data/labels/gen* contains context-dependent labels corresponding to the test sentences.

Now that all the necessary files were in the right places, it was time to start running the HTS-demo. The first stage in running the HTS-demo is to configure it. Configuring the HTS-demo refers to setting the paths to the packages required to run it. The *INSTALL* file on the HTS-demo gives a guideline on how to set up these paths and alter certain parameters. It also indicates the packages required to run the HTS-demo and gives the universal resource locators (URLs) for their download. The configure statement used for this project is given as follows:

```
./configure --with-matlab-search-path=/usr/local/MATLAB/R2011b/bin \
            --with-straight-path=/home/ntsako/STRAIGHTV40pcode
            --with-tcl-search-path=/home/ntsako/tcl/bin \
            --with-fest-search-path=/home/ntsako/festival/examples \
            --with-sptk-search-path=/usr/local/SPTK/bin \
            --with-hts-search-path=/usr/local/HTS-2.2beta/bin \
            --with-hts-engine-search-path=/home/ntsako/hts_engine_API-1.06/bin \
```

The configure statement used for this project. It should be noted that the first two parameters are only applicable when using the HTS-demo with STRAIGHT vocoding and the third parameter should be excluded when using this HTS-demo, i.e., all the text in red font. The normal HTS-demo will, therefore not require only the first two parameters of the *configure* statement, i.e., all text highlighted in turquois.

There are certain files which needed to be modified in order to run the demo successfully. In order to be able run the HTS-demo using data specific to the present study, values of certain variables had to be changed in the files *makefile* and *config.pm* found in the directories */data* and */scripts* respectively, which are on the main HTS-demo directory.

makefile	***Config.pm***
# setting *SPEAKER = bn* *DATASET = ul_tso*	# only for HTS-demo with STRAIGHT *$CONVM = 0;* # converting mmfs to the hts_engine file format *$ENGIN = 0;* # synthesizing waveforms using hts_engine

After all the initialization preparations were done, it was then time to run the HTS-demo. Using the bash terminal the main HTS-demo directory was navigated into by using **cd** command and the HTS-demo was run by using the **make** command. The commands used to achieve these two operations are:

```
cd HTS-demo_CMU-ARCTIC-SLT
make
```

When the training process completes, the hts_engine synthesized the waveform of the test labels (sentences) which were on the path */data/labels/gen*. The synthesized wave files are found on the path */HTS-demo_CMU-ARCTIC-SLT/gen/qst001/ver1/hts_engine*. The demo usually takes several hours to days for successful completion on a normal desktop computer. In order to determine whether the demo is finished processing, one should check the log file on the main HTS-demo directory and if the last lines on the log file are similar to the ones below, then *training.pl* script of the HTS-demo may be done with processing.

```
.
Synthesizing a speech waveform from ul_tso_bn_0510.mgc and ul_tso_bn_0510.lf0...done
Synthesizing a speech waveform from ul_tso_bn_0522.mgc and ul_tso_bn_0522.lf0...done
done
```

5.3.4 HTS-demo_ADAPT

Most steps and changes that were applied to the normal HTS-demo are also applicable to the HTS-demo_ADAPT. This section, therefore, discusses only those changes that are unique to the HTS-demo_ADAPT by assuming that the rest of the details can be implemented the same way as in the HTS-demo. In order to prepare utterances (.utt) of a different speaker, **bn** was changed to the identifier of the appropriate speaker in (ul_tso_**bn**_0001 "...") of txt.done.data. The new speaker's identifier had to be limited to exactly two characters as that was the naming convention chosen for this project. This was done because of the changes made to */scripts/config.pm*, wherein *$spkrPat* was assigned a new value as follows: *$spkrPat = "\"*/ul_tso_**%%**_*\""*.

For the adaptation demo, data by five speakers was used both for training and adaptation. Speech recorded by two male and two female speakers was used for training. Data from a female voice of the fifth speaker were used for adaptation. Each of the five speakers had a unique identifier of two characters composed from their last name and first name. Table 5–1 records the number of recordings made by each of the speakers, their unique identifiers and the purpose for which the data were used.

Identifier	**Category**	**Number of Recordings**
Bn	Training	453
mk	Training	131
mt	Adaptation	63
mn	Training	440
tw	Training	452

Table 5–1. The number of utterances used for training and adaptation

Similar to the normal HTS-demo, the HTS-demo_ADAPT was also unpacked using the tar command on the terminal, after navigating to the folder where the HTS-demo_ADAPT was. All files and directories on the paths */data/raw* and */data/utts* were replaced with raw and utt files of the five speakers. Each speaker's data was contained in directories identified by their unique identifier. The label files on the path

/data/labels/gen, were also replaced by our own label files. Several changes that were made to both the *makefile* and *config.pm* files are given in the following segments:

Makefile

```
# setting
DATASET = ul_tso

# list of speakers
TRAINSPKR = bn mn mk tw
ADAPTSPKR = mt

# F0 search ranges (spkr1 lower1 upper1 spkr2 lower2 upper2 ... )
# Order of speakers in F0_RANGES should be the same as that in ALLSPKR
export F0_RANGES=bn 40 210  mn 40 280  mk 110 280  tw 40 210  mt 110 280

$${spkr}/$(DATASET)_$${spkr}_*.cmp;
# Delete $(ADAPTHEAD)
```

config.pm

```
$spkrPat = "\"*/ul_tso_%%_*\"";      # speaker name pattern

# only for  HTS-demo_ADAPT  with STRAIGHT
$CONVM = 0; # converting mmfs to the hts_engine file format
$ENGIN = 0; # synthesizing waveforms using hts_engine
```

5.4 Live Xitsonga TTS Demo

Any TTS synthesis system should offer some means to receive text and produce the corresponding synthetic speech as output. A live demonstration of the Xitsonga TTS synthesis system was developed. The graphical user interface (GUI) of the TTS synthesis system has been developed using java programming language. The internal logic of the TTS synthesis system was developed using Perl programming language and terminal (or console) commands. The text analysis part of the system, which is the most time consuming, was developed using festvox. The TTS synthesis system takes in either direct text or file input and then synthesizes the speech waveform as depicted in Figure 5–4. The hts_engine was used to synthesize the speech waveform. Sox was used to produce the equivalent waveform file of the synthesized speech.

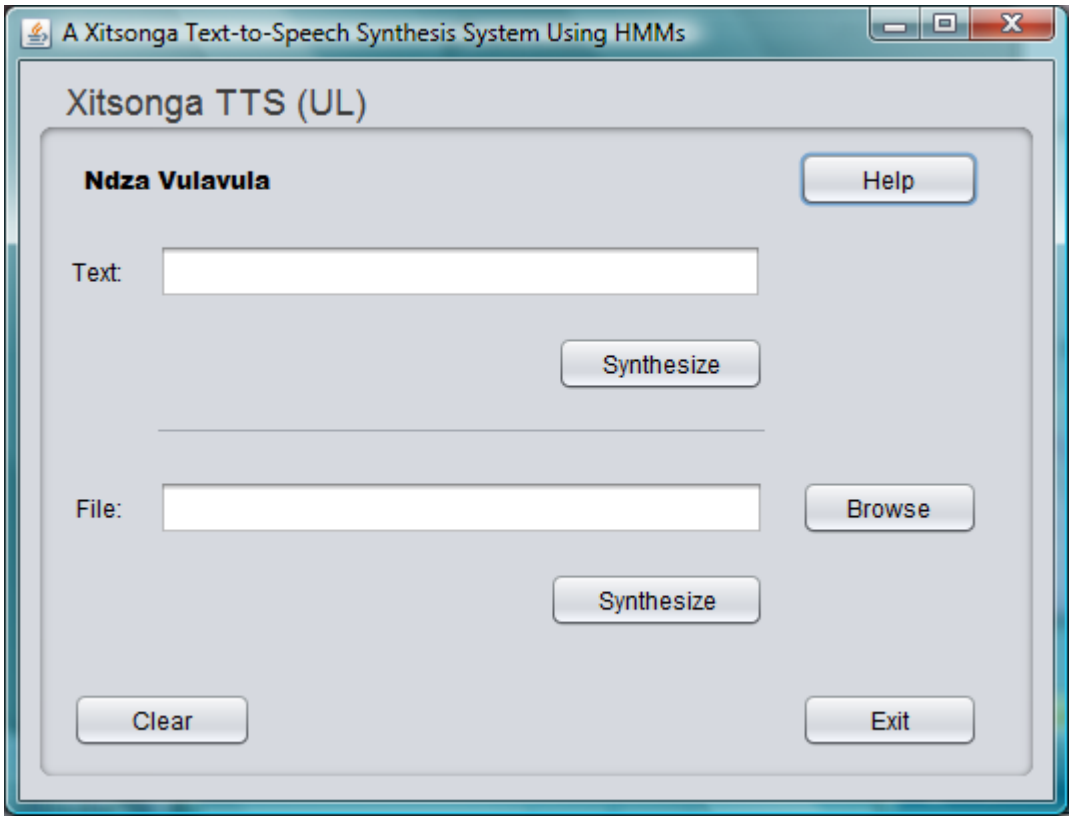

Figure 5–4. A Xitsonga TTS system live demonstration application

This application used festvox and festival for text analysis. The utterance to labels conversion was done using festival's dumpfeats function similar to the way it is done in the HTS-demo. In order to synthesize speech using the trained HMMs, hts_engine was used with the voice files that were output on the HTS-demo's */voices* directory. The live demonstration application has been created for the normal HTS-demo only.

5.5 Chapter Summary

This chapter discussed the process of developing a TTS synthesis system. It started by discussing the data and files that required preparation. It then advanced to the discussion of the packages required of installation with reference to the actual installation of the software. A hands-on approach of the actual system development and implementation was also discussed. This chapter was concluded by discussing a live demonstration of the TTS synthesis system.

Chapter 6

6 Evaluation

This chapter discusses the methods used for evaluating the TTS synthesis system developed. It also details the criteria used to select test sentences and evaluation subjects. The actual evaluation of the results together with their analysis is then performed.

6.1 Testing Procedure

The most popular and effective way of evaluating TTS synthesis systems is through *listening tests*. This is significant in that potential end-users are made key to the evaluation process. Moreover, the popularity of such systems is dependent on whether ordinary users find them easy and pleasant to use. Normally, in the discipline of Computer Science (specifically in Software Engineering), there are different methods which can be used to test software. The test method used in our research project synthesis is very closely related to software engineering's user *acceptance test* [42]. In TTS systems research/design there are certain factors that system designers might deem important focus areas, whereas general users deem such factors as insignificant; hence, the need to have listening tests.

Listening tests can be done at many different levels, ranging from initial (Diagnostic Rhyme Test) and/or final (Modified Rhyme Test) consonant comparison of similar sounding monosyllabic words to full sentence ratings. Furthermore, evaluators (or human test subjects) can match an utterance to a transcription, transcribe an utterance, or just type in the text to be synthesized, in which case they already know what to expect. The evaluators then rate the speech signal produced based on the given categories (e.g., naturalness, intelligibility, etc.) on a scale of 1 (worst) to 5 (best). The average of the resulting response values is calculated. The computed average is referred to as the *mean opinion score* (MOS).

For this project the evaluation process is twofold. First, test subjects listed to random speech renditions and matched them to their corresponding word sequence. In the second phase an open response identification (ORI) [3] method is used. With this method, evaluators listen to a synthetic speech rendition and write down (transcribe) or say what they think is the corresponding word sequence.

6.2 Evaluation Criteria

The recruitment process for evaluators or test subjects was structured as follows:

- Fluent Xitsonga speakers (not necessarily mother tongue speakers)
- Both male and female test subjects ranging between the ages 18 and 35
- The number of test subjects was sixteen: 8 – female and 8 – male
- People from a broad spectrum: undergraduate and postgraduate students, retail store sales personnel, security guards, etc.

Test sentence selection criteria:

- Sentences that do not form part of training set
- The number of test sentences was fixed to 14
- Sentences were selected from the general domain (normal day-to-day conversation)
- The test sentences included numbers, money or currency, dates, etc.

For this project, all evaluators were required to fill a consent form requesting their biographic information. Evaluators were made aware of the special nature of synthetic speech produced by the TTS systems so that they could appropriately adjust their expectations regarding the output from the system [3]. They then listened to utterances and mapped them to their corresponding transcriptions. This was done to ensure that they were, able to fairly judge the system as they would know what output to expect. Moreover, this also allows evaluators to comment on words not properly or wrongly synthesized.

The evaluation process is mainly based on the four concepts – naturalness, intelligibility, pleasantness, flexibility, and similarity to original speaker – which were discussed in chapter 2 (2.2.2). Naturalness has to do with the human-like sounding

of the system, whereas intelligibility/understandability has to do with the ability for one to hear the speech synthesized. Pleasantness – which is how enjoyable it is to listen to the synthesized speech is also evaluated. Flexibility has to do with how well tokenization is performed – tokenization is discussed in chapter 5 (5.1.5). Similarity to original speaker deals with the voice characteristics that are unique to the trained voice of the speaker. Lastly, the overall system performance impression is determined.

6.3 Evaluation Results and Analysis

The main TTS synthesis system without adaptation was trained using 453 sentences recorded by the author. Fourteen test sentences which did not form part of the training data were selected to test the main TTS synthesis system based on the MOS method. An additional five sentences which also did not form part of the training data were used for the ORI test of the main TTS synthesis system. Evaluators from diverse backgrounds and languages were used to test the system. Sixteen people, 8 males and 8 females, tested the system. The people that evaluated the system either spoke Xitsonga as their home language or could fairly hear and read the language. Thirteen evaluation subjects were Xitsonga speakers. Both Northern Sotho and Tshivenda speakers used as evaluation subjects were two and one respectively. A pool of five evaluation subjects from the 16 was used to perform the ORI test in addition to performing the general MOS test.

The MOS scores used in the evaluation process ranged from one (worst) to five (best). The evaluation categories used for the MOS method were: understandability, naturalness, pleasantness, flexibility, similarity to original speaker, and overall system impression. None of the subjects found the system hard to understand. The evaluations revealed that 43.75% of the people said that the system sounded like a human voice and the worst case being from one (6.25%) person saying that the system was not natural sounding. The evaluation revealed that 37.5% of the subjects found the system very pleasant to listen to and 6.25% of the subjects found the system horrible to listen to. The evaluation results also revealed that all subjects found the system flexible, with 44.75% saying the system was excellently flexibly,

12.5% saying that the system was acceptably flexible, and the rest saying that the system was flexibly good. Although the synthesized voice was found to sound close to that of the original speaker on average, only 12.5% of the subjects said that the voice sounded the same as that of the original speaker, with 18.75% of the subjects saying that the voice sounded nothing like the original speaker.

The evaluation results indicate that the overall system was found to be good on average. Only 25% of the subjects said that the overall system was excellent, 37.5% of the respondents said the system was good, and the other 37.5% said that the system was acceptable. The overall system, therefore, received a 100% acceptability rate. A summary of the MOS results is given in Table 6–3 as well as in Figure 6–1.

Question	MOS	Meaning
Understandability	3.9	Little effort required
Naturalness	3.9	Natural enough to listen to
Pleasantness	3.6	Quite pleasant
Flexibility	4.3	Good
Similarity to Original Speaker	3.0	Close
Overall	3.9	Good

Table 6–1. Tabular view of evaluation results

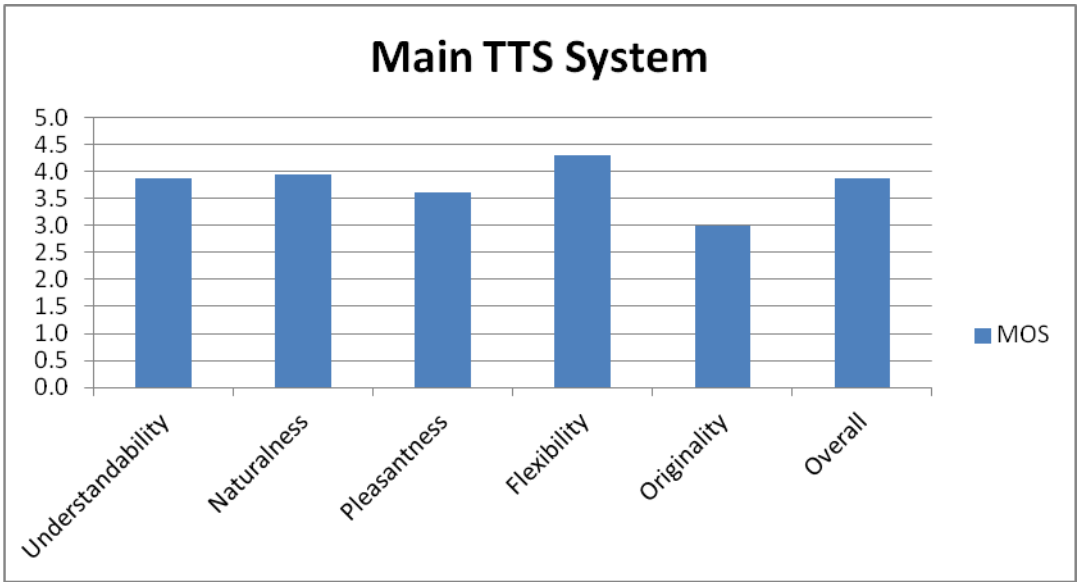

Figure 6–1. A diagrammatic view of the MOS of the main TTS system

A female voice was adapted to an extended version of the main HTS-demo by averaging the voice models of four other speakers used for training the system. Two of the 16 evaluators were also requested to evaluate the adapted voice. The evaluation results of the adapted voice are very closely related to those of the main TTS system as shown in Figure 6–2.

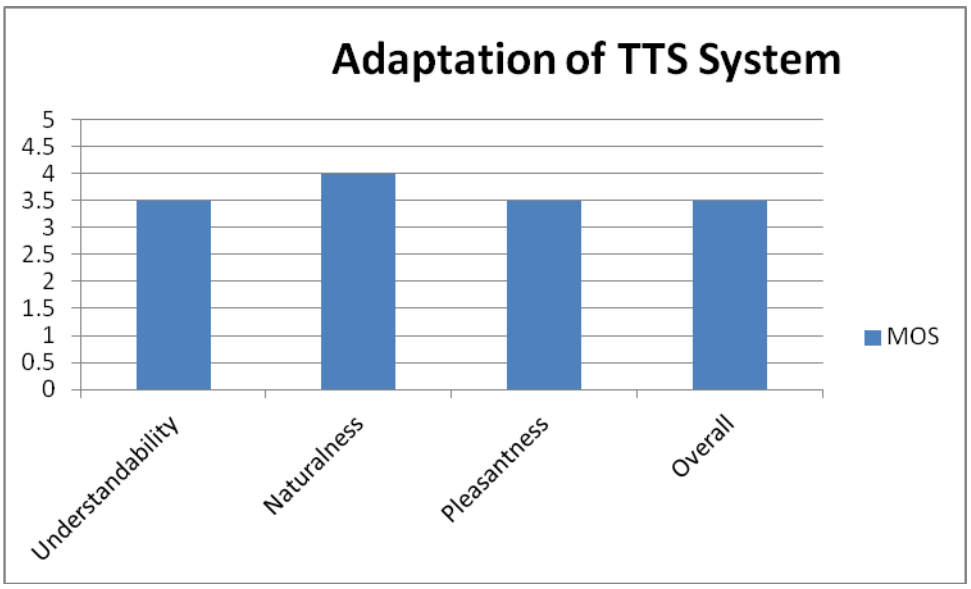

Figure 6–2. A diagramatic represantation of the MOS of the adapted speaker

In order to further verify the *intelligibility* of the system, an ORI evaluation was performed. For the ORI evaluation, a list of five sentences from a stored file was synthesized. A file was used in order to simulate a blind evaluation, whereby, users did not know the sentences that they would be tested on. Five subjects were used. It was found that the subjects were able to correctly transcribe synthesized sentences. Only a few misses (often one word per sentence misspelled or missed) were recorded, mainly due to the fact that most people often confuse spoken language with written language. This is because when people listen to a person talking, they usually strive to capture the meaning rather than the exact syntax of the spoken words.

Speech quality remains a concern. Several evaluators commented on the quality of the synthesized speech. A few others, even highlighted that voice clarity should be improved as they sometimes had to listen carefully to the synthesized speech in order to clearly hear the waveform rendition. Some of the comments were, of course,

due to evaluators' unfamiliarity with certain words. On the other hand, some of the comments particularly on attentiveness were due to the fact that the sentences were unrelated and as a result, there was no particular context with which evaluators could relate the sound they were hearing to. Speech quality and voice clarity are very important to TTS synthesis systems and it should be what every TTS synthesis system designer strives for at the very least. Although the voices of the two systems were found to be closely correlated, the loudness of the output or synthesised speech differed. The loudness of the adapted voice was found to be very low with an average of 37.66 decibels (dB) as can be seen in Figure 6–3. The loudness of the main TTS synthesis system was found to be fairly high in contrast to that of the adapted voice, with an average of 85.05 dB as can be seen in Figure 6–4. This difference, though significant, can be ignored as the loudness of a sound file can be increased (or modified) by using special speech processing tools like the Praat program.

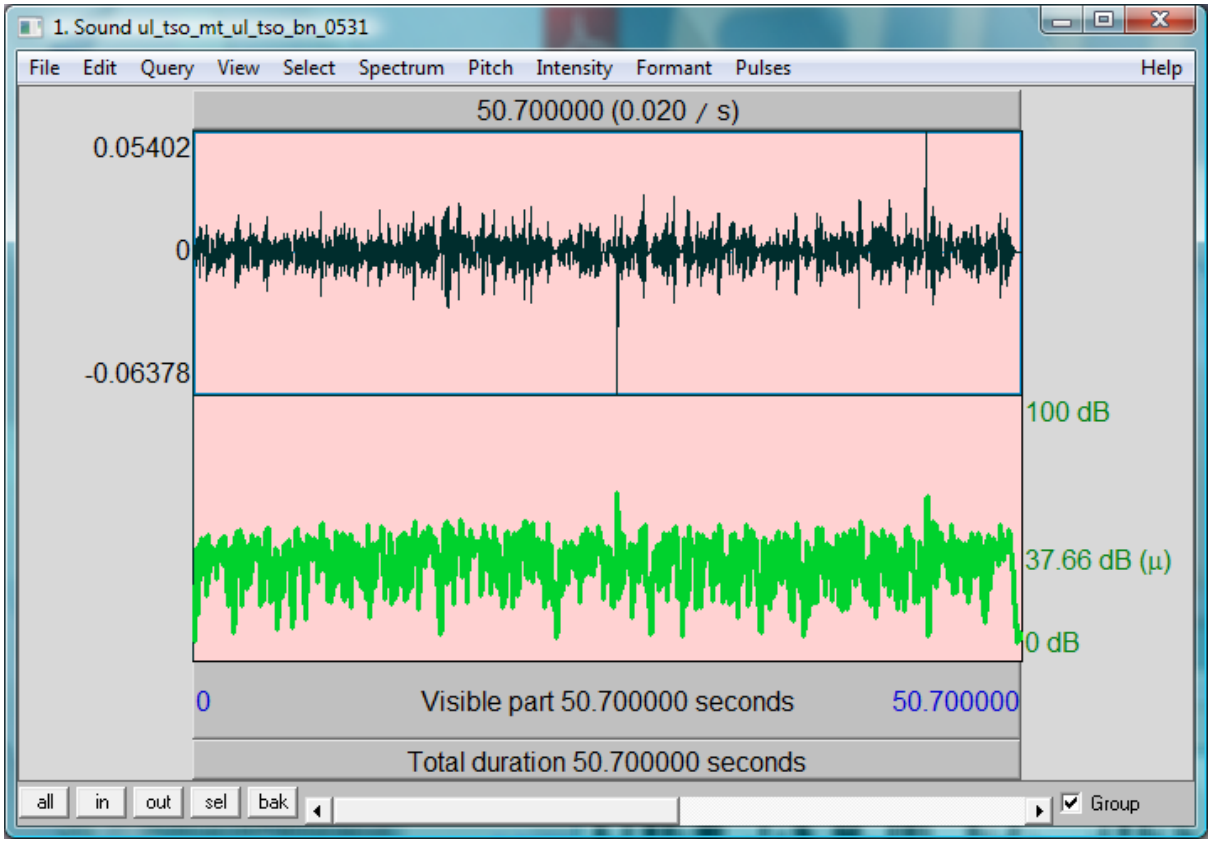

Figure 6–3. A representation of the loudness of a sound file in dBs for an adapted speaker

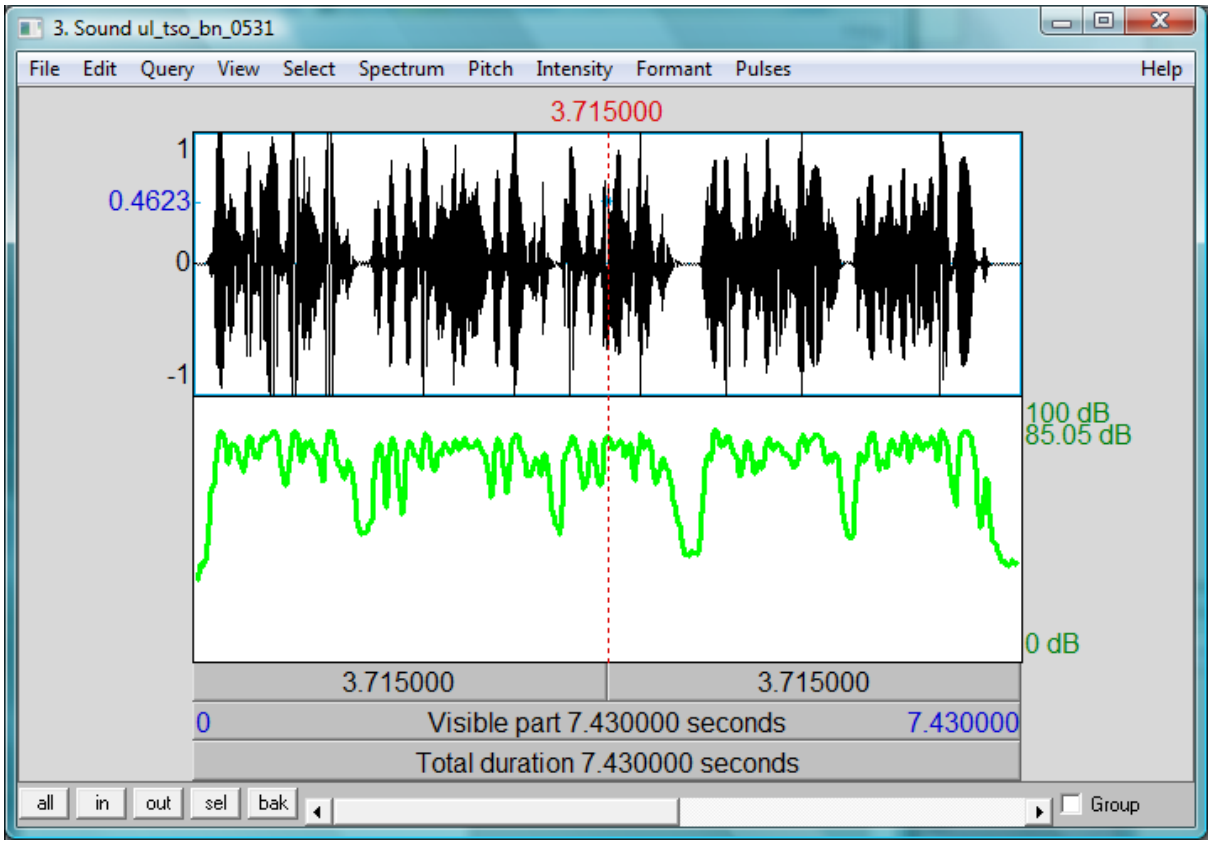

Figure 6–4. A representation of the loudness of a sound file in dBs output by the main TTS

6.4 Chapter Summary

This chapter discussed the evaluation method and criteria for selecting sentences and evaluators. The results of the evaluation were given and the analysis of the results outlined. It is this chapter that determines whether or not the project was a success. The analysis paves the way forward for changes that can be implemented to improve on the results.

Chapter 7

7 Conclusion and Future Work

This research report presented a baseline Xitsonga TTS synthesis system that synthesizes speech that is both intelligible and fairly natural sounding. The TTS synthesis system designed was extended to include an adaptation phase. The Xitsonga baseline speech synthesis system was initially trained using data from one speaker. Both the voices from the initial (or main) baseline and the one adapted to the final system were found to exhibit intelligibility, flexibility and fair naturalness. The voice synthesized by the main TTS synthesis system also proved to contain speaker specific voice characteristics. A hidden Markov model based approach was used for the development of the system. Although this method currently does not produce the best quality of speech as compared to some of the best unit selection based systems, it is a very flexible method that offers room for developing TTS synthesis systems with less corpus and data preparation challenges. The TTS synthesis system developed showed an ability to synthesize understandable speech though it had no associated pronunciation dictionary but only depended on grapheme-based rules and the phone set. Several methods were applied to the system in an attempt to better the quality of synthesized speech. The order of mel-cepstrum was increased to 40 and an attempt to use a vocoding method called STRAIGHT did not produce the expected results as it did not show a significant improvement to the results already obtained from the main (or normal) TTS synthesis system. The system received an overall MOS score of 3.9.

7.1 Conclusion

The evaluation results show that the system built is both intelligible and fairly natural sounding. These findings are in accordance with the expectations of the system. The evaluation results are approximately the same as those obtained from TTS synthesis systems recently developed in other institutions using the same method, although the strictness of the evaluation process may differ.

7.2 Proposed Future Improvements

The issues listed below are some amongst many that should be looked into towards developing an even better TTS synthesis system or improving on the/an already built TTS synthesis system:

- Good quality microphones are critical in order to achieve better quality speech sound [12]. Such microphones do not necessarily have to be state-of-the-art, but should be able to clearly and distinctly capture speech utterances. This requirement is not as stringent as it might appear as most hand held devices are already equipped with such microphones. Microphone effects on the recorded speech should be as minimal as possible in order to avoid having monotone or robotic speech.
- Speech recordings should be done from a laptop or other hand held devices for noise reduction [12]. There are different sources of background noise that affect recordings. A computer fan can be one of the significant noise sources that affect the quality of recorded speech when recording from a desktop computer, thereby affecting the quality of synthesized speech. Other noise sources that could be caused by the recorder (e.g., lip smacks) or the environment (e.g., machinery) under which the speech recording takes place need to be avoided. These can be avoided by educating recording subjects and using a specialized noise free room. Noise free speech is very important in speech synthesis contrary to speech recognition which sometimes might require speech with noise. Unlike in speech recognition, noise-free-speech in speech synthesis is a priority unless the TTS synthesis is to be trained using data collected for ASR or be used with an ASR.
- Using a much better labelling algorithm than EHMM labeler and/or using hand-labelling. Better algorithms are continuously being developed. It is, therefore, better to either design a better and more accurate labelling algorithm or use a state-of-the-art labelling algorithm.

- Including a Xitsonga language pronunciation dictionary can also significantly positively affect the quality of synthesized speech. The creation of such a dictionary will be of great benefit to the language itself as it is difficult to find a pronunciation dictionary for Xitsonga. This will in turn further play a role in adding to the pool of essential components or resources of South African indigenous spoken language systems and the language itself.
- Incorporating intonation into the system will also undoubtedly better the quality of synthesized speech as it has been proven in several research articles [1].
- The live demonstration interface of the Xitsonga TTS system should be improved to include more functionality beyond only being used for direct text and file input. An adaptation component together with additional input methods (email, web pages, etc.) should be included. The tokenizer file should also be expanded to cover more scope.
- Training the demo and adapting it to become a festival voice in order to achieve faster processing of speech output. At the moment the text processing component is the most time consuming part of the system. The effect is clearly seen when synthesizing a file with a considerable number of sentences.
- STRAIGHT vocoding could not be explored to a satisfactory level due to time constraints as the MATLAB software system was only obtained towards the end of the research project. STRAIGHT vocoding has, however, been proven to produce better results by several researchers that used the HTS toolkit. Further exploration of this method is therefore recommended towards achieving even better results.

Issues that should be looked at include but are not limited to those listed above. The method (HMM-based speech synthesis) used in this project may be younger than most other speech synthesis methods, but it offers great opportunities for future research as it promises to outperform many of the other speech synthesis methods.

References

[1] L.Mohasi, “Design of an Advanced and Fluent Sesotho Text-to-Speech System through Intonation”, Master’s thesis, University of Cape Town, May 2006

[2] T. Masuko, “HMM-Based speech synthesis and its applications”, PhD thesis, Tokyo Institute of Technology, November 2002, pp. 1-84

[3] X. Huang, A. Acero, and H. Hon, Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, 2001

[4] F. Rousseau, D.J. Mashao, “Increased Diphone Recognition for an Afrikaans TTS system”, Proceedings of PRASA., 2004, pp 113-117, Cape Town

[5] Department of Arts and Culture. (2006). Departmental Vision and Mission., [online]. Available: http://www.dac.gov.za/vision_mission.htm, last accessed: 11/06/2012

[6] Meraka Institute. (2010). Areas of research. Tshwane, CSIR., [online]. Available: http://www.meraka.org.za/lwazi/index.php, last accessed: 11/06/2012

[7] T. Raitio, “Hidden markov model based Finnish text-to-speech system utilizing glottal inverse filtering”, Masters Thesis, 2008

[8] History of Speech Synthesis. Wolfgang von Kempelen’s speaking machine and its successors., [online]. Available: http://www2.ling.su.se/staff/hartmut/kemplne.htm, last accessed: 11/06/2012

[9] S. Lammetty, Review of Speech Synthesis Technology, Master's Thesis, Helsinki University of Technology., [online]. Available: http://www.acoustics.hut.fi/publications/files/theses/lemmetty_mst/thesis.pdf, last accessed: 11/06/2012

[10] J.L. Flanagan, J.B. Allen, and M.A. Hasegawa-Johnson, Speech Analysis Synthesis and Perception, 3rd ed. (2008)

[11] J.A. Louw, M. Davel, and E. Barnard, “A general-purpose isiZulu speech synthesizer”, South African Journal of African Languages., 2005, 2

[12] S.T. Phihlela, “Text-to-Speech Synthesis in Northern Sotho”, unpublished, Master’s thesis, University of Limpopo, December 2005

[13] P. Scholtz, A. Visagie, and J. du Prez, “Statistical Speech Synthesis for the Blizzard Challenge 2008”, Stellenbosch University, 2008

[14] Meraka Institute. (2010). Publications. Tshwane, CSIR., [online]. Available: http://www.meraka.org.za/lwazi/publications/louw08speectmultilingual.pdf, last accessed: 11/06/2012

[15] S. King, The Edinburg Speech Tools Library. The Center for Speech Technology Research., [online] Available: http://www.cstr.ed.ac.uk/projects/speech_tools/, last accessed: 11/06/2012

[16] A.W. Black, festvox. University of Edinburgh Festival Speech Synthesis System, November 2010, [online]. Available: http://festvox.org/festival/, last accessed: 11/06/2012

[17] Census 2001: Census in brief. Pretoria: Statistics South Africa, 2003, [online]. Available: http://www.statssa.gov.za/census01/html/CInBrief/CIB2001.pdf, last accessed: 11/06/2012

[18] Statistical release: Mid-year population estimates 2011. Pretoria: Statistics South Africa, July 2011. [online]. Available: http://www.statssa.gov.za/publications/statsdownload.asp?PPN=P0302&SCH=4986, last accessed: 11/06/2012

[19] D. Joffe, African Languages, [online]. Available: http://africanlanguages.com/tsonga/#map, last accessed: 11/06/2012

[20] I. Hone, “The History of the Development of Tsonga Orthography”, Honours Thesis, University of South Africa, January 1981, pp. 1-37

[21] C.T.D. Marivate, TSONGA: Only study guide for TSG301-K, University of South Africa, Pretoria, 1991

[22] MILAWU YA MAPELETELO NA MATSALELO YA XITSONGA. PANSALB. 2008, [online]. Available: http://www.language-inc.org/index.php?option=com_joomdoc&task=doc_download&gid=17&Itemid=173, last accessed: 11/06/2012

[23] C.P.N. Nkondo, Xiletelo xa Xitsonga, 2nd Ed., 1986

[24] A.S. Hornby, Oxford Advanced Learner's Dictionary of Current English. Oxford University Press, 7th Ed,, 2005

[25] K. Tokuda, H. Zen, and A.W. Black, “An HMM-based speech synthesis system applied to English”. Proc. of 2002 IEEE Workshop on Speech Synthesis, pp. 227–230, Sep. 2002

[26] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, “Acoustic modeling of speaking styles and emotional expressions in HMM-based speech synthesis,” IEICE Trans. Inf. & Syst., Vol. E88-D, No. 3, pp. 503–509, March 2005

[27] H. Zen, K. Oura, T. Nose, J. Yamagishi, S. Sako, T. Toda, T. Masuko, A.W. Black, and K. Tokuda, “Recent development of the HMM-based speech synthesis system (HTS)”. In Proc. 2009 Asia-Pacific Signal and Information Processing Association (APSIPA), Sapporo, Japan, Oct 2009

[28] HTK, [online]. Available: http://htk.eng.ac.uk, last accessed: 11/06/2012

[29] HTK, [online]. Available: http://htk.eng.ac.uk/extensions/index.shtml, last accessed: 11/06/2012

[30] K. Oura, HMM-based Speech Synthesis System (HTS), [online]. Available: http://hts.sp.nitech.ac.jp/, last accessed: 11/06/2012

[31] A. Tamamori, S. Shinji, K. Tokuda and K. Oura, Sourcefoge, Speech Signal Processing Toolkit (SPTK), [online]. Available: http://sourceforge.net/projects/sp-tk/files/SPTK/, last accessed: 11/06/2012

[32] HTS, hts_engine API, [online]. Available: http://hts-engine.sourceforge.net/, last accessed: 11/06/2012

[33] ActiveState, ActiveState Downloads, May 2010, [online]. Available: http://downloads.activestate.com/ActiveTcl/releases/, last accessed: 04/05/2011

[34] STRAIGHT trial page, Jan 2009, [online]. Available: http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTtrial/, last accessed: 11/06/2012

[35] MathWorks, [online]. Available: http://mathworks.com/products/matlab/, last accessed: 11/06/2012

[36] C. Bagwell, Sourceforge, Sox-Sound eXchange, 2011, [online]. Available: http://sourceforge.net/projects/sox/files/sox/14.3.1/, last accessed: 11/06/2012

[37] A.L. Black, festvox, Copying, Jan 2007, [online]. Available: http://festvox.org/download.html, last accessed: 11/06/2012

[38] P. Boersma and D. Weenink. Praat: Doing Phonetics by Computer. [online]. Available: http://www.fon.hum.uva.nl/praat/, last accessed: 11/06/2012

[39] J. Beskow and K. Sjolander, Sourceforge, [online]. Available: http://sourceforge.net/projects/wavesurfer/, last accessed: 11/06/2012

[40] A.W. Black and K.A. Lenzo, Building Synthetic Voices, Statistical Parametric Synthesis, Building a CLUSTERGEN Statistical Parametric Synthesizer, Jan

2007, [online]. Available: http://festvox.org/bsv/book1.html, last accessed: 11/06/2012

[41] N. Baloyi, M.J.D. Manamela, and N. Gasela, A Text-to-Speech Synthesis System Using Hidden Markov Models for Xitsonga, Sep 2011, [online]. Available: http://www.satnac.org.za, last accessed: 11/06/2012

[42] R.S. Pressman, Software: A practitioner's approach, McGraw-Hill International Edition, 7th Ed., 2010

[43] Wayne's Blog, [online]. Available: http://zwe0405.blogspot.com/2010/09/ubuntu-1004-hts-341-hts-211.html, last accessed: 11/06/2012

Appendices

Appendix A – txt.done.data

Appendix B – Initial and Final Labels and Utterances

Appendix C – Question Set (questions_qst001.hed)

Appendix D –Phone Set Radio (radio_phones.scm)

Appendix E – Tokenizer (ul_tso_bn_tokenizer.scm)

Appendix F – Installation

Appendix G – SATNAC Complete Paper 2011

Appendix H – SATNAC Work in Progress Paper 2010

Appendix I – SATNAC Work in Progress Poster 2010

Appendix J – FSA-PG Paper 2011

Appendix A – txt.done.data

```
(ul_tso_bn_0001 "ndzi dya matandza")
(ul_tso_bn_0002 "tiyani u dya matimba")
(ul_tso_bn_0003 "mbhongolo ya kokwana yi ni matimba")
(ul_tso_bn_0004 "mufana u senga homu")
(ul_tso_bn_0005 "xikoxa xi nwa mati")
(ul_tso_bn_0006 "i goji ro enta")
(ul_tso_bn_0007 "a vona mali a nga tikhomi")
(ul_tso_bn_0008 "va kale va ku nwela na kofi")
(ul_tso_bn_0009 "lava ngoti ya ku boha tihunyi")
(ul_tso_bn_0010 "nkwaa!")
(ul_tso_bn_0011 "mbhoo!")
(ul_tso_bn_0012 "i ncini?")
(ul_tso_bn_0013 "munhu a nga kala a ku rivala mani")
(ul_tso_bn_0014 "wa ha humile")
(ul_tso_bn_0015 "va siye bege ebazini")
(ul_tso_bn_0016 "ahanti!")
(ul_tso_bn_0017 "chava ku tirha na tata wa wena")
(ul_tso_bn_0018 "u ta vuya a dya")
(ul_tso_bn_0019 "nkambani wa")
(ul_tso_bn_0020 "leswi a nga dyela embiteni ke")
(ul_tso_bn_0021 "byi ri vudyelela bya yini")
(ul_tso_bn_0022 "mudyeleriwa wa kona hi yena gezani")
(ul_tso_bn_0023 "va dyeteterile hinkwaswo")
(ul_tso_bn_0024 "xikomu xa n'wina xi tika ngopfu")
(ul_tso_bn_0025 "va ha vhakacherile n'anga ya meno")
(ul_tso_bn_0026 "ku lo dzwii!")
(ul_tso_bn_0027 "wa ha ku heta ku kukula yindlu yakwe")
(ul_tso_bn_0028 "murisi wa yi ba homu")
....
(ul_tso_bn_0525 "u vulavula xinghexi, xivhenda, xisuthu na xixhoza")
(ul_tso_bn_0526 "ku enghenela mijekajekisano eka tindhawu to
hambanahambana")
(ul_tso_bn_0527 "switshunxo na swiboho")
(ul_tso_bn_0528 "nhluvukiso wa ririmi ra xitsonga")
(ul_tso_bn_0529 "minongoti leyintshwa leyi tumbulukaka")
(ul_tso_bn_0530 "tima timboni ta le handle")
(ul_tso_bn_0531 "wa nga ndzi nyiketa namuneti kwalaho xigwitsirisini")
(ul_tso_bn_0532 "madyelo ya rihlaza")
(ul_tso_bn_0533 "byetlelo bya xiyimo xa le henha")
(ul_tso_bn_0534 "ku duma ka bazi")
(ul_tso_bn_0535 "ndzi ta khandziya thekisi yo ya ethohoyandou")
(ul_tso_bn_0536 "xinkhubyana xa n'wana")
(ul_tso_bn_0537 "mukhuhlwani wu horile")
(ul_tso_bn_0538 "ximfukwana xa kokwani")
(ul_tso_bn_0539 "xibukwana xa sosayiti")
(ul_tso_bn_0540 "nwana masi")
(ul_tso_bn_0541 "u kumule makumemune nhlanu wa tiphesente")
(ul_tso_bn_0542 "va lava madzananharhu makumemunentlhanu wa tirhandi na
makumenharhunharhu wa tisente ya rendzodyondzo")
(ul_tso_bn_0543 "ku sarile xiphemu xa n'we xa nhungu")
(ul_tso_bn_0544 "ndzi ta mi nyika kwalomuya ka ntlhanu ku fika ka nhungu wa
makwembe")
(ul_tso_bn_0545 "hi ta hlangana hi siku ra mbirhi n'wendzamhala
magidimbirhi na khume n'we")
```

Appendix B – Initial and Final Labels and Utterances

Initial label	**Initial utterance**
# 0.2750 100 pau 0.3850 100 n 0.4950 100 d 0.6050 100 z 0.7150 100 i 0.8250 100 d 0.9350 100 y 1.0450 100 a 1.1550 100 m 1.2650 100 a 1.3750 100 t 1.4850 100 a 1.6500 100 n 1.8150 100 d 1.9800 100 z 2.1450 100 a 2.4200 100 pau	EST_File utterance DataType ascii version 2 EST_Header_End Features max_id 53 ; type Text ; iform "\"ndzi dya matandza\"" ; Stream_Items 1 id _1 ; name ndzi ; whitespace "" ; prepunctuation "" ; 2 id _2 ; name dya ; whitespace " " ; prepunctuation "" ; 3 id _3 ; name matandza ; whitespace " " ; prepunctuation "" ; 4 id _6 ; name matandza ; pbreak B ; pos nil ; 5 id _5 ; name dya ; pbreak NB ; pos nil ; 6 id _4 ; name ndzi ; pbreak NB ; pos nil ; 7 id _7 ; name B ; 8 id _8 ; name syl ; stress 0 ; 9 id _13 ; name syl ; stress 0 ; 10 id _17 ; name syl ; stress 0 ; 11 id _20 ; name syl ; stress 0 ; 12 id _23 ; name syl ; stress 0 ; 13 id _28 ; name pau ; dur_factor 1 ; end 0.275 ; source_end 0.050581 ; 14 id _9 ; name n ; dur_factor 1 ; end 0.385 ; source_end 0.20417 ; 15 id _10 ; name d ; dur_factor 1 ; end 0.495 ; source_end 0.314607 ; 16 id _11 ; name z ; dur_factor 1 ; end 0.605 ; source_end 0.420998 ; 17 id _12 ; name i ; dur_factor 1 ; end 0.715 ; source_end 0.52206 ; 18 id _14 ; name d ; dur_factor 1 ; end 0.825 ; source_end 0.651934 ; . . . . End_of_Relation Relation US_map ; () 1 52 0 0 0 0 End_of_Relation Relation Wave ; () 1 53 0 0 0 0 End_of_Relation End_of_Relations End_of_Utterance
Final (or context dependent) label	**Final utterance**

0 2750000 x^x-pau+n=d@x_x/A:0_0_0/ B:x-x-x@x-x&x-x#x-x$x-x!x-x;x- x|x/C:0+0+4/D:0_0/E:x+x@x+x&x+x#x+x/ F:content_1/G:0_0/H:x=x@1=1|0/ I:5=3/J:5+3-1 2750000 3850000 x^pau-n+d=z@1_4/ A:0_0_0/ B:0-0-4@1-1&1-5#1-1$1-1!0- 0;00\|i/C:0+0+3/ D:0_0/E:content+1@1+3&1+2#0+1/ F:content_1/G:0_0/H:5=3@1=1\|NONE/ I:0=0/J:5+3-1 . . . . 21450000 24200001 z^a-pau+x=x@x_x/ A:0_0_4/B:x-x-x@x-x&x-x#x-x$x-x!x-x;x- x|x/C:0+0+0/D:content_3/ E:x+x@x+x&x+x#x+x/F:0_0/G:5_3/ H:x=x@1=1|0/I:0=0/J:5+3-1	EST_File utterance DataType ascii version 2 EST_Header_End Features max_id 71 ; type Text ; iform "\"ndzi dya matandza\"" ; filename prompt-utt/ul_tso_bn_0001.utt ; fileid ul_tso_bn_0001 ; Stream_Items 1 id _1 ; name ndzi ; whitespace "" ; prepunctuation "" ; 2 id _2 ; name dya ; whitespace " " ; prepunctuation "" ; 3 id _3 ; name matandza ; whitespace " " ; prepunctuation "" ; 4 id _6 ; name matandza ; pbreak B ; pos nil ; 5 id _5 ; name dya ; pbreak NB ; pos nil ; 6 id _4 ; name ndzi ; pbreak NB ; pos nil ; 7 id _7 ; name B ; 8 id _8 ; name syl ; stress 0 ; 9 id _13 ; name syl ; stress 0 ; 10 id _17 ; name syl ; stress 0 ; 11 id _20 ; name syl ; stress 0 ; 12 id _23 ; name syl ; stress 0 ; 13 id _28 ; name pau ; dur_factor 1 ; end 0.16 ; source_end 0.050581 ; 14 id _9 ; name n ; dur_factor 1 ; end 0.215 ; source_end 0.20417 ; 15 id _10 ; name d ; dur_factor 1 ; end 0.265 ; source_end 0.314607 ; 16 id _11 ; name z ; dur_factor 1 ; end 0.3 ; source_end 0.420998 ; 17 id _12 ; name i ; dur_factor 1 ; end 0.375 ; source_end 0.52206 ; 18 id _14 ; name d ; dur_factor 1 ; end 0.48 ; source_end 0.651934 ; . . . . 1 51 0 0 2 0 2 52 0 0 0 1 End_of_Relation Relation US_map ; "(" ")" ; 1 53 0 0 0 0 End_of_Relation Relation Wave ; "(" ")" ; 1 54 0 0 0 0 End_of_Relation End_of_Relations End_of_Utterance

Appendix C – Question Set (questions_qst001.hed)

```
QS "LL-Voiced"    {a^*, e^*, i^*, o^*, u^*, b^*, by^*, d^*, dy^*, dh^*,
dz^*, dzw^*, dzh^*, g^*, gw^*, gh^*, h^*, j^*, l^*, lw^*, m^*, mh^*, n^*,
nd^*, ndl^*, ng^*, nh^*, nkh^*, nw^*, ny^*, n'^*, n'w^*, q^*, r^*, rhw^*,
v^*, vh^*, w^*, y^*, z^*}
QS "LL-Unvoiced"  {c^*, ch^*, f^*, hl^*, hlw^*, hw^*, k^*, kh^*, kw^*,
kwh^*, p^*, ph^*, phy^*, py^*, s^*, sw^*, t^*, th^*, thy^*, tl^*, tlh^*,
tshw^*, ty^*, x^*, xw^*}
// QS "LL-Murmered"     {bh^*, dh^*, gh^*, dzh^*, jh^*, mh^*, nh^*, nwh^*}
QS "LL-Silence"    {sil^*, pau^*}

QS "LL-Vowel"           {a^*, e^*, i^*, o^*, u^*}
QS "LL-Voiced_Vowel"          {a^*, e^*, i^*, o^*, u^*}
QS "LL-Front_Vowel"           {i^*}
QS "LL-Central_Vowel"         {e^*, o^*}
QS "LL-Back_Vowel"            {a^*, u^*}
QS "LL-High_Vowel"            {e^*, i^*, o^*, u^*}
QS "LL-Low_Vowel"       {a^*}
QS "LL-Rounded_Vowel"         {o^*, u^*}
QS "LL-Unrounded_Vowel"       {i^*, e^*,a^*}

QS "LL-Vowel_I"         {i^*}
QS "LL-Vowel_E"         {e^*}
QS "LL-Vowel_A"         {a^*}
QS "LL-Vowel_O"         {o^*}
QS "LL-Vowel_U"         {u^*}

QS "LL-Voiced_Consonant"       {b^*, d^*, dz^*, g^*, h^*, j^*, l^*, m^*,
n^*, ny^*, n'^*, q^*, r^*, v^*, vh^*, w^*, y^*, z^*}
QS "LL-Unvoiced_Consonant"    {c^*, f^*, hl^*, k^*, p^*, py^*, s^*, sw^*,
t^*, tl^*, tlh^*, tshw^*, x^*}

QS "LL-Alveolar"        {d^*, dl^*, dz^*, hl^*, l^*, n^*, t^*, r^*, s^*,
sw^*, tl^*, ts^*, z^*}
QS "LL-Bilabial"        {b^*, m^*, p^*, py^*}
QS "LL-Labial"          {dz^*, k^*, kw^*, l^*, s^*, sw^*, t^*, y^*}
QS "LL-Labiodental"     {bv^*, f^*, pf^*, v^*, vh^*}
QS "LL-Nasal"     {m^*, n^*, ny^*, n'^*}
QS "LL-Palatal"   {ny^*, py^*, q^*, y^*}
QS "LL-Velar"           {g^*, j^*, k^*, n'^*}
QS "LL-Fricative"       {f^*, l^*, s^*, sw^*, v^*, vh^*, x^*, z^*, zw^*}
QS "LL-Affricative"     {bv^*, c^*, dl^*, j^*, pf^*, ts^*}
QS "LL-Implosive"       {b^*, c^*, gq^*, k^*, p^*, q^*, t^*, tl^*,ts^*}
QS "LL-Explosive"       {b^*, d^*, dl^*, g^*, k^*, p^*, r^*, t^*, tl^*}
QS "LL-Aspirated"       {c^*, ch^*, dhl^*, h^*, hl^*, k^*, kh^*, t^*, th^*,
xj^*}

QS "LL-Voiced_Alveolar_Affricate"   {dl^*, dz^*, c^*}
QS "LL-Voiced_Alveolar_Fricate"     {dz^*, l^*, ts^*, z^*}
QS "LL-Voiced_Labiodental_Fricate"        {v^*, vh^*}
QS "LL-Unvoiced_Aleveolar_Fricate"        {s^*, hl^*}

QS "LL-a"         {a^*}
QS "LL-b"         {b^*}
QS "LL-bv"        {bv^*}
QS "LL-c"         {c^*}
QS "LL-d"         {d^*}
QS "LL-dhl"       {dhl^*}
```

```
QS "LL-dl"        {dl^*}
QS "LL-dz"        {dz^*}
QS "LL-e"         {e^*}
QS "LL-f"         {f^*}
QS "LL-g"         {g^*}
QS "LL-gq"        {gq^*}
QS "LL-h"         {h^*}
QS "LL-hl"        {hl^*}
QS "LL-i"         {i^*}
QS "LL-j"         {j^*}
QS "LL-k"         {k^*}
QS "LL-l"         {l^*}
QS "LL-m"         {m^*}
QS "LL-n"         {n^*}
QS "LL-ny"        {ny^*}
QS "LL-n'"        {n'^*}
QS "LL-o"         {o^*}
QS "LL-p"         {p^*}
QS "LL-pf"        {pf^*}
QS "LL-py"        {py^*}
QS "LL-q"         {q^*}
QS "LL-r"         {r^*}
QS "LL-s"         {s^*}
QS "LL-sw"        {sw^*}
QS "LL-t"         {t^*}
QS "LL-tl"        {tl^*}
QS "LL-ts"        {ts^*}
QS "LL-u"         {u^*}
QS "LL-v"         {v^*}
QS "LL-vh"        {vh^*}
QS "LL-w"         {w^*}
QS "LL-x"         {x^*}
QS "LL-xj"        {xj^*}
QS "LL-y"         {y^*}
QS "LL-z"         {z^*}


QS "L-Voiced"     {*^a-*, *^e-*, *^i-*, *^o-*, *^u-*, *^b-*, *^by-*, *^d-*,
*^dy-*, *^dh-*, *^dz-*, *^dzw-*, *^dzh-*, *^g-*, *^gw-*, *^gh-*, *^h-*,
*^j-*, *^l-*, *^lw-*, *^m-*, *^mh-*, *^n-*, *^nd-*, *^ndl-*, *^ng-*, *^nh-
*, *^nkh-*, *^nw-*, *^ny-*, *^n'-*, *^n'w-*, *^q-*, *^r-*, *^rhw-*, *^v-*,
*^vh-*, *^w-*, *^y-*, *^z-*}
QS "L-Unvoiced"   {*^c-*, *^ch-*, *^f-*, *^hl-*, *^hlw-*, *^hw-*, *^k-*,
*^kh-*, *^kw-*, *^kwh-*, *^p-*, *^ph-*, *^phy-*, *^py-*, *^s-*, *^sw-*,
*^t-*, *^th-*, *^thy-*, *^tl-*, *^tlh-*, *^tshw-*, *^ty-*, *^x-*, *^xw-*}
// QS "L-Murmered"      {*^bh-*, *^dh-*, *^gh-*, *^dzh-*, *^jh-*, *^mh-*,
*^nh-*, *^nwh-*}
QS "L-Silence"     {*^sil-*, *^pau-*}
```

Appendix D – Phone Set Radio (radio_phones.scm)

```
(defPhoneSet
  radio
  ;;;  Phone Features
  (;; vowel or consonant
   (vc + -)
   ;; vowel length: short long dipthong schwa
   (vlng s l d a 0)
   ;; vowel height: high mid low
   (vheight 1 2 3 0)
   ;; vowel frontness: front mid back
   (vfront 1 2 3 0)
   ;; lip rounding
   (vrnd + - 0)
   ;; consonant type: stop fricative affricate nasal lateral approximant
   (ctype s f a n l r 0)
   ;; place of articulation: labial alveolar palatal labio-dental
   ;;                        dental velar glottal
   (cplace l a p b d v g 0)
   ;; consonant voicing
   (cvox + - 0)
   )
  ;; Phone set members
  (
   (pau  - 0 0 0 - 0 0 -)  ;; slience ...
    ;;vc    vlng  vheit vfrnt vrnd  ctype cplac cvoice
   (a    +   a   3   3   -   0   0   +)
   (e    +   a   3   3   -   0   0   +)
   (i    +   a   1   1   -   0   0   +)
   (o    +   a   3   3   -   0   0   +)
   (u    +   a   3   3   -   0   0   +)
   (bv   -   0   0   0   0   a   b   0)
   (b    -   0   0   0   0   0   l   +)
   (c    -   0   0   0   0   a   0   -)
   (dhl  -   0   0   0   0   a   0   0)
   (dl   -   0   0   0   0   a   a   0)
   (dz   -   0   0   0   0   a   a   +)
   (dz   -   0   0   0   0   a   a   +)
   (d    -   0   0   0   0   0   a   +)
   (f    -   0   0   0   0   f   b   -)
   (gq   -   0   0   0   0   0   0   0)
   (g    -   0   0   0   0   0   v   +)
   (hl   -   0   0   0   0   a   a   -)
   (h    -   0   0   0   0   a   0   +)
   (j    -   0   0   0   0   a   v   +)
   (k    -   0   0   0   0   a   v   -)
   (l    -   0   0   0   0   f   a   +)
   (m    -   0   0   0   0   n   l   +)
   (ny   -   0   0   0   0   n   d   0)

...

  )
)

(PhoneSet.silences '(pau h# brth))

(provide 'radio_phones)
```

Appendix E – Tokenizer (ul_tso_bn_tokenizer.scm)

```
;;; Load any other required files
(set! ul_tso_digit_names
 '((0 "tandza")
   (1 "n#we")
   (2 "mbirhi")
   (3 "nharhu")
   (4 "mune")
   (5 "ntlhanu")
   (6 "ntsevu")
   (7 "nkombo")
   (8 "nhungu")
   (9 "nkaye"))
)

;; Punctuation for the particular language
(set! ul_tso_bn::token.punctuation "\"'`.,:;!?(){}[]")
(set! ul_tso_bn::token.prepunctuation "\"'`({[")
(set! ul_tso_bn::token.whitespace " \t\n\r")
(set! ul_tso_bn::token.singlecharsymbols "")

;;; Voice/tso token_to_word rules
(define (ul_tso_bn::token_to_words token name)
  "(ul_tso_bn::token_to_words token name)
Specific token to word rules for the voice ul_tso_bn.  Returns a list
of words that expand given token with name."
  (cond
    ((string-matches name "10")
    (list "khume" ))

 ((string-matches name "1[0-9]")
    (append
     (list "khume" )
      (ul_tso_bn::token_to_words token (string-after name "1"))))

((string-matches name "20")
    (list "makumembirhi" ))

 ((string-matches name "2[0-9]")
    (append
     (list "makumembirhi" )
      (ul_tso_bn::token_to_words token (string-after name "2"))))

((string-matches name "30")
    (list "makumenharhu" ))

 ((string-matches name "3[0-9]")
    (append
     (list "makumenharhu" )
      (ul_tso_bn::token_to_words token (string-after name "3"))))

((string-matches name "40")
    (list "makumemune" ))


...
```

Appendix F – Installation

Details relating to the installation of all packages can also be found within the packages themselves, often in a file named INSTALL. Additional information can be found on the internet. The installations provided below were mainly guided by the instructions in Wayne's Blog [43].

sudo aptitude update
sudo aptitude install build-essential
sudo apt-get install gfortran
$ sudo apt-get install build-essential
$ sudo apt-get install libx11-dev
$ sudo apt-get install libncurses5-dev
$ sudo apt-get install libncursesw5-dev
$ sudo apt-get install sox (or follow the download and install procedure)

Speech Tools

$tar zvxf speech_tools-2.1-release.tar.gz
$cd speech_tools
speech_tools$make
speech_tools$sudo make install

Festival

$tar zvxf festival-2.1-release.tar.gz
tar xvf festvox_kallpc16k.tar.gz
tar xvf festlex_POSLEX.tar.gz
tar festlex_POSLEX.tar.gz
tar xvf festvox_cmu_us_slt_arctic_hts.tar.gz
$cd festival
festival$./configure
festival$make
festival$sudo make install

HTK, HDecode and HTS

```
$ tar zvxf HTK-3.4.1.tar.gz
$ tar vxf HDecode-3.4.1.tar.gz
$ tar -xf HTS-2.2_for_HTK-3.4.1.tar.tar -C ./htk
$ cd htk
htk$ patch -p1 -d . < HTS-2.2_for_HTK-3.4.1.patch
htk$ ./configure
htk$ make all
htk$ sudo make install
```

hts_engine

```
$ tar zvxf hts_engine_API-1.06.tar.gz
$ cd hts_engine_hts_engine_API-1.06
hts_engine_API-1.06$ ./configure
hts_engine_API-1.06$ make
hts_engine_API-1.06$ sudo make install
```

SPTK

```
$ tar zvxf SPTK-3.4.1.tar.gz
$ cd SPTK-3.4.1
SPTK-3.4.1$ ./configure
SPTK-3.4.1$ make
SPTK-3.4.1$ sudo make install
```

ActiveTcl

```
$ tar zvxf ActiveTcl8.4.19.4.292682-linux-ix86.tar.gz
$ cd ActiveTcl8.4.19.4.292682-linux-ix86
ActiveTcl8.4.19.4.292682-linux-ix86$ sudo ./install.sh
```

Festvox

```
$ tar zvxf festvox-2.1-release.tar.gz
$ cd festvox
festvox$ ./configure
festvox$ make
```

A Text-to-Speech Synthesis System Using hidden Markov models for Xitsonga

Ntsako Baloyi*, MJD Manamela, N Gasela
Department of Computer Science
University of Limpopo (Turfloop Campus),
Private Bag X1106,
Sovenga, 0727
Tel: +27 15 2682751, Fax: +27 15 2683183
email: {200522530, jonas.manamela, nalson.gasela}@ul.ac.za; nbaloyi11@gmail.com

Abstract- From several current spoken language processing research reports it has been shown that the development of a text-to-speech synthesis system using the HTS toolkit based on hidden Markov models can be achieved without requiring a huge training speech corpus. Intelligible and natural sounding speech is therefore achievable. The quality of synthesized speech, however, does not equal that of recorded speech. Some speaker characteristics and speaking styles are modeled by trained hidden Markov models from context-dependent labels. Xitsonga is a resource-scarce language spoken in more than three Southern African countries. The speech synthesis system for Xitsonga based on hidden Markov model discussed in this paper focuses on Xitsonga language as it is spoken in the Republic of South Africa. The preliminary results obtained after training the text-to-speech synthesis system show that the developed system is intelligible and it also obtains some level of naturalness. The mean opinion score tests attain 92.3% of acceptability from first language speakers.

Index Terms— text-to-speech, speech synthesis, Xitsonga, spoken language processing

I. INTRODUCTION

A text-to-speech (TTS) synthesis system for a particular natural language is a technology for translating or converting a given typed or stored text input into its equivalent spoken format. The focus of building the Xitsonga baseline TTS synthesis system has been on producing flexible, intelligible and natural sounding speech. The 2001 Republic of South Africa (RSA) census revealed that over 4.5 million South African citizens were completely illiterate, and that the number for those without a Grade 12 certificate exceeded 18 million people [1].

The process of developing a text-to-speech system using one of the indigenous languages of South Africa is a practical step towards making both information and technology accessible and easy to use by all individuals at different literacy levels. The 2001 census results also revealed that by that year Xitsonga was spoken by 4.44% of South African citizens; this indicates that about two million of the approximately forty five million citizens spoke Xitsonga as their first language [1]. With an increase of close to 6 million people from 2001 to July 2011 [2], Xitsonga speakers are most likely above 2 million in number.

The choice of speech synthesis method based on a hidden Markov model (HMM), called HTS, over other speech synthesis methods was inspired by its ability to synthesize intelligible and natural sounding speech without requiring a huge training corpus [3, 4]. This method achieves this task by statistically modeling speech parameters using HMMs. Furthermore, the run time synthesis engine of HTS – the toolkit used for HMM-based speech synthesis – can be about 2 to 25 Megabytes excluding the text analysis component. Low memory requirements and flexibility of HTS are some of the factors that favoured the choice of this method of speech synthesis. It therefore becomes easy to implement a system built using HTS on handheld devices. TTS systems can be used as message readers, learning assistants, tools to aid in communication and learning for the handicapped and visually challenged people. Meraka Institutes' human language technology division developed a multilingual system called *speect* using the eleven South African official languages [5].

Although the speech quality of statistical parametric speech synthesizers may be lower than that of state-of-the-art unit selection systems, HMM-based speech synthesis systems offer much flexibility and ease of capturing statistical properties of speakers. With HMM-based speech synthesis systems it is easy to model various speaker characteristics and speaking styles. The context-dependent HMMs in such a system are the equivalent of phoneme-sized units.

The remainder of this paper is outlined hereafter. Section II discusses the use of the HMM-based speech synthesis system toolkit. This section starts by discussing software requirements in A, data preparation in B, the training phase and synthesis phase in C and D respectively. Section III looks into some challenges that were encountered during the development of the system, whereas, Sections IV, V, VI, and VII deal with results and analysis, future work, conclusion and acknowledgements respectively.

II. HMM-BASED SPEECH SYNTHESIS SYSTEM (HTS)

The HMM-based speech synthesis system (HTS) is a toolkit that is designed to be patched to the Hidden Markov Model Toolkit (HTK). HTK is a toolkit that is primarily used in speech recognition research for building and manipulating hidden Markov models. It is released under a

free license and requires that one also obeys the license of HTK to which it is patched.

A. Software Tools

The HTS toolkit is used for implementing HMM-based speech synthesis. HTS-2.1.1 was applied as a patch to HTK-3.4.1. HDecode-3.4.1 for HTK-3.4.1 was also installed. Festival-2.1, speech_tools-2.1, SPTK-3.1, openfst-1.2.6, ActiveTcl8.4.19.4, festvox-2.1, and other support software tools were installed in setting up the TTS synthesis system experimentation platform. All the above-mentioned tools are downloadable from their respective websites.

B. Data Preparation

There are 36 consonants and 5 vowels in Xitsonga [6]. A phone set for Xitsonga language was created using the 36 consonants and 5 vowels according to the procedure outlined in the *festvox* system for statistical parameter synthesis [7]. A set of letter-to-sound rules was also created. The prompts, labels and utterances were generated using the *festvox* system. The EHMM labeler included in the *festvox* system was used to generate HTK style labels. The utterances generated by the *festvox* system on the other hand, are festival style utterances.

The five phone set vowels are [6]:
[a, e, i, o, u].

The 36 phone set consonants are [6]:
[b, bv, c, d, dhl, dl, dz, f, g, gq, h, hl, j, k, l, m, n, n', ny, p, pf, py, q, r, s, sw, t, tl, ts, v, vh, w, x, xj, y, z].

Praat was used to record the speech corpus. The speech utterance recording was done in an office space with minimal noise. One speaker – the author in this case - was used to record the speech corpus of 158 sentences. The speech was recorded at 44.1 KHz stereo. The waveform files were normalized and changed to conform to 16 KHz, 16 bit, RIFF format in the *festvox* system and was then converted to little endian raw files using *Praat*. The speech signal was windowed by a 25-ms Blackman window with a 5-ms shift. We used 5-state, left-to-right context-dependent multi-stream multi-space probability distribution hidden Markov models (MSD-HMMs).

For HTS the context-dependent structure for Xitsonga was not changed since it is the same as that of English. A question set for Xitsonga was created.

C. Training Phase

The training phase begins with the extraction of speech parameters from raw files that make up the speech database and then calculate their dynamic features as depicted in Fig 1 [1, 8]. The speech parameters that are extracted are spectrum and excitation parameters which refer to mel-cepstral and log fundamental frequency respectively. Spectrum and excitation parameters are used to model context-dependent phoneme HMMs.

Speaker characteristics and speaking styles are modeled by both spectral and excitation parameters [9, 10]. The trained HMMs will, therefore, be a good representation of the individual's speaking styles and characteristics. Although mel-cepstral, log F0, and state durations are modeled simultaneously in a unified framework, they are individually modeled by multivariate Gaussian distributions, MSD-HMMs, and context-dependent n-dimensional Gaussian distributions respectively [1, 8, 11].

A decision-based clustering technique which is based on the Minimum Distance Length (MDL) criterion was applied in isolation to the distributions of mel-cepstral, log F0, and state durations of context-dependent phoneme HMMs [8, 9, 10, 12]. The impracticality and impossibility of the task of preparing a speech database that can model all combinations of contextual factors has called for such factors to be tied using the MDL clustering technique. The tied contextual factors include phoneme identity, stress-related contexts and locational contexts. In addition to tying contextual factors, the generation of spectrum and excitation parameter of newly observed vectors is done by the MDL clustering technique [13]. Expectation-maximization (EM) was used to re-estimate clustered context-dependent phoneme sequences [9, 10].

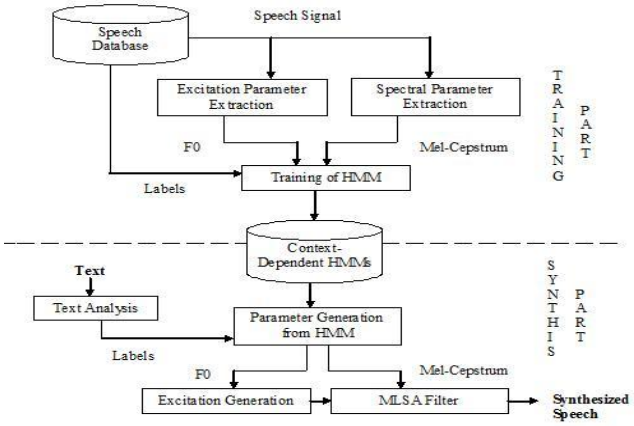

Fig 1. An HMM-based synthesis system [8]

D. Speech Synthesis

Arbitrary text to be synthesized is input, analyzed, and then transformed into a context-dependent label sequence. The HTS toolkit does not, however, include a text analyzer [14]. For this research project work, festival speech synthesis system was used as a text analyzer. The context-dependent labels are generated from utterances using festival's *dumpfeats* function. Trained context-dependent phoneme HMMs are concatenated according to the context-dependent label sequence thereby generating a sentence HMM. State duration distributions are used to determine state durations of the label sequence [10]. By combining labels from the text analyzer and context-dependent HMM, the speech parameter generation algorithm generates spectrum and excitation parameters as depicted in Fig 1. A synthesis filter receives the generated excitation and spectral parameters and produces the corresponding speech waveform.

III. SOME SPEECH SYNTHESIS CHALLENGES

The misalignment of phonemes at boundaries has been the greatest challenge to speech quality. Fig 2 is an example

of misaligned phonemes which result in the poor quality of synthesized speech. This happens because phonemes are not well represented by their corresponding sound. The given label is the equivalent of what would be generated from an utterance representing the corresponding waveform file before alignment. Fig 3 on the other hand, depicts a much better representation of the waveform by the label as a result of the automatic alignment performed by the *festvox* system.

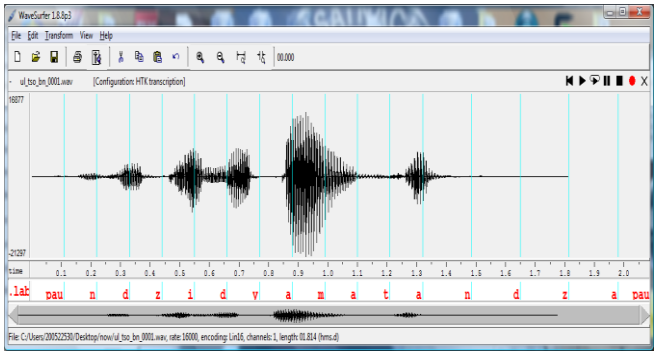

Fig 2. Default waveform to label representation before alignment of the statement - **ndzi dya matandza**

This misalignment would be engraved in an utterance file, label file and/or both. As a result, the *festvox* system generates new utterances and labels based on the automatically aligned data. When labels are generated from

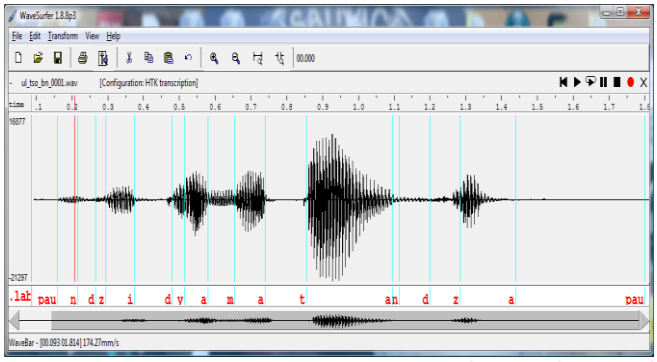

Fig 3. An automatically aligned label to waveform by festvox of the statement - **ndzi dya matandza**

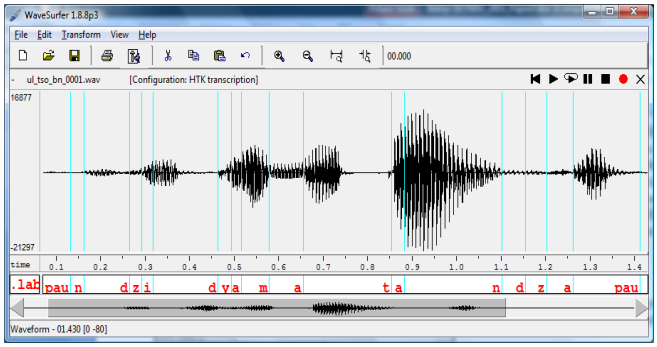

Fig 4. A manually re-aligned label to waveform by hand of the statement - **ndzi dya matandza**

utterances, it is important that the utterance files should be well aligned with the speech corpus it represents. Fig 4 depicts an attempt to further better the alignment of the speech waveform with the corresponding phoneme label by means of hand re-alignment using *wavesurfer*. In order to achieve accurate segmentation it is common for expert human labelers to approximate boundary locations of the speech waveform. Such labelers, however, take into account several available features of the waveform in order to guide them during the labeling process.

The phenomenon of noise remains a serious problem that badly affects the speech quality. Although the environment used for recoding had several noise sources, caution was taken to minimize the noise. There is, however, still a need to record the training speech corpus in a much more noise controlled environment. Recording speech at a higher sampling rate and then down sampling it to the required level seems to produce better results than recording speech at the required sampling rate.

IV. RESULTS AND ANALYSIS

The system was trained using 158 sentences and 9 sentences which were not part of the training set were used to test the system. After the synthesis of the 9 test sentences, 13 Xitsonga mother tongue speakers rated the system on the following categories: understandability, naturalness, pleasantness, and overall system impression. Rating scores ranged from 1 to 5 for each category, with 1 representing the worst case and 5 the most desirable case.

Question	**MOS**	**Meaning**
Understandability	4.1	Little effort required
Naturalness	3.8	Natural enough to listen to
Pleasantness	3.4	Acceptable
Overall	3.5	Good

Table 1 shows the results of the evaluation.

The overall system was rated as *excellent* by 7.7%, *good* by 38.9%, *acceptable* by 46%, and *poor* by 7.7% of the respondents. This means that it received an acceptability level of 92.3%. The results in Table 1 are not an indication of a perfect system, but opens up ample room for further synthesis system improvements as specified in the next section.

V. FUTURE WORK

For future work on a TTS synthesis system for Xitsonga, speech by more recruited speakers of both genders will be recorded. Training the system with speech from different speakers will be followed by adapting new speakers. Further speech recording for both training and adaptation should be done in a much more noise controlled environment. The system should also be subjected to serious evaluation by increasing the number of respondents amongst other things. More research work will be done to improve on the current quality of synthetic speech produced. A high quality vocoding method called STRAIGHT will also be used to address the speech quality factor.

VI. CONCLUSION

A highly intelligible and acceptably natural sounding speech synthesis system in Xitsonga has been developed. The developed system serves as a baseline for future improvement. This system will not only add to the pool of essential components/resources of indigenous South African spoken language systems. It will also break new grounds for further development of spoken language systems for the

language and make similar systems available for use by those that need them most.

VII. ACKNOWLEDGEMENTS

God is supreme and He comes first. Credit has to be given to Telkom South Africa (SA) and National Research Foundation for the amazing funding support they provide for research, particularly the Centre of Excellence for Speech Technology. Further appreciation goes to Telkom SA for funding my studies. Dr Nxumalo of the African Language Department at the University of Limpopo (Tufloop Campus) has made insightful linguistic inputs to this research project. My supervisors, MJD Manamela and N Gasela are indeed to be treasured. The support of my family and friends remains very dear to me.

VIII. REFERENCES

[1] Census 2001: Census in brief. Pretoria: Statistics South Africa, 2003, [online]. Available: http://www.statssa.gov.za/census01/html/CInBrief/CIB2001.pdf, accessed on: 11 May 2011

[2] Statistical release: Mid-year population estimates 2011. Pretoria: Statistics South Africa, July 2011. [online] Available: http://www.statssa.gov.za/publications/statsdownload.asp?PPN=P0302&SCH=4986, accessed on: 03 August 2011

[3] X. Huang, A. Acero, and H. Hon, Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, 2001

[4] F. Rousseau, D.J Mashao, "Increased Diphone Recognition for an Afrikaans TTS system", *Proceedings of PRASA.*, 2004, pp 113-117, Cape Town

[5] H. Zen, K. Oura, T. Nose, J. Yamagishi, S. Sako, T. Toda, T. Masuko, A.W. Black, and K. Tokuda, "Recent development of the HMM-based speech synthesis system (HTS)". In *Proc. 2009 Asia-Pacific Signal and Information Processing Association (APSIPA)*, Sapporo, Japan, Oct 2009

[6] MILAWU YA MAPELETELO NA MATSALELO YA XITSONGA. PANSALB. 2008, [online]. Available: http://www.language-inc.org/index.php?option=com_joomdoc&task=doc_download&gid=17&Itemid=173, accessed on: 11 May 2011

[7] A. Black and K. Lenzo, "Building synthetic voices", 1999. Available: http://festvox.org/bsv.pdf, accessed on: 11 May 2011

[8] T. Masuko, "HMM-Based speech synthesis and its applications", PhD thesis, Tokyo Institute of Technology, November 2002, pp. 1-84

[9] A. Conkie, "Robust unit selection system for speech synthesis", *Proc. Joint Meeting of ASA, EAA and DEGA*, Berlin, Germany, March 1999.

[10] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "Acoustic modeling of speaking styles and emotional expressions in HMM-based speech synthesis," *IEICE Trans. Inf. & Syst.*, Vol. E88-D, No. 3, pp. 503–509, March 2005

[11] J.A. Louw, M. Davel, and E. Barnard, "A general-purpose Isizulu speech sythesizer" *South African Journal of African Languages*., 2005, 2

[12] K. Tokuda, H. Zen, and A.W. Black, "An HMM-based speech synthesis system applied to English". *Proc. of 2002 IEEE Workshop on Speech Synthesis*, pp. 227–230, Sep. 2002

[13] T. Raitio, "Hidden Markov model based Finnish text-to-speech system utilizing glottal inverse filtering", Masters Thesis, 2008

[14] Meraka Institute. (2010). Publications. Tshwane, CSIR., [online]. Available: http://www.meraka.org.za/lwazi/publications/louw08speectmultilingual.pdf, accessed on: 11 May 2011

Ntsako Baloyi received his Bachelor of Science degree and his Bachelor of Science (Honours) degree in Computer Science in 2008 and 2009 respectively from the University of Limpopo. He is presently studying towards his Master of Science degree in Computer Science at the same institution. His research interests include Speech Technology, Human Language Technology and Computer Security.

An HMM-based Text-to-Speech Synthesis System for Xitsonga

Ntsako Baloyi*, MJD Manamela
Department of Computer Science
University of Limpopo (Turfloop Campus),
Private Bag X1106,
Sovenga, 0727
Tel: +27 15 2682751, Fax: +27 15 2683183
email: {200522530, jonasm}@ul.ac.za, nbaloyi11@gmail.com

Abstract-This paper outlines the plan to build an HMM-based baseline speech synthesis system for the Xitsonga language. The system to be built should produce natural sounding synthetic speech given typed or stored text input. It should further be able to model some speaker characteristics and speaking styles. Using HMMs such a system can be built without requiring a very large speech corpus for training the system. This research project forms part of the broader speech technology project that aims to develop spoken language systems for human-machine interaction using the eleven official languages of South Africa.

Index Terms—HMM-based speech synthesis, text-to-speech

1. INTRODUCTION

This study focuses on building a general-purpose Xitsonga speech synthesis system that is flexible, intelligible and natural sounding. The system should be able to model some of the desirable speaker characteristics and speaking styles. Speech synthesis systems also referred to as text-to-speech (TTS) systems receive typed or stored text as input and produce the equivalent speech waveform as output as depicted in Fig 1.

Fig1. A general outline of a TTS showing text as input, the TTS system, and speech waveform as output

The synthesis task will be carried out using an HMM-based speech synthesis (HTS) approach. This method makes it easy to accomplish our task without requiring a very large training speech corpus. The HTS approach requires much less speech data to train the system as compared to the concatenative-based synthesis approach. In addition to a smaller training corpus, HTS also requires very little memory for the synthesis engine at runtime. As a result, TTS systems based on this approach can easily be integrated into handheld devices [1].

Speech synthesis systems have over the years been developed for various languages all over the world. Most of those systems have been developed using unit selection methods which have proved to produce very natural sounding speech; though at the cost of very large speech training corpus. There is currently no TTS in Xitsonga at the University of Limpopo speech technology program, thus the system to be built will be the first of its kind for this language. The results of the 2001 census indicated that by that year Xitsonga was spoken by about 4.44% of South African citizens as their home language.

Speech synthesis systems have various areas of applications: hands and eyes-free computer interaction, email readers, translation systems, learning assistant systems for the handicapped, speaker verification systems etc. [1]

The remainder of the paper is outlined as follows: section 2 gives a brief overview of components that makeup our HMM-based speech synthesis system. Section 3 gives concluding remarks.

2. THE HMM SPEECH SYNTHESIS SYSTEM

2.1. Recording of speech database

A regular good-quality microphone will be used to record the speech corpus. The recording process will be executed in a specialized noise-free room. Speech by several speakers will be recorded and stored in the speech database, though there will be one main speaker. The recording process will occur over a number of days for all the speakers in order to capture variability in speakers' speaking characteristics. Phonetically balanced sentences in Xitsonga will be used for recording. These sentences should be from a wide variety of areas in order to make the system as much general-purpose as possible. Multiple instances of phoneme HMM utterances may be stored in the speech database in order to represent different contexts.

2.2. Training phase

Initially, spectrum (mel-cepstral coefficients) and excitation (log fundamental frequency or log F0) parameters are extracted from the speech database and their dynamic features (delta and delta-delta coefficients) are calculated [1, 2]. The extracted parameters model speaker characteristics and speaking styles and they are used to train (or model) the context-dependent phoneme HMMs [3]. Spectrum parameters are modeled by multivariate Gaussian distributions, whereas excitation parameters are modeled by multi-space probability distribution hidden Markov models (MSD-HMMs) [1].

A decision-based clustering technique which uses the Minimum Distance Length (MDL) criterion is applied separately to distributions of mel-cepstral, log F0 and state durations of the context-dependent phoneme HMMs [2, 3]. This technique ties contextual factors (i.e. phoneme identity, stress-related and locational contexts) that are almost similar. This is done because it is both impractical and impossible to prepare a speech database that can model all combinations of contextual factors. A re-estimation of the clustered context-dependent phoneme sequence will then be performed using the expectation-maximization (EM) algorithm [3]. Clustering is also used to generate excitation and spectrum parameters for newly observed vectors, i.e. observation vectors not included in the training corpus [4].

State durations are modeled by context-dependent n-dimensional Gaussian distributions which are then clustered by a decision tree. State densities capture/model the temporal structure of speech [2, 5]. Mel-cepstral coefficients, log F0 and state durations will be modeled simultaneously in a unified framework of HMM [2, 5].

2.3. Adaptation phase

A speaker adaptation technique is used to adapt a target speaker using the trained HMM from the training phase. Tikashi Masuko in [1] indicates that the adaptation technique requires only a small amount of speaker adaptation data from the target speaker. The adaptation process may be done using an individual speakers' speech data or by averaging several speakers' speech data [1]. Speaker voice characteristics, styles or even emotions can be modified/updated by transforming HMM parameters using adaptation or other methods (such as interpolation and eigenvoices) [3].

2.4. Synthesis phase

An arbitrary text to be synthesized will be input, it will then be transformed into a context-dependent phoneme label sequence. A sentence HMM should then be generated by concatenating the adapted context-dependent phoneme HMMs from the adaptation phase according to the context-dependent phoneme label sequence. State duration distributions are then used to determine state durations of the label sequence [3]. A speech parameter generation algorithm is then used to generate spectrum and excitation parameters from the context-dependent phoneme HMMs. A synthesis filter is used to synthesize a speech waveform from both spectral and excitation parameters [2, 3].

2.5. Evaluation of the HTS

A black-box evaluation method [6] will be used to evaluate the performance of the system for naturalness and intelligibility. Xitsonga mother tongue speakers will be selected ranging from less literate to professionals. These candidates will evaluate the system and rate it with respect to its observed naturalness and intelligibility separately. They will be given a range of possible options to select one that best characterizes their opinion. The mean opinion score (MOS) of all the results will be calculated at the end.

2.6. HTS Toolkit

The HMM-based synthesis system has a toolkit which is provided as a patch to HTK. For this project, the HTK 3.4.1 embedded with HTS version 2.1.1 will be used for experimentations [7].

3. CONCLUSION

The development of the first baseline Xitsonga TTS using an HMM-based speech synthesis (HTS) approach as part of the broader project of speech technology will increase the number of essential components of indigenous South African spoken language systems. The system should be highly intelligible and natural sounding; and should also model some of the desirable speaker characteristics well.

4. REFERENCES

[15] T. Masuko, "HMM-Based speech synthesis and its applications", PhD thesis, Tokyo Institute of Technology, Nov. 2002, pp. 1-84

[16] K. Tokuda, H. Zen, and A.W. Black, "An HMM-based speech synthesis system applied to English". Proc. of 2002 IEEE Workshop on Speech Synthesis, pp. 227–230, Sep. 2002

[17] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "Acoustic modeling of speaking styles and emotional expressions in HMM-based speech synthesis," IEICE Trans. Inf. & Syst., vol. E88-D,no. 3, pp. 503–509, Mar. 2005

[18] T. Raitio, "Hidden markov model based Finnish text-to-speech system utilizing glottal inverse filtering", Masters Thesis, 2008

[19] H. Zen, K. Oura, T. Nose, J. Yamagishi, S. Sako, T. Toda, T. Masuko, A.W. Black, and K. Tokuda, "Recent development of the HMM-based speech synthesis system (HTS)". In *Proc. 2009 Asia-Pacific Signal and Information Processing Association (APSIPA)*, Sapporo, Japan, Oct 2009

[20] X. Huang, A. Acero, and H. Hon. Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, 2001

[21] K. Tokuda, H. Zen, J. Yamagishi, T. Masuko, S. Sako, A.W. Black, and T. Nose, HMM-based Speech Synthesis System (HTS). May 2010, [Online] Available: http://hts.sp.nitech.ac.jp/

Baloyi Ntsako received his Bachelor of Science degree and his Honours degree in Computer Science in 2008 and 2009 respectively from the University of Limpopo and is presently studying towards his Master of Science degree at the same institution. His research interests include Speech Technology and Computer Security.

An HMM-based Text-to-Speech Synthesis System for Xitsonga

Ntsako Baloyi*, MJD Manamela

Department of Computer Science, University of Limpopo, South Africa, 200522530@ul.ac.za

Abstract

This paper outlines the plan to build an HMM-based baseline speech synthesis system for the Xitsonga language. The system to be built should produce *natural sounding and intelligible* synthetic speech given typed or stored text input. It should further be able to model some speaker characteristics and speaking styles. Using HMMs such a system can be built without requiring a very large speech corpus for training the system. This research project forms part of the broader speech technology project that aims at developing spoken language systems for human-machine interaction using the eleven official languages of South Africa.

Introduction

Speech synthesis systems also referred to as text-to-speech (TTS) systems receive typed or stored text as input and produce the equivalent speech waveform as output as depicted in Fig 1. The system to be built is for the Xitsonga language and will be built using an HMM-based synthesis approach. In 2001, the census results indicated that Xitsonga was spoken by 4.44% (almost 2 million) of the South African population. Synthesis systems can be used by the visually challenged as learning assistants or as e-mail readers.

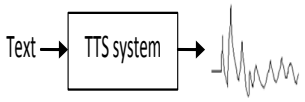

Fig. 1. A typical TTS system block diagram

HMM-based Speech Synthesis System (HTS)

The HMM-based speech synthesis approach, unlike the unit selection approach does not require very large speech corpus for training the system [1]. Although the naturalness of the speech produced by HMM-based approaches is a bit lower than that of unit selection, they make it easy to adapt new speakers with little adaptation data. The small size requirements of its runtime synthesis engine (about 2MB), makes it easy for use in portable devices [2].

HTS is a free toolkit that is used for HMM-based synthesis. This toolkit is provided as a patch to HTK [2]. HTS does not, however, include a text analyzer; hence Festival will be used for that purpose in this project.

First, speech by target speakers will be recorded and stored on the speech database.

The *training* phase is similar to that in ASR [2]. In this phase, excitation and spectral parameters are extracted from the speech database and their dynamic features calculated as in Fig 2. These parameters and their dynamic features are used for training and the end product of the training phase, are context-dependent phoneme HMMs.

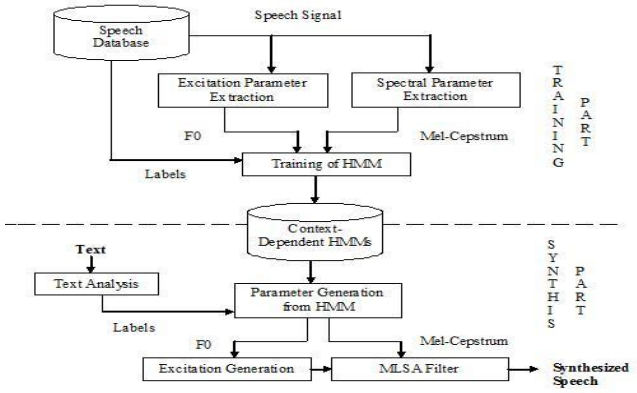

Fig 2. An HMM-based synthesis system

For *synthesis*, the input text is transformed into a label sequence [1]. Sentence HMMs are generated from the label sequence and their state durations are determined. Spectral and excitation parameters are generated and the MLSA filter then synthesizes the speech waveform.

A speaker *adaptation* technique is used to adapt a target speaker using the trained HMM from the training phase [1]. The system will be evaluated based on speaker characteristics, naturalness, and intelligibility.

Conclusion

The development of the first baseline Xitsonga TTS using an HMM-based speech synthesis (HTS) approach as part of the broader project of speech technology will increase the number of essential components of indigenous South African spoken language systems. The system should be highly intelligible and natural sounding; and should also model some of the desirable speaker characteristics well.

References

[1] T. Masuko, "HMM-Based speech synthesis and its applications", PhD thesis, Tokyo Institute of Technology, Nov. 2002, pp. 1-84

[2] K. Tokuda, H. Zen, and A.W. Black, "An HMM-based speech synthesis system applied to English". Proc. of 2002 IEEE Workshop on Speech Synthesis, pp. 227–230, Sep. 2002

TCoE4ST

an initiative of the dti

A Text-to-Speech Synthesis System for Xitsonga

Ntsako Baloyi[1], **MJD Manamela**[1], **Nalson Gasela**[1]
University of Limpopo[1], *Department of Computer Science, Private Bag X1106, Sovenga, 0727*
e-mail address: nbaloyi11@gmail.com

Abstract

The development of a text-to-speech synthesis system using the HTS toolkit based on hidden Markov models can be achieved without requiring a massive training speech corpus. Intelligible and natural sounding speech is therefore achievable. The quality of synthesized speech, however, does not equal that of recorded speech. Some speaker characteristics and speaking styles are modeled by trained hidden Markov models from context-dependent labels. Xitsonga is a resource-scarce language spoken in more than three Southern African countries. The mean opinion score tests attain 92.3% of acceptability from Xitsonga mother tongue speakers.

Introduction

A text-to-speech (TTS) synthesis system for a particular natural language is the technology for converting a given typed or stored text input into its equivalent spoken waveform format. The focus of building the Xitsonga baseline TTS synthesis system has been on producing flexible, intelligible and natural sounding speech – a first to our speech technology research group. The process of developing a text-to-speech system using one of the indigenous languages of South Africa is a practical step towards making both information and technology accessible and easy to use by all individuals at different literacy levels, even for the visually challenged. TTS systems can be used as message readers, learning assistants, tools to aid in communication and learning for the handicapped and visually challenged.

Methodology

A statistical parametric approach to speech synthesis based on hidden Markov models (HMM) has been the method of choice over other speech synthesis methods like articulatory synthesis, unit selection, formant synthesis, etc. An HMM-based speech synthesis toolkit called HTS, which is provided as a patch to an HMM-based speech recognition toolkit called HTK is used. By using this method, flexible, intelligible, and natural sounding speech can be synthesized without requiring a large training corpus and using a small runtime synthesis engine [1]. The runtime synthesis engine excluding text analysis can be as small as 2MB, thereby making it viable for use even in mobile devices. In the training stage, spectrum (mel-cepstrum) and excitation (log F0) parameters, which model speaker characteristics and speaking styles are extracted. Mel-cepstrum, log F0 and state durations are simultaneously modeled using a unified framework. The synthesis phase starts with text input which is converted into a context-dependent label. Context-dependent labels, together with trained HMMs are fed into a parameter generation algorithm. The parameter generation algorithm produces excitation and spectrum parameters which will be used by the synthesis filter to output a synthesized speech waveform.

Results and discussion

The system was trained using 158 sentences and 9 sentences which were not part of the training set were used to test the system. After the synthesis of the 9 test sentences, 13 Xitsonga mother tongue speakers rated the system on the following categories: understandability, naturalness, pleasantness, and overall system impression. Rating scores ranged from 1 to 5 for each category, with 1 representing the worst case and 5 the most desirable case.

Question	Mean Opinion Score	Meaning
Understandability	4.1	Little effort required
Naturalness	3.8	Natural enough to listen to
Pleasantness	3.4	Acceptable
Overall	3.5	Good

Table 1. The mean opinion scores (MOS) obtained during system evaluation

The overall system was rated as *excellent* by 7.7%, *good* by 38.9%, *acceptable* by 46%, and *poor* by 7.7% of the respondents. As can be seen from Table 1, the understandability is said to be natural enough to listen to, with the overall system being rated as good at 3.5 MOS. The overall acceptability level of the system was found to be 92.3%. A highly intelligible and acceptably natural sounding speech synthesis system has been developed. There, however, remains a lot to be done towards improving the naturalness of the system. The training corpus will be increased from 158 to about 500, with the number of speakers also being increased to two females and two males for training the system. The order of mel-cepstrum will be increased from 24 to 40, the number and diversity of respondents will also be increased. Spectral transformation and representation of weighted spectrum (STRAIGHT) will be used, the adaptation phase will be included and the number of test sentences will be increased.

References

[1] K. Tokuda, H. Zen, and A.W. Black, “An HMM-based speech synthesis system applied to English”. Proc. of 2002 IEEE Workshop on Speech Synthesis, pp. 227–230, Sep. 2002