INVESTIGATION INTO AUTOMATIC SPEECH RECOGNITION OF DIFFERENT DIALECTS OF NORTHERN SOTHO

by

Madimetja Asaph Mapeka

[redacted]

Mini-Dissertation

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

in the

SCHOOL OF COMPUTATIONAL AND MATHEMATICAL SCIENCES,

FACULTY OF SCIENCES, HEALTH AND AGRICULTURE

at the

UNIVERSITY OF LIMPOPO - TURFLOOP CAMPUS

Private Bag X1106, SOVENGA, 0727, South Africa

SUPERVISOR: PROF HJ OOSTHUIZEN

CO-SUPERVISOR: MR MJD MANAMELA

DECEMBER 2005

ACKNOWLEDGEMENTS

It is a very pleasant task to express my thanks to all those who contributed in many ways to the success of my study.

Firstly, let me take this opportunity to thank the Almighty God for giving me life and making all things possible.

A big thank you goes to my supervisor Prof HJ Oosthuizen and co-supervisor Mr MJD Manamela for their tremendous educational guidance, time, patience, encouragement, critical remarks and support throughout my master's studies. It has always been a great pleasure to work with the two of them.

Let me thank my late grandparents, Madimetja Asaph and Mantati Jane Mapeka, for having taught me the rules and regulations of life and for encouraging me to further my studies at tertiary level. I am very proud of you, just wish you were here.

Special and heartfelt thanks to my mother, Mahlodi Grace Mapeka, for making it possible for me to archive my goal. Studies of this nature would be impossible without your support, morally and financially. Without you, I would not be where I am today.

I owe a lot of thanks to my sister, Sekgabela Frangelinah Mapeka, for her love and continued encouragement. Remember, you were always there when I needed to talk through how I was feeling about my work and thanks for listening to my endless complaints when things weren't going my way.

Of course I would like to extend my sincere acknowledgement to my family, friends and colleagues for their endless, supportive ideas.

This list would be incomplete without a very big thank you to TELKOM SA, for sponsoring me, HP SA, Marpless and National Research Fund through THRIP for making this study possible financially.

Finally, let me thank all participants. Guys, you played a very important role in my research project.

DECLARATION

I declare that this dissertation hereby submitted to the University of Limpopo for the degree of Master of Science has not previously been submitted by me for degree purposes at this or any other university, that it is my own unaided work in design and in execution, and that all material contained therein has been duly and appropriately acknowledged.

Signed: _____

Date: _____

ABSTRACT

This research report focuses on investigating the extent to which it would be possible to develop and train a speaker-independent automatic continuous speech recognizer that can handle three different dialects of Northern Sotho (Selobedu, Setlokwa and Sepedi). For this research project, telephone speech data was collected from the different dialect regions of the target speakers. A total number of 90 calls constituting 30 per dialect were considered valid. Manual transcription of the collected speech data preceded the creation of the speech recognizer. The speech recognizer was based on hidden Markov models as used in the Hidden Markov Model Toolkit (HTK).

The pronunciation dictionary which incorporated multiple pronunciations of the three selected dialects was designed. An overall performance of 43% was obtained from the pre-recorded test samples. As far as recognition accuracy is concerned, 16.7% of sentence accuracy and 52.7% of word recognition was obtained from live test of the system. An automatic speech recognizer thus produced will have an appeal to a wider section of the Northern Sotho speakers, thus enhancing the chances of more user-friendly interaction in voice-driven applications for the provision of e-services. This automatic speech recognizer will also play an important role in the marketing of speech technology emerging and enabling human-computer interface technology suited to local and domain specific consumer applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND SYMBOLS

AI	Artificial Intelligence
ASR	Automatic Speech Recognition
BSAE	Black South African English
CNF	Context Free Grammars
DARPA	Defense Advanced Research Projects Agency
DTMF	Dual-Tone Multi-Frequency
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
HTK	Markov Model Toolkit
ISDN	Integrated Services Digital Network
IVR	Interactive Voice Response
LDA	Linear Discrimant Analysis
MAP	Maximum-A-Priori
MFCC	Mel-Frequency Cepstrum Coefficients
ML	Maximum Likelihood
MLF	Master Label File
OOV	Out Of Vocabulary
RSI	Repetitive Strain Injuries
SA	South Africa
TTS	Text-to-Speech
VODER	Voice Operating Demonstrator
WER	Word Error Rate

CHAPTER 1: INTRODUCTION

1.1 Background

Automatic speech recognition (ASR) is amongst the technologies that enable human beings to interact with computers by simply talking to them using their natural languages. With increased developments in modern technology, communication between humans and computers is beginning to take place more frequently.

The current state of computer technology allows that an automatic speech recognition system can be designed and developed by humans. They can train it to behave according to their specific design requirements. This process of developing a "hearing" system requires hard-working, dedicated and committed intellectuals/researchers for useful results to be generated. Human-computer interaction (HCI) forms one of the modern ways to facilitate the use of computers by illiterate and physically challenged individuals.

1.2 Research goals and rationale

The main purpose of this research project is the investigation as to what extent it would be possible to build an automatic continuous speech recognizer to handle a few different dialects of Northern Sotho simultaneously. Comparison with the baseline system developed by Modiba [25] is also made to appraise the results obtained.

Another reason for embarking on this speech technology research project is to contribute useful tools for HCI in an attempt to bridge an existing technological illiteracy of most rural communities of the selected speaker population. We presume that building a speech recognizer will entice and enable them to interact with the computers in a more user-friendly way using any of the three dialects of Northern Sotho (*Setlokwa, Selobedu and Sepedi*). This research project will also play an important role in laying the foundation for the marketing of speech technology for consumer applications.

For application development purposes, the resultant speech recognizer might serve as an important tool for use in e-government service provisioning, for example, reporting of births and deaths of people from remote locations using an interactive voice response system.

The main objectives of this speech technology research project are as follows:

- Fully understand the process of automatic continuous speech recognition. This will be accomplished by considering relevant research topics from speech technology journals and books as well as mastering how the selected technological approach to ASR, namely, the use of the hidden Markov model toolkit (HTK), works and its utility in developing and training a speech recognizer.
- Investigation of a variety of supporting speech technology techniques as well as new approaches for systematic application of the techniques to our research project.
- The training of an automatic speech recognizer for a few dialects of Northern Sotho.

1.3 Research Expectations

This research project serves as an initial study on natural language dialects in the context of speech technology. It attempts to show how to train an automatic speech recognizer for the different dialects. One of the major expectations is to produce a reasonably working automatic speech recognizer which will be trained on dialects of Northern Sotho. The user would say what s/he wants to say in Northern Sotho and the recognizer will reproduce what has been said. The size of the training vocabulary should be large enough to allow users to say common words in Northern Sotho, particularly those that are dialect sensitive.

The speech recognizer should ideally be speaker-independent (i.e., designed for a variety of users) and should as well allow continuous speech input from the users. The recognizer should produce a low word error rate (WER), but on the other hand

produce high recognition accuracy (which generally measures the performance of the recognizer).

Finally, the interface on which the user interacts with the recognizer should be simple and user-friendly and thus fitting the demands of most users.

1.4 Dissertation Outline

In this section we briefly give details of what constitutes each chapter included in this mini-dissertation. Chapter 2 starts by giving a brief history on speech technology and its architectures. A convergence graph on speech recognition technology is also shown. An underlying theory of speech architectures and how they were incorporated to yield the existing speech-enabled applications are covered. The research projects that dealt with the different dialects and the results obtained are also discussed in this chapter.

The different approaches of automatic speech recognition and basically the theoretical background of establishing a speech recognizer are covered in Chapter 3. The approaches mentioned are Acoustic-phonetic, Statistical Pattern Recognition and Artificial Intelligence (AI). The engine on which our speech recognizer runs, called the HTK toolkit, is also discussed. This chapter also covers the ways in which speech samples were modeled and ends by discussing the different phases of developing and training a speech recognizer.

Chapter 4 outlines the speech data collection process. It includes the manner in which the respondents were recruited, the design and distribution method of prompt sheets, the validated data that was recommended to be utilized for the building process. It also includes a brief history on the origin of the language called Northern Sotho and its dialects. A map reflecting the dialectal regions where speech data was collected is shown. The main phonemes that brought changes according to their pronunciation variation are outlined in this chapter.

In chapter 5, we discuss the practical aspects of building a speech recognizer. This is the actual implementation of the speech recognition system. Of utmost importance are the files needed for building the recognizer, for example, the presentation of the pronunciation lexicon covering multiple pronunciations. This chapter also discusses some of the important commands invoked during processing, together with their meanings.

Chapter 6 analyses the results that were obtained during the experimentation phase and compare them to the results of the baseline system developed by Modiba [25]. A discussion on how the initial dialect-based system was improved to yield the reported results is also included in this chapter. Finally, the major difficulties encountered in building a dialect-based automatic speech recognizer are discussed.

Chapter 7 serves as the final chapter and states the conclusion which is based on the evaluation of the performance of the developed system. This chapter also includes a discussion on proposed future work to be done on the developed Northern Sotho dialect-based automatic speech recognizer.

1.5 Summary

In this chapter a brief outline of the contents of this mini-dissertation was given.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter starts by giving a brief outline of speech technology architectures and an overview of the speech technology research field. It also presents an underlying theory of such architectures and how everything is integrated to establish existing applications. A brief history of speech technology from early times till the present is also included. A detailed outline of research conducted into the speech recognition of different dialects of natural languages is included. Finally, the technological applications that came about after the invention of speech technology are mentioned.

2.2 Overview of Speech Technology

Speech technology (usually encompassing speech synthesis and speech recognition) is one of the technological fields that has been developed gradually and possesses large and meaningful implications for the transcription service organization [4]. Broadly stated, speech technology can also be viewed as a technology that gives special attention to direct interaction between human beings and computers, through a communication mode humans commonly use among themselves [35]. For this technology to mature and reach its current levels it took researchers a lot of time and effort. Figure 2.1 clearly depicts speech technology research fields and a large variety of existing categories.

> "*Acceptance of a new technology by the mass market is almost always a function of utility, usability and choice. This is particularly true when using a technology to supply information where the former mechanism has been human*" [24].

Speech as a medium of communication between humans has shown its importance in their day-to-day interactions. Only recently has it become apparent that it can also be a proper medium of communicative interaction with machines in current applications such as those embedded in electronic devices [13]. Early attempts of deploying speech technology applications in business settings came into existence. Voice operating

demonstrator (VODER) is amongst the earliest applications and was developed by Bell Labs [1].

However, the majority of the earliest speech applications were developed to satisfy restricted academic requirements and they worked successfully at that level. When implemented at the realistic business and user level, disappointments were met because they could not perform satisfactorily enough at this level. Since then, the speech technology research field struggled to develop sound and accurate applications [8].

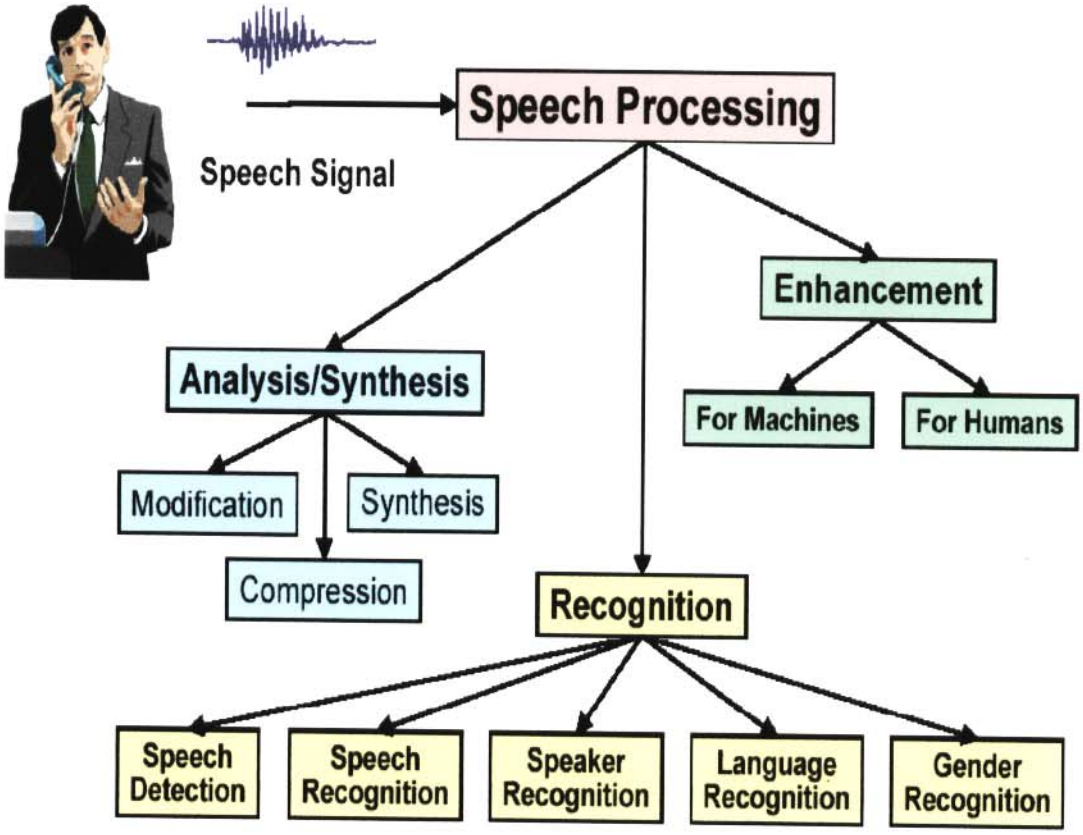

Figure 2-1: Categories of speech technology (adapted from [15]).

As Dunn [15] captures the complete picture, Figure 2.1 clearly shows how a speech processing hierarchy plays a prominent role in speech technology for human-computer interaction and generally shows how wide this field of research is. In this chapter, we will briefly cover only the descriptions of both speech recognition and speech synthesis.

2.2.1 Speech Recognition

Cole [11] defines ASR as simply a way of changing an acoustic signal or spoken language, uttered through a microphone or a telephone line, to a set of words or similar form, preferably written text. The recognized words can then serve as final results, i.e., in applications such as command and control, data entry, document preparation and call centers. In the context of interactive voice response (IVR) systems, such an ASR system can be regarded as a multimedia browsing tool which allows us to access internet resources such as recorded audio and video data [13].

The growth of the automatic speech recognition research has been progressing well over the past few years [1]. Many companies continue to utilize the recognition software for cost reduction and an improvement in customer satisfaction. Early ASR software packages such as those produced in the mid-nineties were restricted to perform only specific tasks and it required a long time for the user to train, or adapt, the speech technology software to function with a particular dialect [35].

Speech recognition incorporates a useful form of input during interaction with machines and plays an important role for people working in interactive environments such as hospitals, people with handicaps such as blindness or palsy, people who are illiterate or even when their eyes or hands are busy [13]. With the availability of automatic speech recognition applications, other tasks such as typing would be minimized since the best possible and natural mode of input would be to utter words into the computer before it generates the corresponding text.

Among the earliest applications for ASR were automated systems and medical dictation software. If it was really possible to combine natural language understanding with speech recognition, the union of the two would make the use of computers more appealing in many disadvantaged communities who do not view learning the technical side of using computers as an advantage.

In addition, vocabulary limitation was another form of channeling the speech recognition software to perform well. Obviously, these systems would often fail when

the speaker has an accent (dialect) that was not fully defined during the training of the system, i.e., the performance of the system deteriorates when test data contains dialects that are dissimilar from the original training sample speech data [6].

As the research into ASR proceeds to meet customer requirements and demanding industrial factors, software development convergence would probably be one of the threats that researchers might encounter. Since ASR research has become more adequate, the transcription service industry has operated simultaneously with the research to develop a tight relationship with computers. Figure 2.2 clearly reveals the possible convergence of transcription industry practices with speech recognition technology.

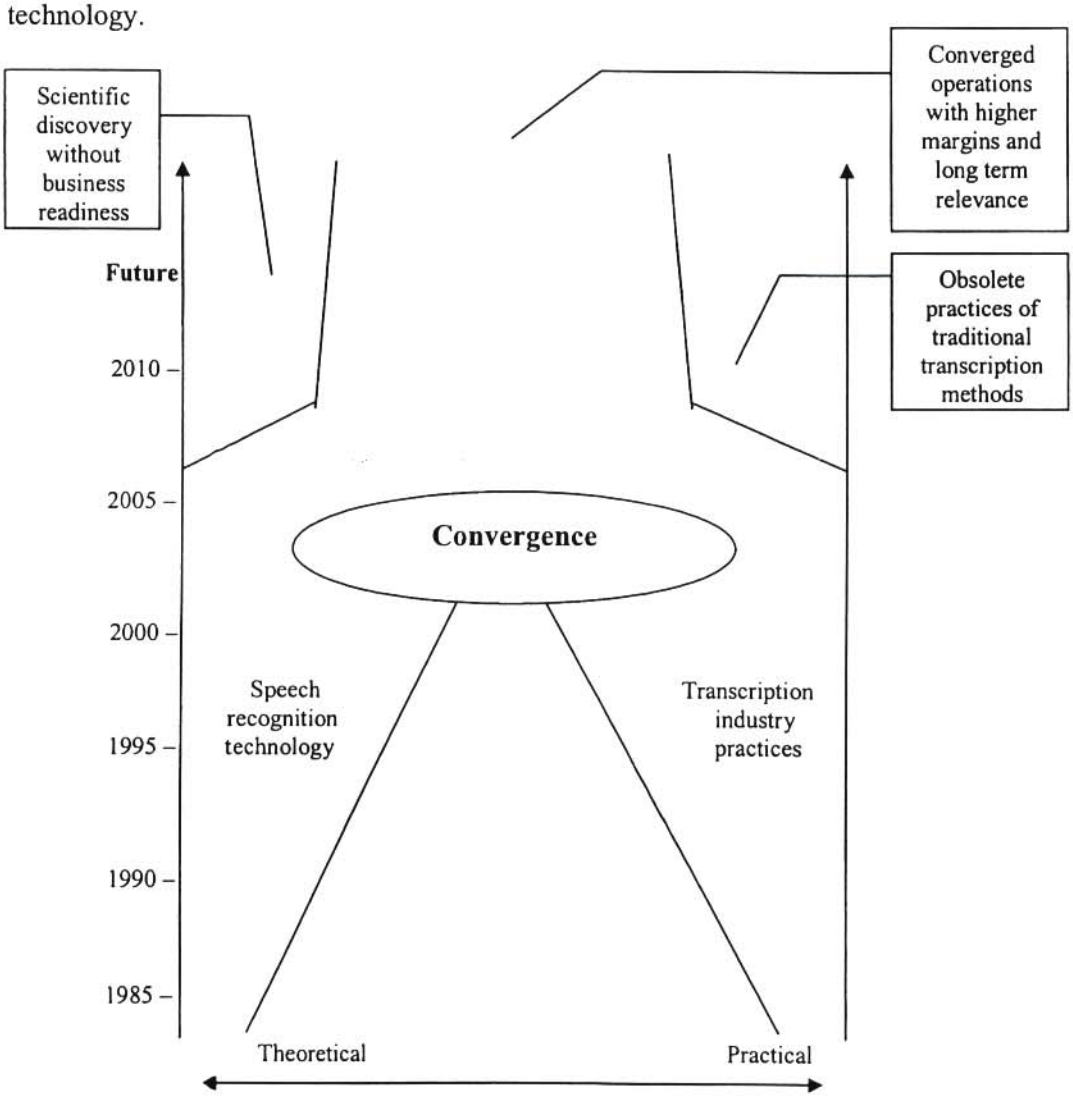

Figure 2-2: Convergence graph (adapted from [4]).

Although most commercially available speech recognition systems use similar processes, e.g. phoneme-based hidden Markov models (HMMs) and statistical language models, there are distinct differences in speed and error rates. But more essentially, there will be dissimilarities in the manner in which the user is required to speak. Some systems require the user to speak with short pauses between each word, a method called discrete speech recognition technology.

In contrast, Philips natural speech recognition technology allows users to speak naturally, without pauses, as if one is having a conversation. ASR systems that possess or aim at a high degree of naturalness have the extensive possibility of becoming accepted by users because they can easily accommodate and handle normal speech as input [8].

2.2.2 Speech Synthesis

Text-to-speech (TTS) synthesis is briefly the art of creating talking machines. The process includes transforming a string of words into spoken language that is played through the computer speakers or any other related output devices [13]. The implementation steps of a TTS system are not included in this dissertation.

It has always been a goal of speech technology researchers to construct a machine that would actually generate speech but only within the last 50 years it became clear that such a machine could really exist [7]. Practical examples of text-to-speech synthesis systems that could say or pronounce any word given to them, although sometimes pronouncing it incorrectly of course, have been accomplished recently. As discussed briefly by Sproat [34], there are several types of speech synthesis applications that have already been looked into such as:

- Articulatory synthesizers, which serves as a preliminary representation for constructing the movement of the articulators and the acoustics of the vocal tract.
- Formant synthesizers which utilizes acoustics as the initial step.

- Concatenative synthesizers which utilize databases of stored speech to fit together the new utterances. Even today, all commercial systems are concatenative-based, with the majority being named under the unit selection approaches.

There has been a steady improvement in the research of speech synthesis [7]. This connotes that the research has progressed positively towards achieving the needs of customers.

2.3 History on Speech technology

With the accelerated pace of technological development, the implementation of speech technology applications has become a common process.

Henry [18] made the following remark that is relevant to the pace of technological development in computational devices: "*Virtual reality and multimedia capabilities that were wild fantasies only five years ago are now commonplace, and computers and embedded systems play an important role in our daily lives*".

The research into speech technology commenced as early as 1936 when AT & T's Bell Labs brought forth the first electronic speech synthesizer called the voice coder (VODER) which was verified beyond doubt at the World Fairs by experts who used a keyboard and foot pedals to play the machine and utter speech. From then, speech technology researchers opted to put everything on halt after having myths that Artificial Intelligence would pave their way for future research, but all in vain [30].

During the 70's, Lenny Baum of Princeton University established a better algorithmic approach called hidden Markov model (HMM) which is one of the methods that could be used to build an automatic speech recognizer. All major companies selling speech technology applications then followed the HMM pattern-matching strategy, seeing it as an advantage in the marketing of the speech technology over other strategies being used [8]. A speech recognizer that processes and understands continuous speech was later established by the Defense Advanced Research Projects Agency (DARPA) under

the speech understanding research program. The main aim behind the building of such a system was to construct an accurate continuous speech recognizer.

Later, techniques were established to produce speech in a way that it could be more easily used in applications. The Texas Instruments Speak 'n Spell toy, was introduced [7]. Charles Schwab devoted resources towards developing speech technology applications with Nuance (supplier of speech technology solutions). Dragon produced "Naturally Speaking" which was the first continuous speech dictation software available [1]. As the speech technology research continued, researchers trained and tested their systems using locally collected speech data without being very attentive to defining the boundaries of the training and testing sets. Table 2.1 shows the timeline of speech technology research highlights, i.e., it briefly outlines in sequence what transpired during the earlier years in this speech technology research area.

The major setback with these early speech technology research endeavors is that their center of interest and focus was primarily based on speech output. As a result, it was very difficult to compare performance across systems and to build an accurate recognition system.

In recent years ASR has attained a high level of performance in aspects such as recognition rate, accuracy, word-error rates, etc [9]. According to Jelinek [21], the current levels of ASR performance has been raised impressively.

1936	AT&T's Bell Labs produces Voder (Synthesizer) and was demonstrated to the world Fairs in 1939.
1969	John Pierce argues that speech & voice recognition will never exist since it requires artificial intelligence.
Early 1970's	Lenny Baum of Princeton University establishes an alternative approach to building a recognizer called HMM and this was adopted by all technology leading companies such as Dragon systems, IBM, Philips and others.
1971	DARPA establishes a program for developing a computer system that

	could understand continuous speech and this program was initiated by Lawrence Roberts.
1978	The popular toy "Speak and Spell", which utilizes a speech chip for establishment of human-like digital synthesis sounds, is introduced by Texas Instruments.
1982	An innovative speech and language technology called Dragon systems is founded by speech technology pioneers, Drs. Jim and Janet Baker. Convox is also brought to birth.
1993	Convox sells its products out to Creative Labs, Inc.
1995	Dragon releases the first time dictation speech recognition technology software and is made available to consumers.
1996	Charles Schwab lends away its resources for the development of an IVR system with Nuance. The program called Voice Broker which allowed customers to call in for quotations on stock was found to be 95% accurate and set the stage for other companies. BellSouth also launches the first voice portal called VAL.
1997	The first "Natural Speaking" dictation software is introduced by Dragon, a continuous speech recognizer.
1998	Microsoft develops a partnership with Lernout & Hauspie giving Microsoft access to utilize speech recognition technology.
1999	Microsoft acquired Entropic, giving Microsoft full access to the most accurate speech recognition system in the world.
2000	World-wide voice portal is introduced by TellMe. Dragon systems are acquired by Lernout & Hauspie for approximately $460 million. NetBytel launches the world's first voice enabler.
2001	ScanSoft closes acquisition of Lernout & Hauspie speech and language assets.

Table 2-1: A timeline of speech recognition (adapted from [1]).

Ever since the start of research in ASR technology, the common barriers which affected the speed and accuracy of speech recognition were basically computer speed and power. But these days' accuracy levels have been extended to 95% and with better transcription speed at over 160 words per minute [1].

Considering the similarities of the earliest speech technology with modern speech technology, the research in ASR has changed from template-based approaches to statistical modeling methods especially when coming to the HMM approach. Although we are now on a platform where talking computers broadly exist, there is still a lot of effort to be put into this field of research.

Speech dialects

Dialects are referred to as sublanguages of another language and include different usage for pronunciation, lexicon and syntax. For example, In Japan, a "0 (zero)" can be pronounced as "dero", "jero", "zeru" [22] and the latter pronunciations are regarded as the dialects of Japanese.

Addressing dialects is fast becoming one of the most important issues in speech recognition since sub-languages with greater incidences of pronunciation variability exist in great numbers. Today, dialect-based systems can reach higher performance levels with an outstanding ability to handle different dialects [31]. A system being able to recognize multiple dialects simultaneously is much more advantageous to market than a system that recognizes only one dialect of a language, mainly for reasons of customer satisfaction.

In 2000, in South Africa (SA), the University of Stellenbosch conducted such a similar speech technology study on Black South African English (BSAE) [31]. The study has been completed under the supervision of Professor J.C Roux and it was primarily focused on the English language as spoken by numerous indigenous speaker populations of SA. English is generally spoken at different levels in the nine provinces of South Africa.

The studies focused on most provinces. The language English serves as a medium of instruction and is spoken by people residing in different provinces. The pronunciation of words differs according to the geographical regions they come from. The difference in pronunciations constitutes dialects. It also covered a variety of other things like, for example, English spoken by an Afrikaner staying in the Western Cape and those staying in Gauteng differs in terms of pronunciation of words.

The developed system made a clear comparison between the recognition rate using BSAE and a pure English-based speech recognizer, as well as other speech applications. That is, it compared the performance levels of two different recognizers: a speech recognizer built for English specifically and the one built for BSAE.

In Japan, a similar study again was undertaken. The study focused solely on the data collection of Japanese dialects and its influence on automatic speech recognition, as reported by Kudo et al. [22] A database with 8 886 speakers was initiated through voice donation by a telephone line. Personal data such as gender, age and place of origin were also of great importance to be included in the sheets distributed for data collection. The geographical areas where data was collected were identified.

Each data collection prompt sheet distributed contained 8 sentences which included items such as telephone numbers, 4 tri-phone balanced sentences and 2 preliminary yes-no answers. Only 122,570 files were used to build the system [22].

Experimentation was made on what percentage does dialects affect the recognition rate, and the influence at the scale of 3 000 speakers produced a 2-4% influence to the speech recognition rate. The error rate increased accordingly with 2-4% and preferably the combination of training and testing sets were selected from the same dialects. The influence on the accuracy rate was cleared through experimentation on speech recognition [22]. This study also covered accent problems such as utterance speed which were very important for building a practical system. Table 2.2 shows the number of participants with reference to their age groups and gender.

Age	-10	20	30	40	50	60	70	80+	UN
Speakers	770	2569	2481	1104	760	300	68	14	774
%	8.7	29.2	28.2	12.5	8.6	3.4	0.8	0.2	
Male	272	948	1115	446	253	134	28	7	
%	3.1	10.8	12.7	5.1	2.9	1.5	0.3	0.1	
Female	498	1621	1366	658	507	166	40	7	
%	5.7	18.4	15.1	7.5	5.8	1.9	0.5	0.1	

Table 2-2: Voices across Japan Participants [22].

A speech technology group in Spain has built a recognizer covering the different dialects of Spanish from Spain and America and was based on recognizing numbers. In Spain, there are a great number of dialectal variations and that is one of the reasons why this study was undertaken. The researchers were actually interested in a system that would handle a large number of users of different regions. As discussed by to de La Torre et al. [12], to accomplish such interest, they performed a test on the possibilities of Maximum a-Posteriori (MAP) to adapt to the original HMMs with the purpose of finding out whether the speech recognizer will manage the dialectal variations.

Evaluation of the recognizer was performed to test whether it can handle the dialectal variations and what performance it would provide in the different cases. The study comprised of two phases, the first phase was to test how the recognizer would behave when the Spanish dialects were incorporated and the second phase was to test the Spanish dialects of Argentina [12]. Table 2.3 shows the record of word and sentence error rates encountered by the baseline system for the three best candidates.

	1st cand.	2nd cand.	3rd cand.
Word Error Rate	2.1%	1.7%	1.4%
Sentence Error Rate	9.6%	5.7%	4.4%

Table 2-3: Baseline System [12].

For the dialects in Spain, a database called VESTEL was established. The database was made up of a large number of speakers throughout Spain and it covered all dialects of Castillian Spanish. The division of dialects in Spain was based on the phonetical and pronunciation similarities. The prompts sheets included questions such as

1. Where were you born?
2. Where do you live [12]?

to keep track of which speaker falls under which dialectal group. Evaluation was performed based on the 7 regions of Spain and Table 2.4 shows the results obtained.

Region	1st cand.	2nd cand	3rd cand
Castile	9.8%	6.8%	5.5%
Catalonia-Valencia-Balearic Isles-Aragon	6.9%	3.5%	3.0%
Extremadura	11.1%	4.4%	2.2%
Basque Country-Navarre	10.6%	4.1%	3.3%
Galicia-Asturias	12.7%	7.9%	5.5%
Andalusia-Murcia	10.2%	5.1%	4.2%
Canary Isles	14.29%	9.52%	9.52%

Table 2-4: Results of the Spanish dialectal regions [12].

The Argentinean Spanish dialects were also evaluated; a total of 1181 pronunciations corresponding to natural pronunciation of telephone numbers were obtained [12]. Table 2.5 displays the results.

	1st cand.	2nd cand.	3rd cand.
Word Error Rate	2.7%	1.7%	1.5%
Sentence Error Rate	10.4%	5.2%	4.0%

Table 2-5: Argentinean-Spanish rates [12].

de la Torre et al. [12] define 1st cand, 2nd cand and 3rd cand as the indication of the three best candidates who participated in the research project. The detailed word error rates and sentence error rates for the three best candidates are compiled in Tables 2.3, 2.4 and 2.5 respectively.

After obtaining poor results (mainly brought by the variations with respect to the dialects), the researchers planned to improve the results by deploying two adaptative techniques called Maximum Likelihood (ML) and MAP [12]. The ML technique estimates the linear transformations and reduces mismatches in the data while MAP is an estimate generated by the acoustic models and is also used to adapt models [38]. And of course, a considerable improvement in performance with respect to the three systems was obtained and is shown in Table 2.6.

In conclusion, it was discovered that in most Spanish dialects, the recognition results nearly equals the results obtained for the Castillian Spanish.

System	Argentinian		Castilian	
	WER	SER	WER	SER
Baseline	0.99%	5.16%	0.95%	5.48%
ML	0.83%	4.10%	2.50%	14.8%
MAP	0.87%	4.20%	1.08%	6.66%

Table 2-6: Results of the three methods after having applied the adaptative techniques for re-estimation of the model for a word "cero" [12].

2.4 Speech technology applications

Any other task that entails interaction with a computer can utilize speech technology procedures. The majority of the speech technology applications are currently deployed and contracted to companies. Despite the few applications areas mentioned hereafter,

there are other existing application areas where speech technology can be very useful, namely, the business, telecommunications industries and the medical field.

2.4.1 Conventional Speech Applications

Initially, speech applications developed were the IVR systems which used speech outputs and telephone keys for interaction. This is one widely used application but still, problems were encountered, such as awkwardness in the Dual-Tone Multi-Frequency (DTMF) interface which resulted in customers not being satisfied. But at a later stage, the DTMF inputs were replaced by speech inputs [36]. Cost savings were no longer an issue because users became more satisfied.

Presently the speech application research basis have amounted to mainly telephony applications and are fairly good, include state-of-the-art recognizers, natural understanding and response generation components and utilize advanced techniques for dialogue management [36]. Even today, the most popular key to developing successful applications is integration. Abara and Wang [5] mention the following as the examples of conventional applications:

2.4.1.1 Dictation

Dictation is one of the speech technology applications that was successfully developed and is utilized after strong innovations of ASR technology. Dictation application is another type of speech technology application which enables users to interact with computers by using authoritative orders. For example, the users can utilize speech instead of typing.

2.4.1.2 Command and Control

This is also the common speech technology application area that is currently being used to ease our work by verbally addressing equipment to perform the required tasks/actions. For example, the user may verbally say to the computer "shut down" and the computer would turn off.

2.5 Summary

In this chapter we have discussed different categories of speech technology namely, speech recognition and speech synthesis. Other categories exist such as speech coding but have not been included for discussion in this chapter.

We have also reviewed the history of speech technology and illustrated with a brief timeline, which showed that this research started as early as 1936 and was initiated by the AT & T's Bell Labs. Since then the research has progressed greatly and today speech recognition systems of different languages and dialects in different fields are being installed and are utilized successfully. A brief overview of research on speech dialects that was conducted in South Africa, Japan and Spain is also included.

We also presented the speech application types and their environments in which the applications can be built on. Other environments in which the speech applications can be built, include areas such as business and the medical field.

In the next chapter, we will cover the discussion on the theoretical part of developing and training a speech recognizer and an outline of what an HTK toolkit is.

CHAPTER 3: RECOGNITION FRAMEWORK

3.1 Introduction

This chapter discusses the theoretical background of building a typical speech recognizer. It also covers the different speech recognition approaches, how the Hidden Markov Model Toolkit (HTK) works and the general challenges that each ASR system encounters.

3.2 Topology of ASR

Automatic speech recognition can be characterized into several different types of classes as shown in the Table 3.1.

Vocabulary and language			
Vocabulary Size	small	middle-size	(very) large
Grammar	phrases	Context Free Grammar	n-gram
Extensibility	fixed	changeable	dynamic
Communication style			
Speaker	dependent	adaptive	independent
Speaking style	discrete	continuous	spontaneous
Overlap	half-duplex	barge-in	full-duplex
Usage conditions			
Environment	clean	normal	hostile
Channel quality	high-quality	normal-quality	low-quality

Table 3-1: Properties of speech recognition systems (adapted from [36]).

3.2.1 Communication Style

Speaker-dependent systems are designed to work within a certain environment and they accommodate only a single speaker. This type of a system can generally recognize a small set of words and can be very accurate [5]. The training part of this system depends on a single speaker such that during the testing phase, the utterance consistency should be maintained to yield accurate results. The speaker-dependent system thus depends on the speaking patterns of a single individual.

With speaker-independent systems, attention with respect to recognition and training is drawn to a wide population of users, i.e., they are trained and tested by different speakers and they are designed to be used by many people from different regions. Speaker-adaptive method is a method used for the extension of an existing speaker-independent system. That is, the speaker-independent system is built first and can later be improved by deploying adoption methods. Among other types of recognition systems, this kind of a system is the best since it provides more flexibility.

The way in which the users speak and the way in which their utterances are recognized also have an impact on the system. The speaking style in Table 3.1 ranges from discrete, continuous to spontaneous speech. We should bear in mind that the earliest systems designed were based on isolated words.

Discrete recognizers

Isolated word recognizers require users to make explicit pauses when uttering a sequence of words/sentences to the system. The speech recognition process in this regard is conducted in a manner that only word at a time is recognized [5].

Connected word recognizers

Connected word recognizer is an improvement of the latter type of a speech recognizer because words to be recognized are concatenated one after another, not necessarily with clear pauses in between, unlike the isolated word recognition systems which strictly cannot recognize any sentence without pauses. Discrete and connected word recognizers are both similar in the sense that they analyze a string of words spoken together, but not at a normal rate.

Continuous speech recognizers

Continuous speech recognizers accommodate continuous utterances at a time. They do not require explicit pauses between words at all and allows normal conversational speech which is the goal of many ASR systems today. It can serve as a real world machine that allows clear man-machine interaction without any explicit pauses.

Spontaneous speech recognizers

Spontaneous speech recognizers are systems based on natural sounding speech without practice in preparation for recognition [5]. They allow users to communicate with them in a more natural way. Such systems should possess the capability of effectively working under any condition such as an utterance starting with or including a non-speech event.

Other properties such as overlapping focus on the capability of the individual system developed. This property specifies whether the system is capable to handle multiple communication channels simultaneously or not. With the full-duplex capability, the communication is handled simultaneously in different directions, i.e., it supports the user and the computer to speak to each other at the same time. The half-duplex overlapping forces the grammar to be designed in a way that supports strictly turn-based communication. The barge-in is similar to the half-duplex but it is more reliable [36].

3.2.2 Vocabulary size and Language

By vocabulary size, we refer to the number of words and sentences that can be incorporated within a speech recognition system. This property also plays a vital role during data collection to keep track of the number of words needed to establish sufficient pool of speech data for a specific recognizer. For example, suppose we want to build a small vocabulary speech recognizer, we will only collect few speech samples with tens of words. According to Abara and Wang [5], the vocabulary range varies from small to very large as described below.

- Small Vocabulary – tens of words
- Medium Vocabulary – hundreds of words

- Large Vocabulary – thousands of words
- Very large – tens of thousands of words.

The grammars are of importance to the development of the speech recognition systems. A grammar contains a set of rules that define the structure of a language to be used for a particular system and to establish a tight relationship between the developed system and the language used to develop that system. An n-gram language model has an option of being engaged in very large vocabularies. They are basically used in dictation and any other task incorporated in very large vocabularies [36].

Abara and Wang [5] state that for increase in accuracy and computation reduction, another method called Context Free Grammars (CFG) is deployed. It puts a limitation to the vocabulary and syntax structure of speech recognition to those words and sentences that can be used in the current state of an application. The text file for our Northern Sotho dialect-based speech recognizer might look like this,

<Start> = (<$Word>) | (Exit application)

Word = Amogela | Batho | Ka | Ntlong | Ya | Morena | Phepheng;

This grammar translates into eight possible words:

Amogela

Batho

Ka

Ntlong

Ya

Morena

Phepheng

Exit application.

The CFG increases the recognition accuracy by minimizing the words to be recognized such that they correspond strictly to the defined grammar. The recognizer will only be able to work with words defined within the grammar; otherwise any other word would be considered as an out of vocabulary (OOV) word and will not be recognized.

3.2.3 Preferred environmental conditions

The environment, in which an ASR system operates, ranges from clean to hostile and from high to low condition quality channels. The low quality channels include a telephone line which is utilized in a very noisy place or public area. Normal quality condition channels can be in shared offices and homes where a form of input is a desktop microphone. The quality condition channel which is completely accurate is called the high quality channel which may include close-talk microphones in a single-person office or an office line operated in a single-person office [36]. More accurate speech recognizers are in most cases built on environments that are very clean and utilizing the data that is recorded from the clean environment is a great advantage to the generation of good results.

3.3 Speech recognition approaches

Automatic speech recognizers are developed utilizing a number of different approaches. These are the Acoustic-phonetic, Statistical Pattern recognition and Artificial intelligence approach.

3.3.1 Acoustic-Phonetic approach

The theory behind acoustic-phonetic approach assumes that the input utterances is divided into finite, distinctive phonetic units and are also characterized by a set of attributes of the speech sample over time [29].

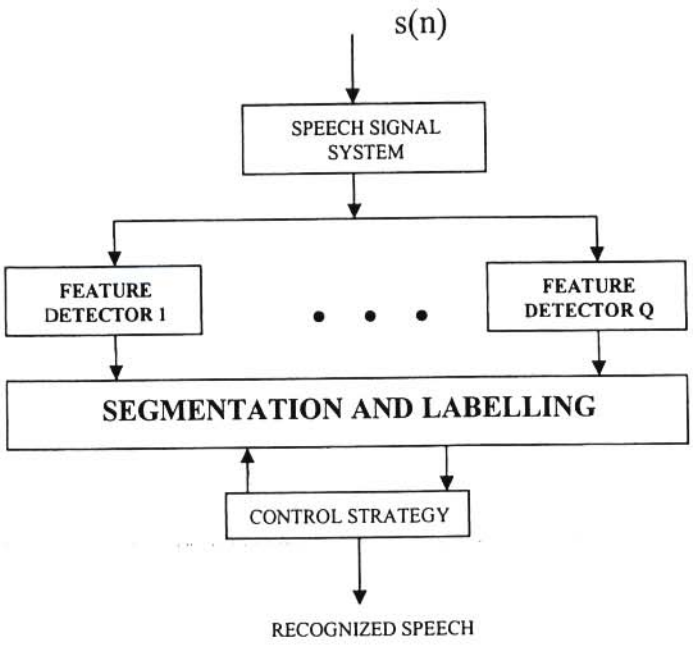

Figure 3-1: Phases of acoustic-phonetic recognition system [29].

As discussed by Rabiner and Juang [29], this approach divides the process of speech recognition into two steps namely:

(i) Segmentation: The speech signal which represents a sentence, a word or any other utterance is partitioned into discrete segments whereby each of the segments corresponds to its defined phonetic set. Then, since each segment has a corresponding phonetic unit, an appropriate phonetic unit is attached to it according to the phonetic properties of the segment.

(ii) Validation: The second step tries to formulate and validate a word from the obtained sequence of phonetic units.

For example: Suppose we have a representation of a Northern Sotho word "amogelang" in the corresponding phonetic format.

$$sil - a - m - o - g - e - l - a - ng - sil$$

The symbol ***sil*** indicates a silence and the rest of the symbols are phonetic representation of a wave form "amogelang".

Figure 3.1 illustrates with a drawing the acoustic-phonetic approach to speech recognition. Firstly, the speech signal is analyzed as in any speech recognition system

and appropriate time-varying characteristics of that wave representation are noticed and detected. Some of the important aspects considered for speech recognition include presence or absence of nasal resonance, presence or absence of random excitation, formant location which is frequencies of the first three resonances, voiced-unvoiced classification and ratios of high- and low-frequency energy [29].

The segmentation and labeling part produce a structure of phonemes. Finally the output of the speech recognizer is a word or a sequence of words forming sentences. The important fact about this approach is that it requires explicit knowledge of phonetic units before a match can be obtained [29].

3.3.2 Statistical Pattern Recognition approach

The statistical pattern-recognition approach compares the speech patterns stored in the system during training "which serves as prerequisite, but unlike the latter approach where phonetic units are stored before" to ones that are currently uttered by the end-users. That is, the speech patterns are first recorded into the system before the development, and during testing; the input utterances are matched against the patterns in the system. For accurate performance, a massive number of patterns should be input into the system. This approach is the simplest and best since it can be easily used to develop some speech recognizers and has been scientifically proven to produce high recognition results [29]. Figure 3.2 gives an illustration of a pattern based recognizer.

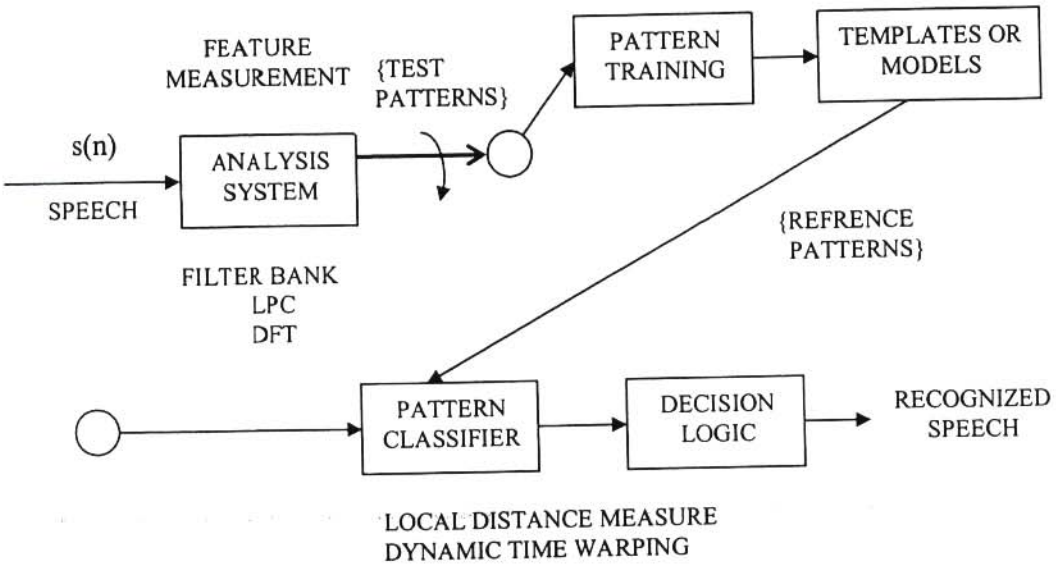

Figure 3-2: Phases of pattern recognition system [29].

In pattern recognition there are basically four steps. The initial step as in the acoustic-phonetic approach is the analysis of the speech signal. Then follows the pattern training, whereby speech patterns corresponding to waves are used as pattern representatives of the features of that wave [29]. Comparison of patterns takes place and they are later classified. A decision on the best match is taken by component decision logic and recognized expression is observed.

3.3.3 Artificial Intelligence approach

In the artificial intelligence approach, knowledge about that specific recognition environment from a large number of knowledge sources is captured, for improvement of the recognition rate [29]. Categorization is shown in the following Figure 3.3.

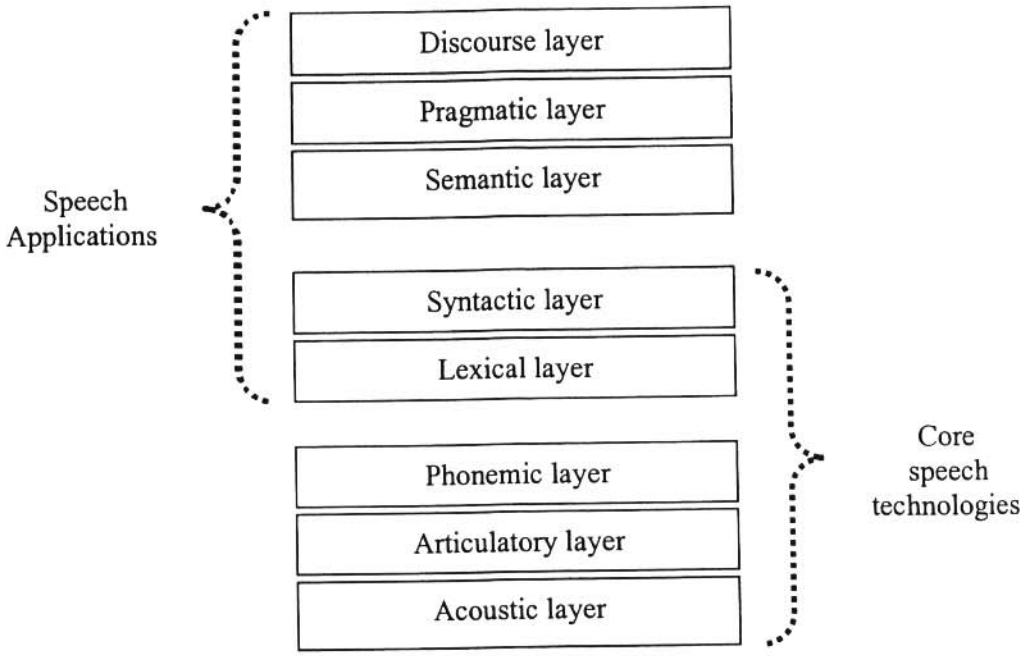

Figure 3-3: Different layers of knowledge sources [36].

The focus in the artificial intelligence approach is on the five layers described by Lau [23] & Rabiner and Juang [29] respectively.

- *Lexical knowledge* – the position of a phone within a word can greatly influence its duration. For example, stressed vowels are much longer than unstressed vowels and consonants in pre-stressed positions are often longer than those in unstressed and post-stressed positions.
- *Syntactic knowledge* – the phrasal pattern of a sentence often affects phone durations. For example, vowels in syllables preceding phrase boundaries are often longer than those in non-phrase final syllables. Also segments preceding a pause are often lengthened.
- *Acoustic knowledge* – evidence of words uttered is based on the spectral measurements and absence or presence of features.
- *Semantic knowledge* – involves validating the meaning of phrases, words or sentences for consistency purposes.
- *Pragmatic knowledge* – takes care of the language usage in the context of ambiguity.

The incorporation of the above mentioned knowledge sources can be done in a number of ways. A much broader discussion on the following approaches below is also covered by Rabiner and Juang [29]:

i. Bottom-up approach: In this approach, the feature extraction, phonetic decoding are initial steps. Figure 3.4 displays the process.

ii. Top-down approach: This is more or less the opposite of the latter approach since it starts from higher levels to lower levels. That is, the language model generates word hypotheses that are matched against the speech signal which builds semantically correct sentences.

iii. Blackboard approach: Is an important approach that considers all knowledge sources independently and data-driven.

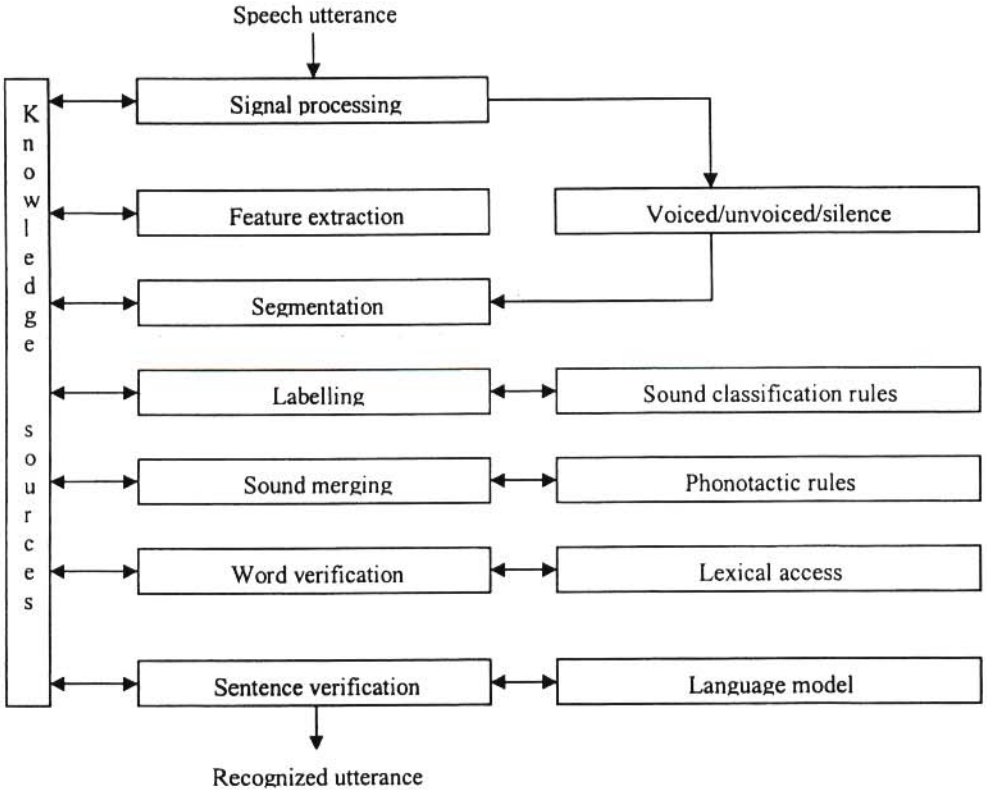

Figure 3-4: A bottom-up approach to artificial intelligence (adapted from [29]).

3.4 The Hidden Markov Model Toolkit (HTK)

The HTK toolkit is a statically-based toolkit that can be used as a graphical user interface among speech applications and has been scientifically proven to create and manipulate HMMs. It is a software product of the Cambridge University [2]. The HMMs are the main aspects forming the basis for the current state-of-the-art speech recognition systems [37]. There are other toolkits for the development of the speech recognizers such as Sphinx, which is also one of the reliable engines used for the development of speech recognizers.

The reason behind the selection of HTK as an engine to be used in this project is that it is freely available on the internet and also runs well on windows platform – our preferred platform. The HTK toolkit also handles the process of feature extraction from speech samples which is the important factor in speech recognition.

The HTK toolkit has a number of library modules and tools which serve as the important requirements for the engine to run effectively. The tools encapsulated within the HTK also provide complicated facilities for the analysis of speech inputs, HMM training, testing and results analysis [2]. Both tools and modules are mostly utilized by the engine when commands are invoked to perform specific tasks such re-estimation of the HMM models.

3.4.1 The hidden Markov models

An HMM is a statistical model of a class of samples which can be utilized to estimate the probability that a given input sequence belongs to the class on which it was trained. It is a method that is widely utilized to characterize the spectral properties of the frame of a pattern [20, 29]. The HMMs are basically characterized by the following aspects which were also discussed by Huang et al. [20],

- $O = \{O_1, O_2, \ldots, O_M\}$ which is an output observation that corresponds to output of the system being modeled.
- $\Omega = \{1,2,\ldots,N\}$ which is a set of states (nodes) representing the state space.
- $A = \{A_{ij}\}$ which is a transition probability matrix, where A_{ij} is the probability of A taking a transition from node i to node j.

- B = $\{B_i(k)\}$ which is an output probability matrix.
- $\pi = \{\pi_i\}$ which is an initial state distribution.

There exists a stationary finite state called Markov chain which is a simple class of processes from which more complex processes can be derived. That is, the HMMs do belong to this class of processes [6]. A Markov chain comes into existence if the outputs of observation alphabets of the HMMs are ignored [28]; it is actually a set of states and associated probabilities of transitions between states [20]. Consider a five state Markov generation as shown in Figure 3.5.

The transition from state 4 to state 5 are also based on probabilities and can be governed by the discrete probability a_{45} for example. For the Markov model, the first and the last node in the chain are called non-emitting states while the three in the middle are called emitting states.

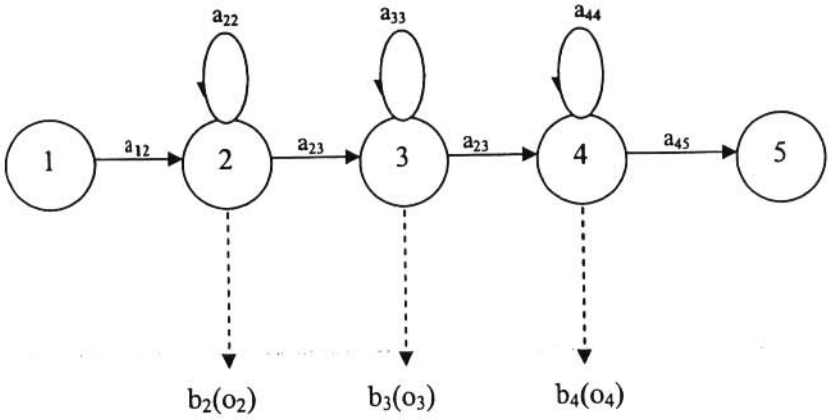

Figure 3-5: Markov generation models [38].

Roughly what transpires in Figure 3.5 is that the input, speech signals that have already been pruned, are mapped against its corresponding developed HMM model. The probability of this input is matched against the master HMMs in the database. Thereafter if an HMM match exists, then the output which is sometimes called an observation in this regard, is produced. Efficient training on the HMMs makes the whole system more reliable and accurate. The more we add words or phrases in the database, the more accurate our recognizer will be.

3.5 Speech Models

The current state-of-the-art systems for ASR are based on the statistical approach of the Bayesian decision theory. The implementation of the Bayesian decision rule for an ASR system is based on two different kinds of stochastic models, the acoustic and the language modeling [33]. The two components are crucial during the development of ASR. But before we go into details of the two models, we first define what we mean by Bayesian decision rule.

3.5.1 Bayesian Rule

Bayesian decision rule is one of the rules utilized by the recognition systems during the testing step to make accurate, wise and meaningful decisions [16]. A discussion on the equation representing the Bayesian rule is derived from Dyer and Skrentny [16] which says:

$$P(words|signal) = P(words)P(signal|words)/P(signal) \quad \text{.....(3.1)}$$

For simplicity sake, since we are given a speech signal as an input, the main goal here is to find a sequence of words that would maximize *P(words|signal)*, where *P(signal)* is a constant for a given acoustic input. Then the term *P(signal)* can be simply dropped, and the equation simplifies to

$$argmax_{words} P(signal \mid words)P(words) \quad \text{.....(3.2)}$$

The *P(words)* represents the language model in that it extends the prior probability of a word string. The *P(signal|words)* is the acoustic model which specifies the probability of the acoustics given that a sequence of words was spoken [16].

The next sections 3.5.1.1, 3.5.1.2 and 3.5.1.3 outline the major algorithms associated with HMMs as outlined by Huang et al. [20].

3.5.1.1 Evaluation : The Forward Algorithm

The dilemma about this aspect is that, given a model and a sequence of observations as depicted in Figure 3.5, what is the probability that the model has generated the observations [20]? To solve this dilemma, the Forward Algorithm is deployed. As discussed by Rabiner and Juang [29], the forward algorithm computes the probability that the observations are derived from the model by summing up the probabilities of all allowed possible paths. In this algorithm, a time variance of the probabilities is utilized to reduce the complexity of the calculation.

3.5.1.2 Decoding : Viterbi Algorithm

The Viterbi algorithm efficiently solves the problem of the most likely state sequence in the model that produces the observation given a model and a sequence of observations [20]. The dilemma is solved by listing all possible sequences of states and finding the probability of the observed sequence for each of the combinations. This approach is possible, but finding the most probable sequences by exhaustively calculating each combination is computationally expensive.

3.5.1.3 Estimation of the HMM parameter: The learning problem

Given a model and a sequence of observations, how can a model parameter be used to maximize the joint probability? The Forward-Backward (also known as Baum-Welch) algorithm permits this estimates to be made based on a sequence of observations known to come from a given set that represents a known hidden set following a Markov model. The algorithm proceeds by making initial guesses of the parameters and refining it by assessing its worth and attempting to reduce the errors it provokes when fitted to the given data [20].

3.5.2 Language model

Language modeling is an important component of the speech recognition systems. The main purpose of the language model is to perfect the regularities of the language used to build a speech recognizer and it also helps a lot during the interpretation of a

particular acoustic input [29]. When word models are concatenated to form phrase models, the language model determines which word comes after the other [10].

A detailed description of the language model was also discussed by Dyer and Skrentny [16] where in Equation 3.1, the term The *P(words)* is the prior probability that a sequence of words *words* = $w_1\ w_2 \ldots w_n$, is more correct in the given natural language. The rule is as follows

$$P(w_1\ w_2 \ldots w_n) = P(w_1)P(w_2 \mid w_1) \ldots P(w_{n-1} \mid w_1, \ldots, w_{n-2})\ P(w_n \mid w_1, \ldots, w_{n-1}) \quad \ldots\ldots(3.3)$$

According to Dyer and Skrentny [16], computations should be made to confirm that the first word in the sentence is really w_1, and the word string coming after the latter should also be proven that it is w_2, given that the first word is truly w_1. The final part is the probability that the last word in the word string is w_n given that the words before it are

$$w_1 \ldots w_{n-1}. \quad \ldots\ldots(3.4)$$

The First-Order Markov assumption has the property that, the probability of a word is only dependent on the previous word. Therefore

$$P(w_n \mid w_1, \ldots, w_{n-1}) = P(w_n \mid w_{n-1}) \quad \ldots\ldots(3.5)$$

Computing the joint probability from the simplified expression yields

$$P(w_1\ w_2 \ldots w_n) = P(w_1)\ P(w_2 \mid w_1) \ldots P(w_n \mid w_{n-1}) \quad \ldots\ldots(3.6)$$

which is our final equation. The language model therefore serves as an important tool for the development of speech recognizers.

3.5.3 Acoustic Model

Acoustic modeling also contributes positively towards the development of speech recognizers. It plays an important role of improving the accuracy of the speech

recognizers. Extraction of acoustic parameters form the input utterances has been realized to be a concatenation of the processes of the HMM [9].

3.6 Speech Recognition

ASR systems are generally characterized by algorithms and modeling of grammar for the purpose of gaining improvement in performance of continuous speech. A simplified procedure of building a recognizer is shown in Figure 3.6 as an adaptation from [33].

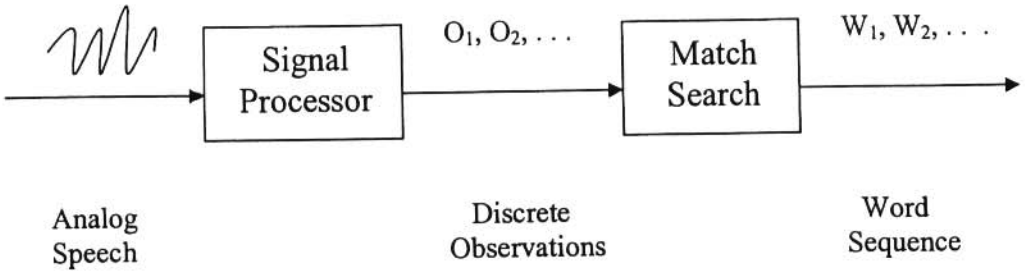

Figure 3-6: Simplified procedure of building a recognizer [33].

The simplified Figure 3.6 can be extended to a detailed process which consists of two phases namely, the training and the recognition phase, according to Doe [14]. The training phase examines a large variety of model parameters while the recognition phase searches through all possible alternative words to find an optimal match.

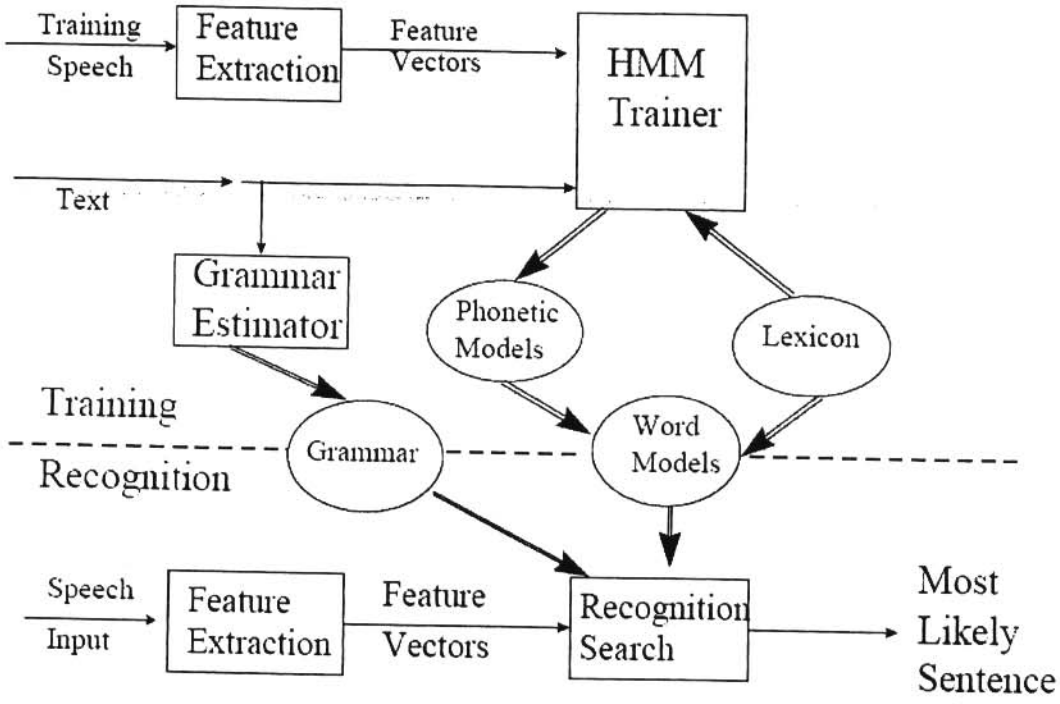

Figure 3-7: Detailed process for building a recognizer [14].

3.6.1 Training phase

As indicated earlier in section 3.6, there are two phases in building a recognizer, which are the training and recognition phases. Both stages possess the feature extraction step. Its aim is to parameterize the speech signal such that it is suitable for the ASR system [32]. That is, the raw speech data waves are parameterized into a sequence of feature vectors.

For telephone recordings, the acoustic features for speech recognition are based on short term Fourier phase spectrum. The new phase features are used together with standard Mel-frequency-cepstral-coefficients (MFCC) and results with Linear Discrimant Analysis (LDA) for choosing relevant features are presented [32].

Those feature vectors and their corresponding transcriptions are then used to train the HMMs, i.e. the acoustic models are trained and phonetic models are produced. The lexicon or pronunciation dictionary is defined to give valid word models for the

recognition. Before using the word models, we have to define the task grammar of the recognizer. Figure 3.7 shows the both the training and the recognition steps clearly.

3.6.2 Recognition phase

In the last phase, the performance of the recognizer is evaluated by being given an unknown speech signal, where the feature extraction is performed once again. Using the generated feature vectors, together with task grammar and the word models, recognized words or sentences are generated.

The decision about which sentence is most likely to occur depends on two models. The language model stores the linguistic information of a certain language which the system depends on, and provides a priori-probability of a word sequence whereas the acoustic model stores the acoustic properties of speech data and establishes the probability of the observed acoustic signal given a hypothesized word sequence [33].

An important part of the ASR system is the search component. We might even call it the heart of any ASR system. If this part fails to give results, then there is no speech recognition at all. In order to produce an output hypothesis, be it a string of words, or a sentence, a hefty chunk of work should be performed for better recognition results. The three algorithms described in sections 3.5.1.1 to 3.5.1.3 are usually implemented here for matching of utterances.

3.7 Challenges in ASR

Since the development of speech recognizers moves at slower but sure pace, researchers are really hoping for the best to come up with more efficient, robust speech recognizers which would behave like a normal human being. Such systems would be of high quality and would be more reliable for utilization for customer satisfaction.

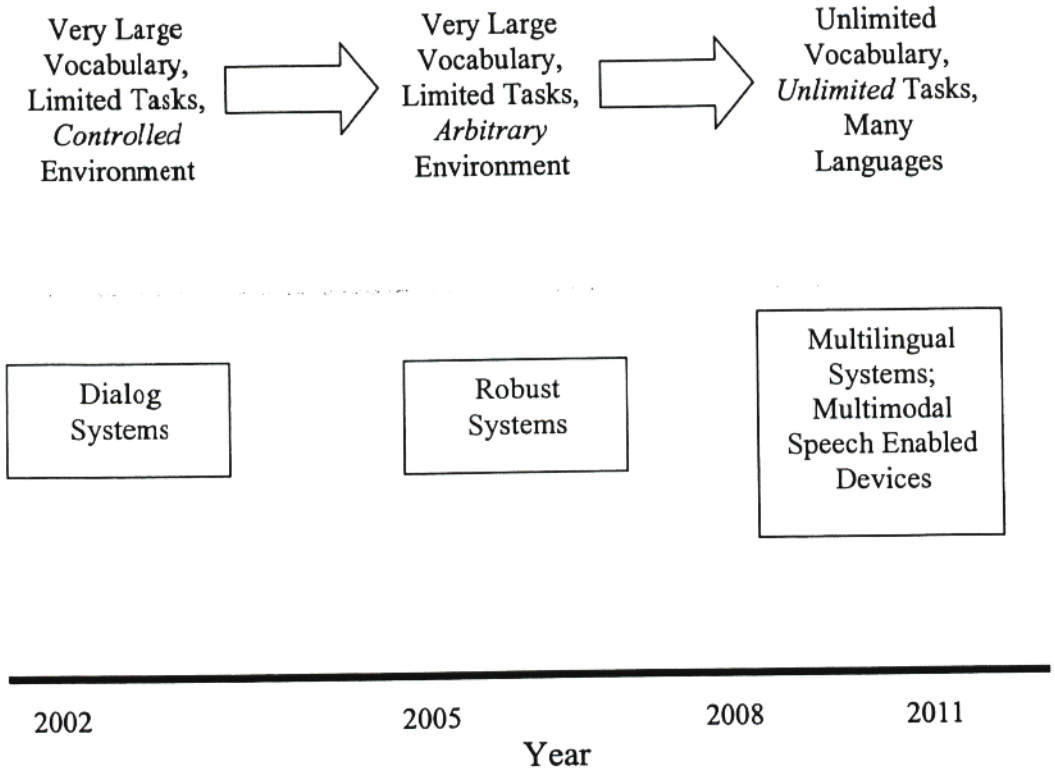

Figure 3-8: Future challenges of speech recognition technologies [27].

The researchers initiated this research on speech technology by an isolated word recognition system with a very small vocabulary which was based on acoustic phonetics. Nowadays, researchers have developed robust systems with very large vocabularies and are based on limited tasks and built to accommodate arbitrary environments. The Figure 3.8 according to Rabiner [27] shows what systems the researchers have already covered. As major challenges, multilingual systems with unlimited vocabularies and performing unlimited tasks should exist.

3.8 Summary

This chapter covered the different factors that characterize a speech recognizer such as the size of the vocabulary and communication style. The three speech approaches (Acoustic-phonetic, Statistical pattern recognition and Artificial intelligence) are also discussed in detail. The various ways in which speech samples were modeled was also

included and the chapter ended up by discussing the different phases one should go through when building a recognizer.

CHAPTER 4: SPEECH DATA COLLECTION PROCESS

4.1 Introduction

In this chapter we discuss how speech data was collected, taking into consideration the three selected dialects of Northern Sotho. The selected dialects are Selobedu, Setlokwa and Sepedi - the latter serves as the main and dominating dialect in the Limpopo province in terms of educational presentations and utility. The aspect which is stressed in this process of data collection is the variation of pronunciation with reference to certain words used in the training and testing of an ASR.

The variation of word pronunciation is strictly attributable to the regional accent and dialects. Though our main focus with respect to this research project is on the language dialects per se; gender, vocal differences, age, etc., are also considered in this pronunciation variation. With dialects, in most cases, the variation is brought by the positioning of a consonant or a vowel in a word, i.e., the occurrence of such can be at the beginning of a word [serving as a prefix], middle and/or at the end [serving as a suffix] of a word.

4.2 Dialects in Northern Sotho

This section covers briefly the historical background on the origin of the above selected dialects. The description is later clarified by stating the geographical regions indicating the areas with major prevalence of these dialects on a map. Obviously all of the dialects converge to the standard language called Sesotho sa Leboa, also known as Northern Sotho. For educational and literacy purposes, the reading and writing of the Setlokwa and Selobedu dialects is used and presented using a standard Northern Sotho orthography.

4.2.1 History of dialects in Limpopo Province

The most spoken and dominating language in the Limpopo Province is generally Sesotho sa Leboa with regard to population [26]. This language has deviated a bit

from other groups of Sotho languages such as Southern Sotho and Setswana. The Sepedi speaking population group covers a number of areas including Middleburg, Groblersdal, Lydenburg, Springs to the western side of the Limpopo province towards the border of Botswana. As Mokgokong [26] puts it, the Sepedi language serves as a standard dialect which can also be stretched and evaluated to yield other dialects as follows:

i. Central Sotho which covers the **Pedi**, Tau, Kone, Kopa tribes.
ii. Eastern Sotho which accommodates the Kuretse, Pai and Pulana tribes respectively.
iii. North-Eastern Sotho which accommodates Phalaborwa, **Lobedu**, Mamabolo, Letswalo, Mametša, Mahlo and lastly the Kgaga tribes.
iv. Northern Sotho which also covers Mphahlele, Tšhwene, Mathabatha, Maja, Mothapo, Matlala, Molepo, **Tlokwa**, Dikgale, Moletši and the Hananwa tribes.

From this stretched tree of dialect clusters, only three dialects which have strong ties with various other dialects were considered for this research project. The main reason behind the selection of the dialects was that Selobedu and Setlokwa were observed as the major dialects amongst the others which show the greatest dissimilarity to the main dialect Sepedi.

The dialect from the central part is Pedi (Sepedi), the one from the North-Eastern side of the province is Lobedu (Selobedu) and lastly the Tlokwa dialect (Setlokwa) which originated from the Northern part of the province [26]. It has been mentioned earlier that the difference between dialects generally is caused by the location of a vowel or a consonant in a word. As an example, Table 4.1 shows some of these differences.

Phone Occurrence	Phone	Dialects			
		English	Sepedi	Setlokwa	Selobedu
Locative suffix	/ng/	Grass	Bjang	Bjane	Bjanye
Causative suffix	/š/	Clothe	Apeša	Apesa	Apesa

Table 4-1: Example of phoneme occurrences in words.

In the above table, occurrences of other phonemes have not been covered. The occurrence of the phoneme /ng/ in Setlokwa changes to /n/ - /e/ while with Selobedu it changes to /ny/ both at the end of a word. Also the phoneme /š/ at word end changes to /s/ both in Selobedu and Setlokwa. However, this occurrence usually applies to Setlokwa. In general, the difference of most words pronunciations in Setlokwa is observed at the end of a word.

4.2.2 Geographical regions

The geographical regions from which the participants come, also plays a very important role in the appropriate location of our process of telephone speech data collection. The pronunciation variants are affected by the geographical areas of the participants.

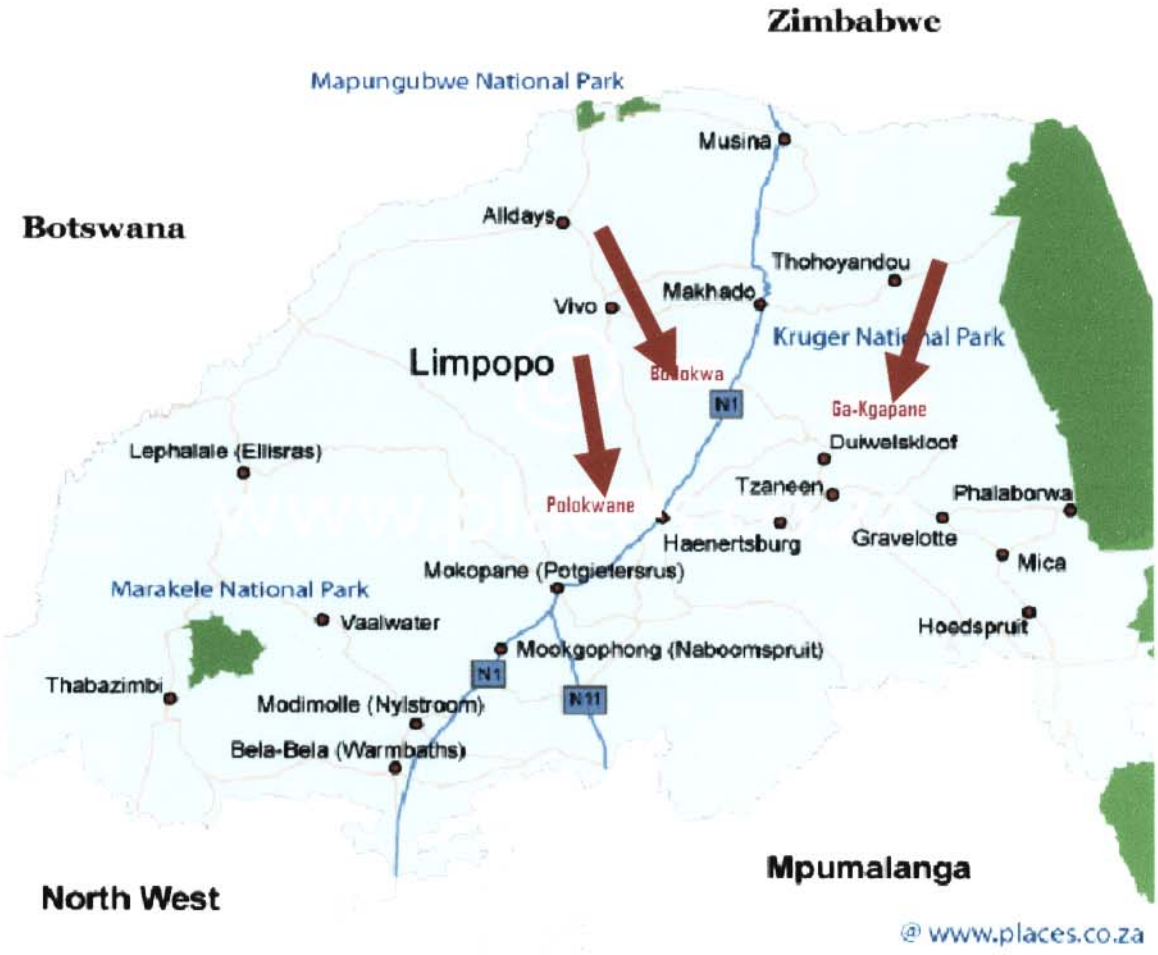

Figure 4-1: Geographical places considered during data collection [3].

The geographical regions with the red arrow on top and their neighboring places are the ones which were visited for the recruitment of volunteering speakers for our speech data collection.

4.3 Collection of telephone speech data

The speech corpus development was carried out telephonically with an integrated services digital network (ISDN) toll-free line installed to keep track of responses from the participants. That is, the initial recordings of the telephone speech prompts were performed successfully and later the toll-free line was activated to accept inputs/responses or voices. Almost all the participants have been people from the highlighted regions in Figure 4.1 - only mother-tongue speakers were considered for the production of quality speech data. Volunteers were given a toll-free number (indicated on the prompt sheet) for the donation of voice samples.

4.3.1 Prompt sheets

A total number of 150 different prompt sheets were designed and distributed to recruited respondents. The number was divided amongst the three dialects which yielded 50 prompt sheets for each dialect. The prompt sheets consisted of a set of instructions to go through before making a call, a number of questions to be answered (prompted spontaneous questions) and items to be read.

The questions to be answered included predominantly yes/no questions, spontaneous dates, prompted time of the day, etc., while items to be read included phonetically rich sentences, isolated words and numerals. *Appendix C lists the number of phoneme occurrences in prompt sheets.* The sheet number was also included to keep track of donated voice responses, otherwise they will be mixed up and proper transcription would then become very difficult.

Table 4.2 shows the structure and contents of prompt sheets that were designed in order to meet the requirements of building a semi-robust automatic speech recognizer.

Sequence of isolated digits	1
Sheet number	1
Home language	1
Telephone number	1
Time of day	1
Full names	1
City of birth	1
Predominantly YES question	1
Predominantly NO question	1
Phonetically rich sentences	2
Phonetically rich words	3
Phonetically rich phrases	3
Gender	1
Age	1
Profession/occupation	1
Natural number	1

Table 4-2: Prompt sheet design.

4.3.2 Distribution methods

Occasionally prompt sheets for speech data collection were distributed around the surrounding areas of Mankweng township in the vicinity of the University of Limpopo. Flyers and posters were pasted around for people to volunteer in donating their voices. For people around the regions, Botlokwa and Bolobedu, telephonical contact was made to set up arrangements for a presentation on how the whole process should be handled. A major problem in distribution amongst the two dialect regions, Bolobedu and Botlokwa, occurred. The number of the prompt sheets which were distributed did not correspond with the number of the returned prompt sheets because recruited candidates did not return all the sheets as expected.

In a way of attempting to solve this major problem, distribution of prompt sheets was repeated after a certain period of time; i.e., we kept track of the speech data already captured and decided whether to distribute more sheets or not, based on the number of responses received.

Dialects	Gender		Total
	Female	Male	
SELOBEDU	25	25	50
SETLOKWA	25	25	50
SEPEDI	25	25	50

Table 4-3: Distribution of prompt sheets.

The projected distribution of prompt sheets is indicated in Table 4.3. The environment in which distribution took place includes government schools, radio-stations, public places and related work-groups. Only people who responded well to the telephone prompts were given a token of appreciation. The participants were verbally assured that the information given by the respondents was going to be kept confidential and private and used for speech technology research purposes only.

4.3.3 Classification of Respondents

Of importance was the specification of respondents according to their age and gender. A balanced number of sheets were distributed among participants but, in return, a semi-balanced number of responses were obtained. Table 4.4 shows the classification of participants and how the sheets were distributed across each dialectal area. NOTE: Table 4.4 applies to all the three dialects.

Age Group	Female	Male
16 – 30	15	15
31 – 45	7	7
46 - 60	3	3

Table 4-4: Distribution scale amongst the respective age groups

The final number of prompts sheets returned is shown in Table 4.5.

Dialect	No of sheets returned
SELOBEDU	39
SETLOKWA	45
SEPEDI	50

Table 4-5: Returned prompt sheets.

The respondents called from different physical environments, i.e., noisy and quiet environments. The noisy environments are places such as public places and busy streets while the quiet environments included single-person office, home, etc., The majority of the responses were from land lines and a very small percentage was from cellular phones. The main reason that caused the small percentage on cellular phones is because of standard charges which applied when calling the toll-free number.

4.3.4 Validated data

The conditions and/or features mentioned in section 4.3.3 affected the telephone speech capturing and the number of the speech samples validated came down to 30 calls per dialect totaling 90 calls for the whole project. This number of validated speech samples was calculated according to the minimum number of responses required per call out of a possible maximum of 21 utterances. If the participant answered 15 or more questions (71%) correctly and in a quiet environment, that

particular call is considered successful. The total number of speech samples validated per dialect is shown in Table 4.6.

Dialect	No of speech files
SELOBEDU	551
SETLOKWA	557
SEPEDI	581

Table 4-6: Validated utterances per dialect.

The overall total number of utterances used for this project was 1689 speech samples. The sampling rate utilized for these speech samples was 8 KHz, 16 bit on a mono-channel. The recording of the speech data was done on a personal computer equipped with a Dialogic card situated at the third party company Molo Afrika Speech Technologies (Pty) Ltd situated in Pretoria. The manual transcription of speech files was later verified and double checked for correctness. On the recorded speech samples, a large number of non-speech events was present and was cleared during this manual transcription to produce clean speech samples for training of the system.

4.4 Phonemes

One of the main issues that is of vital importance during the establishment and training of a speech recognizer is the phone set. This is an orthographically approved set of vowels and consonants in a particular natural language. *Appendix C* depicts the Northern Sotho phone set.

For the phone set defined in *Appendix C*, there are also phonological rules or guidelines which describe explicitly the occurrences of each phone with respect the human mouth and the manner in which each is articulated.

4.4.1 Phoneme ambiguity

The way in which the speaker presents utterances can be associated with the differences of the pronunciation of words. This, in turn, leads to different representation of phonemes with respect to the dialects. This means that the whole process of dialects is significantly conveyed by the differences in specific phoneme pronunciation. Table 4.7 shows the phoneme differences in pronunciation according to the respective dialects under consideration.

SEPEDI	SELOBEDU	SETLOKWA
/s/	/kh/	
/h/	/kh/	
/g/	Removed	/h/
/j/	/dy/	/dy/
/kg/	/kh/	
/ng/	/ny/	/n/ - /e/
/p/	/bh/	
/š/	/s/	/s/
/t/	/dh/	
/tl/	/dlh/	/tld/
/ts/	/dz/	

Table 4-7: Phoneme ambiguity representation.

Appendix D clarifies phoneme ambiguity by showing ambiguity representation using words. The differences in pronunciations of phonemes of Table 4.7 can be at the beginning of a word, the middle and the end of the word.

4.5 Lexicon construction

For a speech recognizer to perform better and recognize words correctly, it has to clearly consult with the dictionary to perform the look-up for the words to be

recognized. Without the pronunciation dictionary, not much would actually happen in automatic speech recognition. The creation of a pronunciation lexicon is therefore of vital importance to the training of a speech recognizer.

A pronunciation lexicon is a file which contains a sorted list of words mapped to their phoneme representations [17]. The lexicon which covered the three dialects had to be designed in a way that is biased to those selected dialects, i.e., all the pronunciation variants of the dialects were added and captured to an existing pronunciation lexicon.

The addition of the pronunciation variants was basically on phonemes which resulted in the mapping of single words to multiple pronunciations (phoneme representations). The main purpose of this kind of representation is to accommodate the three dialects. For example, the word *SESEPE* may have multiple pronunciations such that

SESEPE	*KH E S E B H E*
SESEPE (2)	*S E S E P E*

and *BJANG*:

BJANG	*BJ A NG*
BJANG (2)	*BJ A NY E*
BJANG (3)	*BJ A N E*

Thus the overall coverage of various pronunciation variants is taken care of in this way. Hua and Schultz [19] used this method as an initial attempt; the described method worked but not convincingly. Another method was deployed; i.e., a triphone state tying tree was designed. The relationship between the two methods is that they both try to solve the problem of constructing a pronunciation lexicon that accommodates the selected dialects. This method tries to define all the possible occurrences of phonemes in a word (begin, middle, end). Better results were then obtained through the use of the state tying tree method.

For this research work, we will try both methods and observe the performance of the system. Since we are building a recognizer using 43 phonemes, a total number of 129 decision trees should exist for the clustering method, which is not an easy task to perform. The stated number is determined by defining three occurrences of each of the 43 phonemes.

4.6 Summary

Various aspects which have to be considered before the collection of data, such as the design of speech prompts and prompt sheets were discussed in this chapter. The prompt sheets were designed in a manner which resulted in phonetically balanced data. Other precautions include gender and age grouping specifications which have been considered.

The balancing of phonemes during data collection plays a major role in training of the acoustic models. All the phonemes should be present for the training of the system. The geographical regions from which data was collected are also shown on a map together with the ambiguity of phonemes according to their respective dialects. This chapter also covers the manner in which the prompt sheets were distributed and lastly the way in which the pronunciation lexicon was constructed to accommodate the three dialects.

CHAPTER 5: EXPERIMENTATION

5.1 Introduction

In this chapter, the practical aspects of building a speech recognizer are discussed step-by-step. Building a dialect-sensitive speech recognizer can be a very complex process because it requires hard-working and dedication, although in practice it is actually straight-forward especially when building a speech recognizer that is dependent on only one language.

If the pronunciation lexicon is not correctly set up, the system would obviously mix the generated models up and thus resulting in an unexpected output. Most of the speech recognizers utilize huge amounts of speech data to produce good and near perfect recognition results.

5.2 Development Steps

Acquisition and installation of appropriate software packages is always a preliminary step which should be considered before running the actual engine, i.e., if it does not already exist. As such, a set of instructions on the installation of the HTK engine (the chosen ASR toolkit) should be followed before the actual recognizer development.

For the HTK toolkit to be installed and function successfully, Visual C++ 6.0 and a Perl compiler should be installed. As part of preliminary steps, the transcription files (files containing text representations of the corresponding wave files) together with a dictionary which spans a defined phoneme set should be set-up.

5.2.1 Task grammar and Dictionary

A task grammar is a component of the speech recognizer which clearly defines words and sentences that the system should comply with. Any other words or sentences which are not defined in the task grammar will be considered out-of-vocabulary and will not be recognized. The main purpose of the task grammar is to define a closed set

of words which the system should recognize. The words or sentences which are to be recognized should fall within the task grammar.

With regard to this research project, the task grammar was derived from the transcription files, i.e., the transcriptions were broken down into individual words. Amongst the individual words, some of the words had multiple occurrences. These words were aligned to be unique (removal of multiple occurrences of a word) and later sorted to create a set of words with alternatives, thus constituting what we call the task grammar.

An example of part of the task grammar utilized for this research work is shown in Figure 5.1.

```
$PHRASE = A
| AE
| AGELEDITXWE
| AGELEDITXWEGO
| ALAFXA
| AMOGELA
| AMOGELANG
| AOWA
| APANE
| APEILE
| APOLLA
| BA
| BAANEGWA
| BAFSA
| BAHLOLOGADI
| BAHUMI
| BAHWANA
| BAISANE
| BAKGONYANA
| BANA
| BANNA
| BARUTWANA
| BASADI
| BATAMELA
| BATHO
| BATSWADI
  . . .
(SENT-START <$PHRASE> SENT-END)
```

Figure 5-1: Part of a typical task grammar.

The word network which displays the statistics of the dictionary and task grammar itself is then created by invoking the following command, Young et al. [38]:

HParse gram wdnet.

The HParse module provides word patterns from the defined grammar. Suppose *gram* is the file name of the actual grammar shown in Figure 5.1. The main purpose of executing this command is to provide a clear list of word instances and word-to-word transitions.

The main Northern Sotho dictionary with standard pronunciations and with a one-to-one relationship is built (*see section 4.5*). By one-to-one relationship, we simply mean that a single word is mapped to a single pronunciation or phone level representation. That is, a lexicon with a single pronunciation is developed. Since this project accommodates three dialects of Northern Sotho, we should construct a dictionary that will incorporate all three dialects as well as the system at large.

A list containing a total number of 842 sorted unique words was initially compiled from the original transcriptions of the collected speech data. Amongst the 842 unique words, 283 words had single pronunciation, i.e., one-to-one relationship, 469 words had two pronunciations, i.e., one-to-two relationship and lastly a total number of 91 words had three pronunciations, i.e., one-to-three relationship.

By a single word pronunciation, we mean that one word is pronounced the same for all the selected dialects. For two word pronunciations, we may find that a word is pronounced differently by either one of the dialects, Selobedu or Setlokwa. For three word pronunciations, we mean that all the selected dialects pronounce a single word differently.

In total the pronunciation lexicon contained 1464 words and was later combined with an already existing Northern Sotho pronunciation lexicon and thus forming a massive dictionary with 5422 entries including duplicate words. This entails that the vocabulary is medium-sized.

A	A SP
ADI	A D I SP
AE	A A E SP
AE	A E SP
AGELEDITXWE	A G E L E D I TX W E SP
AGELEDITXWEGO	A G E L E D I TX W E G O SP
AGELEDITXWENG	A G E L E D I TX W E NG SP
ALAFIWA	A L A F I W A SP
ALAFXA	A L A FX A SP
AMOGELA	A M O E L A SP
AMOGELA	A M O G E L A SP
AMOGELA	A M O H E L A SP
AMOGELANG	A M O G E L A NG SP
AMOGELANG	A M O H E L A N E SP
AOWA	A O W A SP
APANE	A P A N E SP
APEILE	A B H E I L E SP
APEILE	A P E I L E SP
. . .	
etc	

Figure 5-2: Multiple pronunciation lexicon.

The various phonemes which account for the difference in phonemic representation of the selected dialects were determined. The representation of an extract of the pronunciation lexicon which was utilized for this research project is shown in Figure 5.2.

5.2.2 Generation of transcription files and training phones

From the original transcription file created manually, word level and phone level transcriptions are generated by running Perl scripts. The word level transcription is stated precisely by invoking a Perl script called *promts2mlf* which forms part of the HTK package. The phone level transcription is generated by invoking the HLEd component which manipulates the label files [38]. Examples of both the word level and phone level master label files (MLF) are shown in Figure 5.3 and Figure 5.4 respectively.

```
"*/LB01012.lab"
KE
METSOTSO
YE
MASOMEPEDI
LE
YE
MEHLANO
O
KWA
IRING
YA
LESOME
.
"*/LB01015.lab"
SEMPHETE
KE
GO
FETE
. . .
etc
```

Figure 5-3: Word level transcriptions.

```
"*/PD04019.lab"
SIL
S
E
B
O
P
E
G
O
S
E
S
E
M
PSH
A
S
A
S
E
G
O
SIL
.
. . .
```

Figure 5-4: Phone level transcriptions.

5.2.3 Feature Vectors

In terms of speech recognition, feature vectors simply mean the extraction of speech wave file to the format which is compatible with the selected toolkit used for developing a speech recognizer. In this step, the speech waveforms are parameterized and converted into a sequence of feature vectors called Mel-frequency-cepstral-coefficients (MFCC). The main reason for choosing MFCC is the fact that it is more compatible with the HTK toolkit and offers informative speech features.

Before the conversion, a configuration file with the correct parameters is defined. The main purpose of the configuration file is define parameters used to convert speech data samples into the corresponding feature vectors and can also be used to re-estimate the HMM models. The configuration file should also be compatible with the training data. Figure 5.5 shows the configuration file that was utilized.

```
# Coding parameters

ENORMALISE =        FALSE
NUMCEPS  =          12
CEPLIFTER =         22
NUMCHANS  =         26
PREEMCOEF  =        0.970000
USEHAMMING =        TRUE
WINDOWSIZE =        250000.000000
SAVECOMPRESSED =    TRUE
TARGETRATE  =       100000
TARGETFORMAT =      HTK
TARGETKIND  =       MFCC_D_A_0
SOURCERATE  =       1250
SOURCEFORMAT =      WAVEFORM
SOURCEKIND  =       WAVEFORM
HEADERSIZE  =       1020
SAVEWITHCRC =       TRUE
```

Figure 5-5: Configuration file.

The parameters of the configuration file are discussed by Young et al. [38] as follows:

- Enormalise: normalizes the speech samples with energy.
- Numceps: specifies the number of MFCC to be produced.
- Ceplifter: specifies the number of liftering coefficient applied to MFCC.

- Numchans: specifies the number of channels utilized.
- Preemcoef: states the way in which the first order pre-emphasis should be applied to the speech samples.
- Usehamming: tells whether the hamming window is utilized or not.
- Windowsize: specifies the size of the recorded audio length.
- Savecompressed: states whether the output should be saved with compression or not.
- Targetrate: specifies the rate with which the speech samples are produced during feature extraction.
- Targetformat: specifies the file format with which the speech samples are produced during feature extraction.
- Targetkind: specifies the kind with which the speech samples are produced during feature extraction.
- Sourcerate: specifies the rate with which the speech samples are read from the speech database.
- Sourceformat: specifies the file format with which the speech samples are read from the speech database.
- Sourcekind: specifies the kind with which the speech samples are read from the speech database.
- Headersize: states the number of bytes in the header.
- Savewithcrc: specifies whether to append the checksum on the data or not.

The path where MFCCs are to be extracted and the one where the wave files are stored should also be specified in the script file. This specification is very important for the MFCCs to be created and for the system to know where the corresponding MFCC of the wavefiles are. An example of the script is shown in Figure 5.6.

```
C:\htk\bin.win32\train\TL06002.wav C:\htk\bin.win32\train\TL06002.mfc
C:\htk\bin.win32\train\TL06003.wav C:\htk\bin.win32\train\TL06003.mfc
C:\htk\bin.win32\train\TL06004.wav C:\htk\bin.win32\train\TL06004.mfc
C:\htk\bin.win32\train\TL06005.wav C:\htk\bin.win32\train\TL06005.mfc
C:\htk\bin.win32\train\TL06006.wav C:\htk\bin.win32\train\TL06006.mfc
C:\htk\bin.win32\train\TL06007.wav C:\htk\bin.win32\train\TL06007.mfc
C:\htk\bin.win32\train\TL06008.wav C:\htk\bin.win32\train\TL06008.mfc
C:\htk\bin.win32\train\TL06009.wav C:\htk\bin.win32\train\TL06009.mfc
C:\htk\bin.win32\train\TL06010.wav C:\htk\bin.win32\train\TL06010.mfc
C:\htk\bin.win32\train\TL06011.wav C:\htk\bin.win32\train\TL06011.mfc
C:\htk\bin.win32\train\TL06012.wav C:\htk\bin.win32\train\TL06012.mfc
. . .
etc.
```

Figure 5-6: Script file used for training.

5.2.4 Model Initialization and Training

In this step, the phoneme model is initiated which would later be utilized for re-estimation of other phoneme models. The initialization is performed by firstly copying the file called *proto* and mapping it to each defined phoneme and call the file containing the HMM definitions *hmmdefs*. *Proto* is a file containing initial values of means and variances of each phone for provision of further estimations.

HERest –C config –I phones0.mlf –t 250.0 150.0 1000.0 –S train.scp –H hmm0/macros –H hmm0/hmmdefs –M hmm1 monophones0.

Later the above command is executed to perform the re-estimation until tied-state triphones are developed. *Config* file is shown in Figure 5-5 and *train.scp* is shown in Figure 5-6. *Monophones0* is a file containing the phone set. The numbers 250 150 and 1000 are threshold values used for pruning the speech data [38].

5.2.5 Evaluation

For the evaluation of the recognizer and to check if it produces results, the speech samples are compared to a network of words by invoking the following command

*HVite –H hmm15/macros –H hmm15/hmmdefs –S test.scp –l * -i recout.mlf –w wdnet –p 0.0 –s 5.0 dict tiedlist.*

The *HVite* component generates the file containing the recorded transcriptions called *recout.mlf* using files such as word network (*wdnet*) as inputs. The *test.scp* is a script file containing the mapping of the testing data to their corresponding feature vectors. *Tiedlist* is a file containing similar models tied together and *dict* is a dictionary created in the first steps of building.

The sections 5.2.4 and 5.2.5 only highlight some of the commands executed during in the construction of a speech recognizer. *Appendix D* shows a list of commands that were executed during the construction of a dialect-based speech recognizer. Young et al. [38] discuss all the steps of building a recognizer including the ones which have been excluded for discussion in latter sub-sections.

5.3 Summary

Building a Northern Sotho speech recognizer is fairly easy when the appropriate files needed are corresponding to the collected speech data, otherwise the whole process would not yield useful results. Important files such as the configuration files should be implemented with great care whenever they are invoked.

Some of the configuration file parameters, such as source rate are determined by the collected data utilized for the development of the speech recognizer. We also outlined the role of other important files used in this experimentation process, such as the training script and the word lexicon.

CHAPTER 6: RESULTS AND EVALUATION

6.1 Introduction

This chapter describes the results that were obtained from a dialect-based system and we compare them to those obtained from a standard Northern Sotho-based system. Also included in this chapter are the difficulties that we encountered during the development of dialect biased ASR system.

6.2 Parameter Evaluation

The initial results that we obtained were using pre-recorded speech data samples, that is, the data had already been recorded and were stored in the database for the purposes of testing the developed dialect-based speech recognizer. Before selecting the training data, we selected the test utterances of the three dialects at random from the pool of collected data. Of utmost importance was to keep track of a balanced set of test data with respect to gender and age.

From the pool of each dialect speech samples, 121 speech data samples of each dialect were chosen to be test utterances. Separate evaluation tests were performed on the dialects (Sepedi, Setlokwa & Selobedu) and the results are shown in Table 6.1. The final accuracy values shown in Table 6.1 are severely compromised by excess of insertions over deletions. The insertion penalty used was too low. If the Viterbi search is rerun with increased insertion penalty (value chosen so that insertions and deletions are approximately equal) the overall system performance would improve significantly.

The overall performance test of the recognizer comprised of the added test utterances of the three dialects which yielded 363 test utterances in total. Table 6.1 shows the results of the ASR system using data that already existed in the database.

Dialect		% correct	N	S	D	I	Accuracy
Sepedi	Word	76.97	595	132	5	139	53.61
	Sentence	36.36	121	-	-	-	
Setlokwa	Word	76.19	609	138	7	172	47.95
	Sentence	40.50	121	-	-	-	
Selobedu	Word	72.02	604	167	2	278	25.99
	Sentence	10.74	121	-	-	-	
Overall System Perfomance	Word	75.06	1808	437	14	589	42.48
	Sentence	29.20	363	-	-	-	

Table 6-1: Results using pre-recorded test samples.

Where: N: total number of words/sentences

S: number of substitution errors

D: number of deletion errors

I: number of insertion errors.

The issue of OOV words occurred when testing the system with the recorded data because the recorded data used for training was not similar to recorded data used for testing. It is extremely difficult to define all natural words in the vocabulary especially when building a medium-sized vocabulary speech recognizer. There were 72 (5.04 %) OOV words in total. The OOV words can also be observed when testing the system with live audio input. With live test, there were 7 (6.9%) OOV words in total. With the reduced number of OOV words, the recognition accuracy would improve significantly.

The system was tested with live audio input to observe its behavior in that situation. In such a situation we don't rely on the data that was already in the database. Table 6-2 shows the results of the live test.

Dialect	Gender	Correct Sentences/ Total Sentences	Correct Words/ Total Words
SETLOKWA	F_1	1/4	7/13
	F_2	0/2	6/9
	M_1	1/3	9/16
	M_2	1/3	4/13
Totals	4	3/12	26/51
SELOBEDU	F_3	1/6	10/20
	F_4	1/4	8/17
	M_3	0/4	5/12
	M_4	0/4	7/12
Totals	4	2/18	30/61
SEPEDI	F_5	1/3	9/15
	F_6	1/5	13/19
	M_5	1/7	13/21
	M_6	0/3	6/17
Totals	4	3/18	41/72

Table 6-2: Results of live audio input.

A total number of 12 participants comprising of 6 males and 6 females tested the system with live speech input. For each dialect 2 males and 2 females tested the system. The overall results of live input yielded 16.7% of correct sentences and 52.7% of correct words. The overall results of live testing are shown in Table 6.3.

Accuracy	Percentages
Sentence	16.7%
Word	52.7%

Table 6-3: Overall results of live input expressed in percentages.

Before testing the ASR system with live audio input, some parameters in the configuration file were changed. Figure 6.1 defines the configuration file utilized for running the speech recognizer with live audio input.

```
# Waveform capture

TARGETKIND =      MFCC_0_D_A
TARGETRATE =      100000.0
SOURCERATE=       1250
SOURCEKIND=       HAUDIO
SOURCEFORMAT= HTK
ENORMALISE=       FALSE
USESILDET=        TRUE
MEASURESIL=       FALSE
OUTSILWARN=       TRUE
```

Figure 6-1: Configuration file used for live audio input data.

The improvement on the overall performance of the system was brought by tree clustering which permits phonemes to share parameters appropriately. A brief discussion on tree clustering is covered in section 6.3. For each phoneme defined, we built three decision trees so as to produce the tying of states for better recognition results. The decision tree included the three phoneme locations in words namely; beginning, middle and end (*See section 4.4.1).* That is, the decision trees on the three locations of every phone defined in the set were established.

6.3 Challenges Encountered

Speaking Rate

The configuration file which had to accommodate the three dialects presented quite a challenge. The speaking rate with which the respondents uttered speech samples during data collection differed according to the dialectal regions volunteers came from. We realized, after having measured the lengths of responses with respect to the selected dialects, that the responses from the people around the region of Bolobedu

were much slower compared to ones uttered by people from Botlokwa. It was difficult to set up configuration parameters to accommodate the different speaking speeds, but after some experimentation this was done.

Phoneme representation

The representation of certain words during transcription in Selobedu and Setlokwa was problematic. We struggled to define some vowels and consonants that can represent certain words and some of them were too difficult to represent as discussed in chapter 4. The variations in pronunciations of words were the main cause of this problem.

We solved the problem by representing the words the way they are pronounced in different dialects by adding two new phones to a set of standard phones of Sepedi. The word "***tšeuwe***" is represented as "***dzweuwe***" in Selobedu. In this case, ***dz*** is used to represent ***tš*** in Selobedu.

Speech Data Transcriptions

The elimination of non-speech events in the collected speech samples was extremely time-consuming. The process involved editing each waveform file individually and removing the noisy utterances. The main purpose of manually editing the speech samples is to train the speech recognizer with clean speech samples to avoid confusions. If the speech recognizer is trained with clean speech, it will perform better in live conditions especially in noisy environments. The cleaned speech samples were listened to at least twice for verification and validity and later the corresponding text representation was created.

Decision tree clustering

The decision tree helps the speech recognition system to make better decisions when coming to the selection of words in the pronunciation dictionary. Suppose we have a phone *kh* which is likely to be confused with the models of *s* for Selobedu. All occurrences of the phone *kh* were defined, especially noting when it should remain unchanged and the time where it should change to *s*. For example, we can map *kh* to change to *s* in a word only when it is followed by phone *e*.

The one major challenge encountered when designed this tree was the one of finding out which vowel might follow a certain consonants and vice versa and all instances of such occurrences were nearly covered. We tried to accommodate all phonemes but out-of-vocabulary words still remained an issue as in most speech recognizers. In conclusion we can say that the decision tree clustering resulted in only minor improvements to the overall performance of the system.

6.4 Comparison with the baseline system

The baseline system [25] was designed considering only one dialect of Northern Sotho, namely Sepedi. It was trained on 2706 sentences in total. The parameters utilized to perform the analysis on the training data were:

Window = Hamming Windows (25 msec)
Frame Period = 10msec
Pre-emphasis = $1 - 0.97\, z^{-1}$
Resolution = 8 Bits
Sampling Frequency = 8KHz.

The pre-recorded speech data was represented by a 39 dimensional feature vectors with 12 MFCC parameters. On the testing data, a total of 332 sentences and 1405 words were used to perform the testing of the recognizer. The results that were obtained for the evaluation of the baseline system are shown in Table 6.4.

Accuracy	Training Speakers	Live Test
Sentence	51 .1 %	14%
Word	84.4 %	62.3%

Table 6-4: Results of the baseline system [25].

From the above analysis of results we can conclude that an ASR system built on the different dialects of Northern Sotho produces lower recognition results than the

speech recognizer built on Sepedi only. Figure 6.2 compares the results of the two systems in a chart. (*See also figure 6.1 and table 6.3).*

The recognizer did not produce the same accuracy when run live compared to the pre-recorded test data. This may be ascribed to the fact that microphone position and surrounding noise may have influenced the testing. Although the results are poorer than the baseline system in this case which is similar to the pre-recorded data tests.

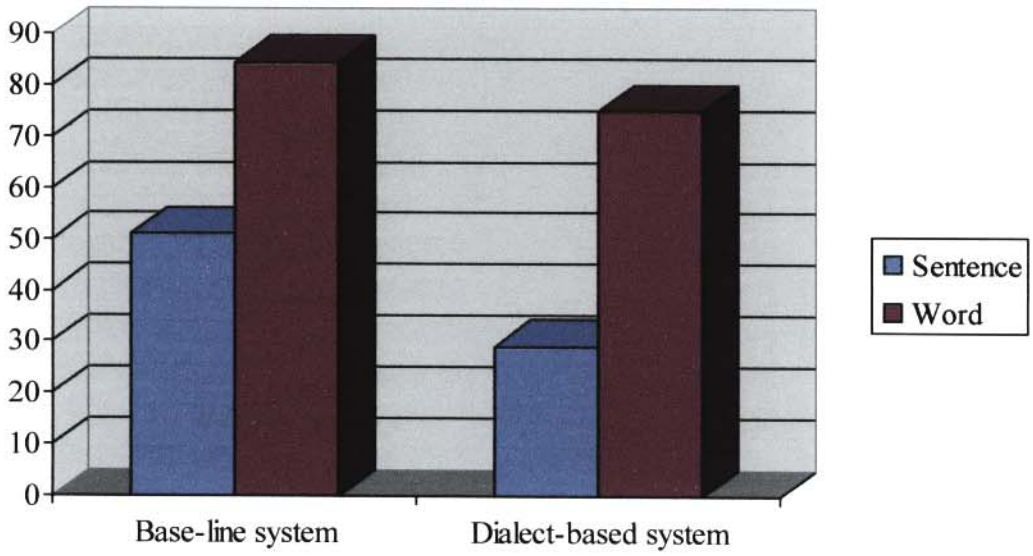

Figure 6-2: Comparison of a baseline and a dialect-based system.

6.5 Summary

In this chapter we discussed and compared the results of the baseline system and the dialect based speech recognition system. A chart representing the comparison is also included. We also stated challenges encountered when building the dialect-based speech recognizer.

CHAPTER 7: CONCLUSION

From our problem statement; the main purpose of pursuing this research project is to investigate as to what extent it would be possible to build a robust automatic continuous speech recognizer that can handle a few different dialects of Northern Sotho simultaneously. We have successfully developed a dialect-sensitive speech recognizer. The results were compared with a Northern Sotho-based speech recognizer with a strong language model that has already been constructed. We noted that our recognition accuracy rate is lower than that obtained for the baseline system.

There are major factors that affect the performance of a dialect-based speech recognizer; one is speaking speed, which makes it difficult to set the parameters of the configuration file *as noted in section 6.3*. That is, the parameters set were not fully compatible with every dialect. The overall results of the dialect-based speech recognizer yielded 43% with a 75% word recognition rate and a 29% sentence recognition rate.

The main reason of producing a low sentence recognition rate is that if in a sentence one or two words are incorrectly recognized, the sentence is regarded as incorrectly recognized. Suppose a sentence consists of five words, if four words are correctly recognized, the sentence recognition rate remains zero because of that single word which is incorrectly recognized.

Since this study serves as the foundation study of the dialects of Northern Sotho, further research is needed to improve its recognition rate. The main important and challenging issue is the representation of the dictionary to accommodate the different dialects. Another reason that led to lower results than what was obtained by the baseline system, is because of the dictionary representation and how the words were selected for recognition. Initially, the selection relied on the first phoneme representation it encountered in the dictionary.

For example, suppose a word *SESEPE* has the following phoneme representations

SESEPE *KH E S E B H E*

SESEPE *S E S E P E*

The phoneme representation of the word *SESEPE* would then be *KHESEBHE*. After having attached a decision tree at a later stage, the speech recognizer had access to multiple pronunciations which resulted in improvements being observed.

Our representation did not function satisfactorily. The added variants in the pronunciation lexicon for accommodation of three dialects caused confusion in the recognition dictionary and also it increased the problems in the training and re-estimation of the phone models by mixing the HMM definitions of other phone models. Confusion arises when the speech recognizer was trained by combined speech samples of the three dialects.

For further improvement, the following can be considered:

- The size of the vocabulary can be increased to tens of thousands of words to yield a larger vocabulary such that all phonemes incorporated into the system are well trained to produce the required high recognition accuracy.
- Researching more on the dialects to find minimal differences of word pronunciations would also serve as another method of improving the system performance because confusion in the pronunciation dictionary will then be minimized.

REFERENCE

[1] "History of speech and voice recognition and transcription software", http://www.dragon-medical-transcription.com/history_speech_recogntion.html (Accessed on 15 May 2005).

[2] "HTK", http://htk.eng.cam.ac.uk/ (Accessed on 02 September 2005).

[3] "Map of Limpopo", http://www.places.co.za/maps/limpopo_map.html (Accessed on 05 October 2005).

[4] "Speech recognition background", http://www.infraware.com/whitepapers/Speech recognition backgrond - infraware.pdf (Accessed on 26 July 2005).

[5] Abara, C. and Wang, P., "Speech Recognition", University of Temple, Philadelphia, http://www.cis.temple.edu/~pwang/203-AI/Project/2001/Abara.htm, (Accessed on 05 September 2005).

[6] Beccheti, C. and Ricotti, L.P., *Speech Recognition: Theory and C++ Implementation*, John Wiley & Sons, 1999.

[7] Black, A.W. and Lenzo, K.A., "Building voices in the festival Speech Synthesis System", Language Technologies Institute, Carnegie Mellon University, http://www.festvox.org/festvox/index.html, (Accessed on 29 October 2004).

[8] Buckley, R., "Infoedge: Speech Recogntion", http://www.infoconomy.com/pages/emerging-technologies/group103384.adp, (Accessed on 09 June 2005).

[9] Cassidy, S. and Harrington, J., *Techniques in Speech Acoustics*, Kluwer Press, 1999.

[10] Chesta, C., Olivier, S. and Lee, C.H., "Maximum a posteriori linear regression for Hidden Markov Model adaption", *In Eurospeech'99*, pp. 211-214, Hungary, 1999.

[11] Cole, R., *Survey of the state of the art in human language technology*, Cambridge University press, 1998.

[12] de La Torre, C., Caminero-Gil, F.J., Alvarez, J., Martin del Alamo, C. and Hernandez-Gomez, L., "Evaluation of the Telefonica I+D Natural Numbers Recognizer over different dialects of Spanish from Spain and America", *In ICSLP-1996*, pp. 2032-2035, Philadelphia, 1996.

[13] Deketelaere, S., Deroo, O. and Dutoit, T., "Speech processing for communications: What's new?", *In Revenue HF*, pp. 5-24, Belgium, 2001.

[14] Doe, L.H., "Evaluating the effects of Automatic Speech Recognition Word Accuracy", Master's Thesis, Virginia Polytechnic Institute and State University, Blacksburg, 1998.

[15] Dunn, B., "Speech signal processing and speech recognition", http://caip.rutgers.edu/~rabinkin/DSP_no_audio.pdf (Accessed on 25 July 2005).

[16] Dyer, C.R. and Skrentny, J.D., "Speech recognition", http://www.cs.wisc.edu/~dyer/cs540/notes/speech-2up.pdf (Accessed on 10 March 2004).

[17] Gurlekian, J., "Database for an automatic speech recognition system for Argentine Spanish", *Proceedings of the workshop on Linguistic Databases*, pp. 99 – 104, University of Pennsylvania, 2001.

[18] Henry, J.S, "Speech Recognition and Voice command in modern commercial aircrafts cockpits", Master's Thesis, Embry-Riddle Aeronautical University, 2000.

[19] Hua, Y. and Schultz, T., "Enhanced tree clustering with single pronunciation dictionary for conversational speech recognition", *In Eurospeech 2003*, pp. 1869-1872, Geneva, 2003.

[20] Huang, X., Acero, A. and Hon, H., *Spoken language processing*, Prentice Hall, Upper Suddle River, N.J., 2001.

[21] Jelinek, F., *Statistical Methods for Speech Recognition (Language, Speech and Communication)*, Cambridge, Massachusetts,1998.

[22] Kudo, I., Nakama T., Watanabe, T. and Kameyama, R., "Data collection of Japanese dialects and its influence into Speech Recognition", *In ICSLP-1996*, pp. 2021-2024, Philadelphia, 1996.

[23] Lau, R., "Subword lexical modeling for speech recognition", PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, 1995.

[24] Levison, S., Olive, J. and Tschirgi, J., "Speech Synthesis in Telecommunications", *IEEE Communication Magazine*, pp. 46-53, 1993.

[25] Modiba, T., "Aspects of automatic speech recognition with respect to Northern Sotho", *Unpublished*, Master's Thesis, University of Limpopo.

[26] Mokgokong, P., *A dialect-Gegraphy survey of the phonology of the Northern Sotho Area*, University of South Africa, 1966.

[27] Rabiner, L., "Challenges in Speech Recognition", http://users.ece.gatech.edu/~chl/ngasr03/chair-rabiner.pdf (Accessed on 12 April 2005).

[28] Rabiner, L. and Juang, B.H., "An introduction to HMMs", *IEEE ASSP Magazine 3*, pp. 4-16, 1986.

[29] Rabiner, L. and Juang, B.H., *Fundamentals of Speech Recognition*, Prentice Hall, New Jersey, 1993.

[30] Reddy, D.R., Erman, L.D., Fennel, R.D. and Lesser, V.R., "Organization of the HEARSAY-2 speech understanding system", *Proceedings of the IEEE International Conference*, Vol. 23, pp.10-23, 1975.

[31] Roux, J.C. and Louw, P., "Black South African English and speech technology applications", *South African Journal of Linguistics*, Supplement 38: pp. 4-13, 2000.

[32] Schlüter, R., "Investigations on Discriminative Training Criteria", PhD Thesis, Aachen University of Technology, Germany, September 2000.

[33] Sixtus, A. and Ney, H., "Training of Across-Word Phoneme Models for Large Vocabulary Continuous Speech Recognition", *Proceedings of the IEEE International Conference*, pp. 849-852, Orlando, 2002.

[34] Sproat, R., "ECE 598: Speech Synthesis", http://catarina.ai.uiuc.edu/ECE598/ (Accessed on 12 May 2005).

[35] Sullivan, A. and Lane, F., "Speech recognition", http://shakti.trincoll.edu/~asulliv2/_project.html, (Accessed on 31 June 2005).

[36] Turunen, M., "Jaspis - A Spoken Dialogue Architecture and its Applications", PhD dissertation, Department of Computer Science, University of Tampere, 2004.

[37] Webb, A.R., *Statistical Pattern Recognition*, John Wiley & Sons, 2002.

[38] Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. and Woodland, P., *The HTK Book (for HTK version 3.3)*, University of Cambridge, 2005.

APPENDIX A

Prompt sheet utilized for speech data collection

Nomoro ya letlakala: PED 07

University of Limpopo (Turfloop Campus)
Telkom Center Of Excellence For Speech Technology (TCoE4ST)
Speech Data Collection Sheet

TCoE4ST

O kgopelwa go tlatša letlakala le gomme o le sware pele ga ge o thoma go bolela ka mogala. Ge o feditše ka lona o le romele atereseng yeo e latelago :

Att: Mapeka M.A
Office 1012 Mathematical Sciences Building
School of Computational and Mathematical Sciences
Department of Computer Science
Private Bag X1106
Sovenga
0727
E-mail address: 200013305@ul.ac.za
Office Tel.: (015) 268 2243

Polelo ya gago ka motato e tla gatišwa, mme dinthla ka moka tšeo o di fago di tla tšeiwa bjalo ka sephiri. Kgatišo ye e tla šomišetšwa fela dinyakišišo le hlabollo ya theknolotši ya polelo ya Sesotho sa Leboa.

Leina: _____ Sefane: _____
Aterese ya Poso: _____
: _____
: _____
: _____
Bong: Mosadi ☐ Monna ☐
Mengwaga: _____
Letšatši la matswalo: _____(Letšatši, Kgwedi, Ngwaga)
Nomoro ya boitsebišo: _____
Lefelo la moo o belegetšwego gona: _____
Leleme la ka gae: _____
O šoma mošomo ofe? _____
Naa o šomisa mogala wa lethekeng goba wa ka ntlong ge o letša gona bjale?
_____

PELE O LETŠA :

- Gopola gore o ka se lefišwe mogala le go tlatša letlakala le.
- Ka kgopelo tima seyalemoya, thelebišene le mogalathekeng le se sengwe se se kago hlola lešata mola o sa bolela ka mogala.

GOPOLA :

- Ge o araba potšišo, ema go fihlela o ekwa sekamodumo, morago ga fao o bolele ka go lokologa ka tsela yeo o bolelago ka mehla.
- O se tshwenyege ge o dirile phošo, tšwelapele ka dipotšišo tše dingwe tše di latelago ka morago letlakaleng.
- Mo nomorong ya boitsebišo, ga o gapeletšege go re fa nomoro ya nnete, nomoro ye nngwe le ye nngwe ya go ba le dinomoro tše lesometharo, e lokile.
- O tla botšišwa potšišo tša kakaretšo ka e tee ka e tee. Mo karolong ya bobedi o kgopelwa gore o bale karolwana yeo e tladitšwego letlakaleng moo go bego le mmala wo mohwibidu fela.
- O kgopelwa go se beye mogala fase pele ga ge o fetša ka letlakala le gore o tle o kgone go ba gare ga bao batlogo humana meputso ya tebogo.
- Leka go fa dikarabo tša gago ka polelo ya geno.
- Netefatša gore o tseba nako yeo o re leletšago ka yona.

BJALE LELETJA NOMORONG YE YA MOGALA : 0800 204 707 PELE GA DI 08 AGOSTOSE 2005

KAROLO YA PELE.

1. Bala nomoro ya letlakala yeo e lego khutlonneng ya letlakala ka gare ga lepokisana le le hwibidu.
2. Leina le sefane sa gago ke mang?
3. Naa o monna goba mosadi?
4. O na le mengwaga e mekae?
5. Re fe nomoro ya gago ya boitsebišo.
6. O belegetšwe kae?
7. O bolela polelo efe?
8. O šoma mošomo ofe?
9. Naa o kwešiša polelo ya geno?
10. Naa o kwešiša Se-japane?
11. Ke nako mang gona bjale?

KAROLO YA BOBEDI.

BALA TŠE LEGO MO FELA

1	**Nomoro**	**5**
2	**Lentšu**	**Bjatladi**
3	**Lefoko**	**Kolobe e ja e pshikologa ka segong**
4	**Sekafoko**	**Ba khotšhe ka bontšhi**
5	**Nomoro ya mogala**	**0725468822**
6	**Sekafoko**	**Ditlholego tša segologolo**
7	**Lentšu**	**Bofša**
8	**Sekafoko**	**Sehlare se a mpšhegiša**
9	**Lefoko**	**A re ka ntahle psha**
10	**Lentšu**	**Fepša**

English translation of the original prompt sheet

Sheet Number: PED 07

University of Limpopo (Turfloop Campus)
Telkom Center Of Excellence For Speech Technology (TCoE4ST)
Speech Data Collection Sheet

TCoE4ST

Please fill in this data sheet and keep it handy when making a call. After the phone call, please post it to the following address:

Att: Mapeka M.A
Office 1012 Mathematical Sciences Building
School of Computational and Mathematical Sciences
Department of Computer Science
Private Bag X1106
Sovenga
0727
E-mail address: 200013305@ul.ac.za
Office Tel.: (015) 268 2243

All information provided by the participant will be kept confidential and only be used for research purposes (developing a recognizer).

Name : _____ Surname : _____
Postal Address : _____
: _____
: _____
: _____
Gender : Male ☐ Female ☐
Age : _____
Date of Birth (DD-MM-YYYY) : _____
Identity Number : _____
Place of Birth : _____
Home Language : _____
Occupation? _____
Are you from a cell phone or a landline? _____

BEFORE YOU CALL

- Please remember that this a free call, i.e the participant will not be charged
- The participant is at least requested to call from a quite environment and also turn off any appliance that might cause noise.

REMEMBER

- The whole process of calling and answering takes about 10 minutes.
- Make sure that you know today's date and time before calling.
- After every question asked, the participant is requested to wait for a tone\beep and answer normally after that tone.
- When a mistake has been made, the participant is requested to continue answering the next items the computer asks.
- On the first section of the data sheet, the participant will be asked question and on the second section required to read the given items.
- Please do not hang up before the end of the call if you want to be eligible for a prize.

NOW PLEASE CALL THIS TOLL FREE NUMBER : 0800 204 707 BEFORE 08 AUGUST 2005

SECTION 1.

1. Read the sheet number on the top right of your page.
2. What is your name and surname?
3. Are you male or female?
4. How old are you?
5. Give us your identity number.
6. Where were you born?
7. Which language do you speak at home?
8. What is your occupation?
9. Do you understand your home language?
10. Do you understand Japanese?
11. What time is it now?

SECTION 2.

READ ONLY THIS PORTION

1	**Nomoro**	**5**
2	**Lentšu**	**Bjatladi**
3	**Lefoko**	**Kolobe e ja e pshikologa ka segong**
4	**Sekafoko**	**Ba khotšhe ka bontšhi**
5	**Nomoro ya mogala**	**0725468822**
6	**Sekafoko**	**Ditlholego tša segologolo**
7	**Lentšu**	**Bofša**
8	**Sekafoko**	**Sehlare se a mpšhegiša**
9	**Lefoko**	**A re ka ntahle psha**
10	**Lentšu**	**Fepša**

APPENDIX C

Table containing phonemes together with their usage counts for this project

New Phone Usage Counts		
1.	A	1169
2.	SP	1115
3.	D	207
4.	I	384
5.	E	997
6.	G	133
7.	L	433
8.	TŠ	84
9.	W	194
10.	O	746
11.	NG	81
12.	F	69
13.	FŠ	15
14.	M	402
15.	H	125
16.	P	113
17.	N	267
18.	B	186
19.	R	128
20.	FS	15
21.	HL	86
22.	U	155
23.	KH	73
24.	S	205
25.	KG	54
26.	NY	54
27.	T	100
28.	TH	58
29.	TS	44
30.	J	34
31.	BJ	27
32.	K	108
33.	TL	71
34.	Š	79
35.	TLH	49
36.	TŠH	39
37.	PH	73
38.	PŠ	12
39.	Z	17
40.	Y	16
41.	PŠH	9
42.	PSH	10
43.	TSH	36

APPENDIX D

Phoneme ambiguity in words

Phone	Sepedi	Selobedu	Setlokwa
/s/	sesepe	khesebhe	sesepe
/h/	huma	khuma	huma
/g/	marega	marea	mareha
/j/	jela	dyela	dyela
/kg/	kgogo	khoo	kgoho
/ng/	bjang	bjanye	bjane
/p/	ripa	ribha	ripa
/š/	lešata	lesata	lesata
/t/	rata	radhwa	ratwa
/tl/	magetla	magedlha	mahetlda
/ts/	matsetse	madzedze	matsetse

APPENDIX E

A list of HTK commands

1. HParse gram wdnet
2. HDman –e c:\htk\bin.win32 –m –w wlist -g global.ded –n monophones1 –l dlog dict spedi
3. Perl prompts2mlf sentencas.mlf sentencas.txt
4. HLEd -l * -d dict -i phones0.mlf mkphones0.led sentencas.mlf
5. HCopy -T 1 -C config -S codetr.scp
6. HCompV -C config -f 0.01 -m -S train.scp -M hmm0 proto
7. HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 monophones0
8. HHEd -H hmm4/macros -H hmm4/hmmdefs -M hmm5 sil.hed monophones0
9. HVite -l * -o SWT -b silence -C config1 -a -H hmm7/macros -H hmm7/hmmdefs -i aligned.mlf -m -t 250 -y lab -I sentencas.mlf -S train.scp dict monophones0
10. HERest -C config -I aligned.mlf -t 250 150 1000 -S train.scp -H hmm7/macros -H hmm7/hmmdefs -M hmm8 monophones1
11. HLEd -n triphones1 -l * -i wintri.mlf mktri.led aligned.mlf
12. perl maketrihed monophones0 triphones1
13. HHEd -B -H hmm9/macros -H hmm9/hmmdefs -M hmm10 mktri.hed monophones1
14. HERest -B -C config -I wintri.mlf -t 250 150 1000 -s stats -S train.scp -H hmm10/macros -H hmm10/hmmdefs -M hmm11 triphones1
15. perl mkclscript.prl TB 350 monophones0 > tree.hed
16. HDMan -b sp -n fulllist -g global.ded -l flog spedi-tri spedi
17. HHEd -H hmm12/macros -H hmm12/hmmdefs -M hmm13 tree.hed triphones1 > log
18. HERest -C config -I wintri.mlf -t 250 150 1000 -s stats -S train.scp -H hmm13/macros -H hmm13/hmmdefs -M hmm14 tiedlist
19. HVite -H hmm15/macros -H hmm15/hmmdefs -S test.scp -l * -i recout.mlf -w wdnet -p 0 -s 5 dict tiedlist
20. HResults -l testref.mlf tiedlist recout.mlf