GRAPHEME-BASED CONTINUOUS SPEECH RECOGNITION FOR SOME OF THE UNDER-RESOURCED LANGUAGES OF LIMPOPO PROVINCE

by

MABU JOHANNES MANAILENG

DISSERTATION

Submitted in (partial) fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

in the

FACULTY OF SCIENCE AND AGRICULTURE

(School of Mathematical and Computer Sciences)

at the

UNIVERSITY OF LIMPOPO

SUPERVISOR : MR M.J.D. MANAMELA

CO-SUPERVISOR : DR M. VELEMPINI

2015

DECLARATION

I, Mabu Johannes Manaileng, declare that the dissertation entitled “GRAPHEME-BASED CONTINUOUS SPEECH RECOGNITION FOR SOME OF THE UNDER-RESOURCED LANGUAGES OF LIMPOPO PROVINCE”, is my own work and has been generated by me as the result of my own original research. I confirm that where collaboration with other people has taken place, or material from other researchers is included, the parties and/or material are appropriately indicated in the acknowledgements or references. I further confirm that, this work has not been submitted to any other university for any other degree or examination.

_____ _____

Manaileng, M.J. (Mr) Date

ABSTRACT

This study investigates the potential of using graphemes, instead of phonemes, as acoustic sub-word units for monolingual and cross-lingual speech recognition for some of the under-resourced languages of the Limpopo Province, namely, IsiNdebele, Sepedi and Tshivenda. The performance of a grapheme-based recognition system is compared to that of phoneme-based recognition system.

For each selected under-resourced language, automatic speech recognition (ASR) system based on the use of hidden Markov models (HMMs) was developed using both graphemes and phonemes as acoustic sub-word units. The ASR framework used models emission distributions by 16 Gaussian Mixture Models (GMMs) with 2 mixture increments. A third-order n-gram language model was used in all experiments. Identical speech datasets were used for each experiment per language. The LWAZI speech corpora and the National Centre for Human Language Technologies (NCHLT) speech corpora were used for training and testing the tied-state context-dependent acoustic models. The performance of all systems was evaluated at the word-level recognition using word error rate (WER).

The results of our study show that grapheme-based continuous speech recognition, which copes with the problem of low-quality or unavailable pronunciation dictionaries, is comparable to phoneme-based recognition for the selected under-resourced languages in both the monolingual and cross-lingual speech recognition tasks. The study significantly demonstrates that context-dependent grapheme-based sub-word units can be reliable for small and medium-large vocabulary speech recognition tasks for these languages.

ACKNOWLEDGEMENTS

As I finish this work, I would like to thank the following people for contributing towards the success of this research study:

- My supervisor and co-supervisor, Mr MJD Manamela and Dr M Velempini for their immense support, encouragement, guidance and friendship.
- My technical advisors, particularly Thipe Modipa and Charl van Heerden, for their advice and imperative contribution to the solutions of some of the technical problems I've encountered.
- The Meraka Institute and the Centre for High Performance Computing (CHPC) both from the Council of Scientific and Industrial Research (CSIR), for their productive training workshops that left me knowledge equipped and technically capable.
- Telkom SA, for funding my post-graduate degree study.
- My friends and colleagues, for their moral support and the hope they've given me to finish this work.
- My amazing family, for always being patient with me in all times, and at all cost.

Without their continued support and interest, this work would not have been the same as presented here.

TABLE OF CONTENTS

LIST OF ACRONYMS

ANN – Artificial Neural Networks

ARPA – Advanced Research Projects Agency

ASR – Automatic Speech Recognition

CMN – Cepstral Mean Normalization

CMVN – Cepstral Mean and Variance Normalization

CSIR – Council for Scientific and Industrial Research

CSR – Continuous Speech Recognition

CVN – Cepstral Variance Normalization

DBN – Dynamic Bayesian Networks

DTW – Dynamic Time Wrapping

ExpGra – Grapheme-based Experiment

ExpPho – Phoneme-based Experiment

FSM – Finite State Machine

GMM – Gaussian Mixture Model

HLT – Human Language Technology

HMM – Hidden Markov Model

HTK – Hidden Markov model Toolkit

IVR – Interactive Voice Response

LM – Language Model

LPC – Linear Predictive Coefficients

LVCSR – Large Vocabulary Continuous Speech Recognition

MAP – Maximum A Posteriori

MFCC – Mel Frequency Cepstral Coefficients

MLF – Master Label File

MLLR – Maximum Likelihood Linear Regression

MLPs – Multi-Layer Perceptrons

MULTI-LING – Multilingual

NCHLT – National Centre for Human Language Technologies

OOV – Out of Vocabulary

PLP – Perceptual Linear Predictive

PMC – Parallel Model Combination

SLU – Spoken Language Understanding

STT – Speech-To-Text

SVN – Support Vector Machine

TCoE4ST – Telkom Centre of Excellence for Speech Technology

TTS – Text-To-Speech

VTS – Vector Taylor Series

WER – Word Error Rate

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background

Within the realm of human language technologies (HLTs), there has been an increase in speech processing technologies over the last few decades (Barnard *et al.,* 2010; Besacier *et al.*, 2014). Modern speech technologies are commercially available for a limited but interesting range of man-machine interfacing tasks. These technologies enable machines to respond almost correctly and reliably to human voices, and provide numerous useful and valuable e-services. It remains a puzzle to develop technologies that can enable a computer-based system to converse with humans on any topic. However, many important scientific and technological advances have taken place, thereby bringing us closer to the “Holy Grail” of computer-driven mechanical systems that generate, recognise and understand fluent speech (Davis *et al.,* 1952).

At the core of speech processing technologies lies the automatic speech recognition (ASR), also known as speech-to-text (STT) conversion, Speech synthesis, commonly referred to as text-to-speech (TTS) synthesis, and Spoken language understanding (SLU) technology. Huang *et al.* (2001) describes ASR as a technology that allows computers to identify the words that a person speaks into a microphone or telephone and convert them to written text. TTS as a technology that allows computers to generate human-like speech from any text input to mimic human speakers. And SLU as one comprises of a system that typically has a speech recogniser and a speech synthesiser for basic speech input and output, sentence interpretation component to parse the speech recognition results into semantic forms – which often needs discourse analysis to track semantic context and to resolve linguistic ambiguities. A dialog manager is the central component of the SLU module that communicates with applications to perform complicated tasks such as discourse analysis, sentence interpretation, and response message generation (Huang *et al.*, 2001).

The speech processing research community is continually striving to build new and improved large vocabulary continuous speech recognition (LVCSR) systems for more

languages and continuous speech recognition (CSR) systems for more existing under-resourced languages in different communities and countries. One of the essential components in building ASR systems is a pronunciation dictionary, which provides a mapping to a sequence of sub-word units for each entry in the vocabulary (Stuker *et al.,* 2004). The sub-word units in the pronunciation dictionary are used to model the acoustic realisation of the vocabulary entries. Phonemes, basic contrastive unit of sound in a language, are the most commonly used sub-word units and have shown a notable success in the development of ASR systems (Kanthak *et al.,* 2003; Stuker *et al.,* 2004). However, the use of graphemes, letters or a combination of letters that represent the orthography of a word, as sub-word units have achieved comparable recognition results (Schukat-Talamazzini *et al.*, 1993; Kanthak and Ney, 2002; Kanthak *et al.,* 2003; Stuker *et al.,* 2004; Sirum and Sanches, 2010; Basson and Davel, 2013; Manaileng and Manamela, 2014).

1.2. Research Problem

As the development of LVCSR systems continues to improve, the performance of continuous speech recognisers has steadily improved to the point where even high CSR accuracies are becoming achievable. However, the optimum recognition accuracy of continuous speech recognisers remains a challenge when dealing with some of the local under-resourced African languages such as Sepedi, IsiNdebele, IsiXhosa, Xitsonga and Tshivenda (van Heerden *et al*., 2012; Barnard *et al*., 2010).

The performance of ASR systems is heavily influenced by the comprehensiveness of a pronunciation dictionary used in the decoding process (Stuker *et al.,* 2004). The best recognition results are usually achieved with hand-crafted, i.e., manually created, pronunciation dictionaries (Kanthak *et al.,* 2003; Killer *et al.* 2003). Human expert knowledge about the targeted language is usually required for crafting a pronunciation dictionary and thus making it a *labour-intensive*, *time-consuming* and *expensive* task. If no such expert knowledge is available or affordable, new methods are needed to automate the process of creating the pronunciation dictionary. However, even the

automatic tools often require hand-labelled training materials and rely on manual revision.

The methods used to build LVCSR systems for lucrative languages require enormous amount of linguistic resources which makes it impractical to use the same methods for languages with little or no such resources (Badenhorst *et al.*, 2011; van Heerden *et al*., 2012). For example, the use of hand-crafted dictionaries raises problems when dealing with rare and under-resourced languages since many of these languages have very little or no computational linguistic tools (Stuker *et al.,* 2004). It therefore becomes impractical or nearly impossible to sustain the creation of hand-crafted dictionaries. Moreover, linguistic experts are often *unavailable*, *unaffordable* or even worse, *non-existent* for most under-resourced languages. This is indeed the case with the most of the official under-resourced indigenous languages of South Africa (Barnard *et al*., 2010). Our research study focuses on three of the official under-resourced indigenous languages of South Africa, namely, Sepedi, IsiNdebele and Tshivenda.

Furthermore, there are two kinds of problems that can be introduced by a crafted pronunciation dictionary. The first one can be introduced during the training phase by a false mapping between a word and its modelling units, resulting in the contamination of the acoustic models. The models will as a result not describe the actual acoustics that they ought to represent. Secondly, the incorrect mapping will falsify the scoring of hypotheses by applying the wrong models to the score calculation.

1.3. Motivation for the Research Study

The practitioners of human language technologies (HLTs) tend to find some spoken natural languages more attractive and popular than others. For this reason, what they find as *unattractive* and *unpopular* languages are often deserted, undeveloped and prone to extinction (Crystal, 2000). Crystal (2000) estimates that on average, one language dies every two weeks. It is for this and other reasons that the development of speech recognition systems and related technologies such as machine translation

systems for literally all spoken languages in the world is highly desirable (Besacier *et al.*, 2014).

South Africa has eleven official languages which have, or are at least intended to have, equal economic relevance and value. Very little documented knowledge exists about most of these languages and hence advanced modern linguistic and computational tools are scarce in their day-to-day usage. This situation makes it very difficult to build the required LVCSR systems for all these official languages (Badenhorst *et al.*, 2011). This study is therefore motivated by the need to use methods which require few linguistic and computational resources to build LVCSR systems with acceptable levels of recognition accuracies.

We suggest adopting an approach of developing ASR systems that rely solely on graphemes rather than phonemes as acoustic sub-word units. The mapping in the pronunciation dictionary now becomes completely trivial, since every word is simply segmented into its constituent alphabetic letters. Intensive linguistic expert knowledge is therefore no longer needed. Using graphemes instead of phonemes as acoustic sub-word units for ASR will reduce the cost and time needed for the development of satisfactory ASR systems for our targeted languages.

1.4. Research Aim and Hypothesis

The purpose of the study is to address the problem of creating pronunciation dictionaries in a non-optimal manner with respect to cost and the duration of time. The study aims to investigate the potential of using graphemes, instead of phonemes, as acoustic sub-word units for the ASR of the three under-resourced languages of Limpopo Province.

The research hypothesis is formulated as follow: the grapheme-based acoustic sub-word units achieve acceptable levels of CSR accuracies when compared to phoneme-based units.

1.5. Research Questions and Objectives

Our research questions are framed as follows:

i. Can we use graphemes, instead of phonemes, as acoustic sub-word units for continuous speech recognition of Sepedi, Tshivenda and IsiNdebele?
ii. How do graphemes perform as compared to phonemes in monolingual and multilingual speech recognition for these languages?

The objectives of the study are to:

i. Develop baseline phoneme-based speech recognition systems using the available hand-crafted and automatically created pronunciation dictionaries.
ii. Create grapheme-based dictionaries from the available phoneme-based pronunciation dictionaries, which should require less effort since we only need to extract the word lists and then separate every word into its constituent alphabetic letters.
iii. Develop grapheme-based speech recognition systems using the new grapheme-based pronunciation dictionaries.
iv. Compare the recognition results attainable in both speech recognition experiments for each language and observe whether or not graphemes have the potential of being similarly used as acoustic sub-word units in the decoding process of an ASR.
v. Build a multilingual speech recognition system using the two approaches and then compare the results.

1.6. Research Method

The ASR experiments conducted in our study used two secondary speech corpora. The first corpus used was the Lwazi ASR corpus (van Heerden *et al*., 2009) and the National Centre for Human Language Technologies (NCHLT) ASR corpus (Barnard *et al*., 2014). Both corpora are freely available on the Resource Management Agency (RMA)

website[1]. The phoneme-based pronunciation dictionaries were also obtained from the RMA website. The monolingual acoustic models were trained with an average of 6.5 hours of speech training data for the Lwazi ASR corpus and an average of 41.9 hours for the NCHLT ASR corpus. The models were further tested on an average of 1.6 and 3.1 hours of speech data for the Lwazi and NCHLT ASR corpus, respectively. The multilingual acoustics models, a combination of the monolingual acoustic models trained with the Lwazi speech data, were trained with 19.58 hours of speech data, and tested with 4.85 hours.

The Mel-frequency cepstral coefficients (MFCCs) were extracted as acoustic features and enhanced with the cepstral mean variance normalization (CMVN). For each language, a third order language model was trained from a corpus of the sentential transcriptions of the training data.

The tools used to conduct our experiments involve, the hidden Markov modelling toolkit (HTK) (Young *et al*., 2006) to train acoustic models, the HDecode tool for evaluating the recognition systems, and the SRILM language modelling toolkit (Stolcke, 2002) to train and evaluate the language models.

1.7. Significance of the Study

This study essentially investigates the potential of grapheme-based speech recognition for selected under-resourced languages. The recognition results obtained will provide insight into the potential of using graphemes rather than phonemes for monolingual and multilingual speech recognition of the three targeted languages.

Should such a potential be found to be reasonably acceptable in relation to the current typical ASR performance measures, then the local speech processing research community can adopt the proposed method. This will reduce the cost and time required

[1] http://rma.nwu.ac.za/index.php/resource-catalogue.html

to build CSR systems for more under-resourced languages and possibly their dialects. Such a development will potentially benefit communities that use most of these heavily under-resourced languages on daily basis by ensuring the development and delivery of the much needed automatic computational linguistic tools.

These tools may significantly help with issues of language preservation, elevation, advancement and modernisation, thereby eliminating or drastically reducing threats of the extinction of under-resourced African indigenous languages. Being a multilingual society, linguistic and digital e-inclusion is vital for South Africa to ensure that e-service delivery can be achieved in any of the eleven official languages across the country.

Furthermore, based on some of the results of this research project, two papers (one full and one short) have been published and presented at conferences. Their details are indicated in Appendix G. Moreover, some short papers were presented at Workshops and Masters and Doctoral (M&D) Symposiums.

1.8. Structure of Dissertation

The rest of the dissertation is organized as follows:

- Chapter 2 provides the theoretical background literature on ASR. The chapter begins by providing a historical perspective and theoretical framework of ASR. It further outlines the basic components, classifications and approaches to ASR.
- Chapter 3 discusses some of the previous studies on grapheme-based speech recognition, which forms a basis for the proposed research study. The chapter further discusses CSR for under-resourced languages, the approaches and also the challenges.
- Chapter 4 presents and discusses the research method used to conduct the research study. A detailed description of the design of the experiments is provided.
- Chapter 5 presents the experimental results of the research study. ASR performance evaluation metrics are discussed, the optimum evaluation

parameters are outlined and the evaluation procedure is also described. Furthermore, analysis of the results is presented.

- Chapter 6 summarises the findings of the research study, give a synopsis of the envisioned future work, recommends potential directions of the future and further gives a general conclusion of the research study.

2. THEORETICAL BACKGROUND ON ASR

2.1. Introduction

This chapter discusses some historical perspective on some key inventions and developments that have enabled significant progress in ASR research. We briefly review the current state of ASR technology and also enumerate some of the challenges that lie ahead of the speech processing research community. The statistical framework and basic components of ASR systems are also discussed in detail.

2.2. Historical Background of ASR

The ASR technology has been a topic of great interest to a broad general population since it became popularised in several blockbuster movies of the 1960's and 1970's. The most notable was the movie "2001: A Space Odyssey" by Stanley Kubrick (Juan *et al.,* 2004). However, early attempts to design ASR systems came in the 1950's and were mostly guided by the theory of acoustic-phonetics. This theory describes the phonetic elements of speech (the basic sounds of a language) and attempts to explain how they are acoustically realised in a spoken utterance (Juan *et al.,* 2004).

What makes ASR research most appealing is the fact that speech is the most natural, easiest, effortless and convenient way to achieve inter-human communication (Juan *et al.,* 2004). With the rapid increase and uptake of information and communication technology in everyday life, there is an increasing need for the communities in computing to adapt and embrace computational devices endowed with some semblance of human behaviour traits, thereby making man-machine interfacing easy to use.

In 1952, Davis, Biddulph, and Balashek of Bell Laboratories built a system for isolated digit recognition for a single speaker (Davis *et al.,* 1952), using the formant frequencies measured during vowel regions of each digit. Fry and Denes also built a phoneme recogniser to recognise 4 vowels and 9 consonants (Fry *et al.,* 1959). In the late 1960's,

Atal and Hanauer formulated the fundamental concepts of Linear Predictive Coding (LPC) (Atal *et al.,* 1971), which greatly simplified the estimation of the vocal tract responses from speech waveforms.

The study of spectral distance measures (Itakura, 1975) and statistical modelling techniques (Juang *et al.,* 1986) led to the technique of mixture density hidden Markov models (HMMs) (Lee *et al.,* 1990) which has since become the popular representation of speech units for speaker-independent continuous speech recognition. The Bell Laboratories also introduced an important approach called “keyword spotting” as a primitive form of speech understanding (Wilpon *et al.,* 1990). Many researchers successfully used the HMM technique of stochastic processes (Poritz, 1982; Liporace, 1982). Another technology that was (re)introduced in the late 1980’s, after failing in the 1950’s, was the idea of artificial neural networks (ANN) (Lippmann, 1989).

2.3. Statistical framework of ASR Process

The speech recognition problem can be formulated as follows (Huang *et al.*, 2001): For a given acoustic signal $X = x_1, x_2,\ldots, x_m$, the main task is to find the word sequence $W^* = w_1, w_2,\ldots, w_n$, which is produced by or corresponds to the acoustic event X. The length of X is m and the length of W^* is n. The word sequence W^* is found by computing the maximum posterior probability that a word sequence W^* was spoken given an observed acoustic signal X – which is expressed as follows:

$$W^* = \underset{W}{\operatorname{argmax}} P(W|X) \tag{2.1}$$

However, the required maximum likelihood probability of the word sequence W^* cannot be directly estimated, it is therefore computed using Bayes’ decision rule as follows:

$$W^* = \underset{W}{\operatorname{argmax}} \frac{P(X|W).P(W)}{P(X)} \tag{2.2}$$

Assuming that the a priori probability $P(X)$ remains constant throughout the decoding process, equation (2.2) can be expressed as follows (also known as the *Fundamental Equation of Speech Recognition* (Huang *et al.*, 2001)):

$$W^* = \underset{W}{\operatorname{argmax}}\, P(X|W).P(W) \tag{2.3}$$

Equation (2.3) can further be classified into the following three basic components:

i. Acoustic Model – the calculation of the conditional probability $P(X|W)$ to observe the acoustic signal *X* given a word sequence *W* was spoken.
ii. Language Model – the calculation of the a priori probability $P(W)$ that word sequence *W* was spoken.
iii. Search – the most efficient calculation of word sequence *W** that maximise $P(W|X)$.

2.4. Basic components of ASR Systems

The speech recognition process seems fairly easy for humans. However, it should be borne in mind that the human intellect uses an enormous knowledge base about the world. The challenges of ASR lie in segmenting the speech data (e.g., determining the start and end of words), the complexity of the speech data (how many different words are there and how many different combinations of all those words is possible), the variability of the speakers (women have a higher fundamental frequency than men), variability of speech channel (microphones, telephones, mobile phones, etc.), ambiguity of spoken words (“two” versus “too”), determination of word boundaries (“interface” versus “in her face”), the semantics and ambiguity in pragmatism (Huang *et al.,* 2001; Juang *et al.,* 2004).

The fundamental goal of an ASR system is to *accurately* and *efficiently* convert a speech signal into a text transcription of the spoken words. The conversion must be independent of the speaker, the device used to record the speech (i.e., the transducer or microphone), or the environment (Rabiner, 2004). Standard ASR systems commonly

consist of five main modules, namely: signal acquisition, feature extraction, acoustic and language modelling and search/decoding. A block diagram of a typical ASR system is depicted in Figure 2.1.

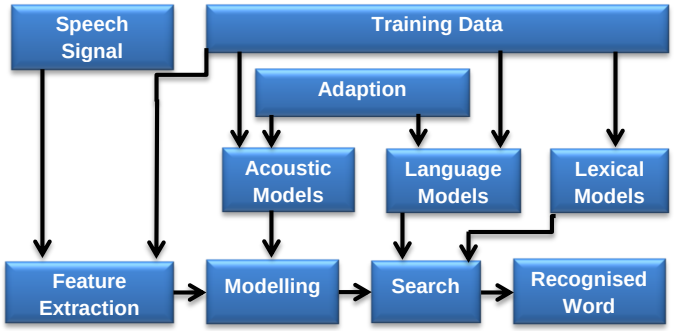

Figure 2.1: ASR block diagram (Wiqas, 2012)

2.4.1. Speech Signal Acquisition

The speech signal acquisition module is responsible for detecting the presence of a speech signal, capturing the signal and passing it to the feature extraction module for further processing. However, the accurate and efficient capturing or acquisition of a speech signal plays a primary role in the entire recognition process since all the succeeding processing modules depend entirely on the accuracy of the signal captured.

2.4.2. Feature Extraction

The primary goal of feature extraction, also referred to as speech parameterization, is to efficiently extract a set of measurable and salient features that characterize the spectral properties of various speech sounds (the sub-word units) (Rabiner, 2004). This is achieved by dividing the input speech into blocks and deriving a smoothed spectral estimate from each block. The blocks are typically 25 milliseconds (ms) long to give a longer analysis window and are generally overlapped by 10 ms. To make this possible, an assumption is made that the speech signal can be regarded as stationary over a few milliseconds.

The standard feature set for most ASR systems is a set of Mel-frequency cepstral coefficients (MFCCs) (Davis, 1980), along with the first and second order derivatives of these features. To produce MFCC coefficients, the spectral estimate is computed using either fast Fourier transform (FFT) (Rabiner, 1975), Linear Predictive Coding (LPC) (Atal, 1971), or Perceptual Linear Predictions (PLP) (Hermansky, 1990).

2.4.3. Acoustic Modelling

The acoustic modelling module forms the central component of an ASR system (Huang *et al.*, 2001). The process of acoustic modelling accounts for most of the computational load and performance of the overall ASR system. As previously indicated, the goal of acoustic modelling is to observe the acoustic signal *X* given that a word sequence *W* was spoken by calculating the conditional probability $P(X|W)$. That is, the acoustic modelling module links the observed features of the speech signal with the expected phonetics of the hypothesis word and/or sentence. Statistical models are used to characterize sound realization. One such statistical model used is the HMM technique (Rabiner, 1989; Young, 2008).

The HMMs are used to model the spectral variability of each of the basic sounds in the language using a mixture density Gaussian distribution, also known as a Gaussian mixture model (GMM). The GMM is optimally aligned with a speech training set and then iteratively updated and improved. That is, the means, variances, and mixture gains are iteratively updated until an optimal alignment and match is achieved (Juang *et al.*, 2004). The HMMs typically have three emitting states and a simple left-right topology (Young, 2008). The models are easily joined by the entry and exit states. Composite HMMs can be formed by merging the entry state of one phone model to the exit state of another, allowing the joining of phone models to form words or words to form complete sentences. The HMMs are mostly preferred because of their flexibility to perform context-dependent and context-independent acoustic modelling (Rabiner, 1989).

2.4.4. Language and Lexical Modelling

The purpose of a language model, or a grammar, is to provide a mechanism for estimating the probability of some word, w_n, occurring in an utterance (or a sentence) given the preceding words, $W_1^{n-1} = w_1, w_2, \ldots, w_{n-1}$ (Jelinek *et al.*, 1991). As stipulated in equation (2.2), $P(W)$ represents the language model. The practical challenge of language modelling is how to build these models accurately so that they can truly and accurately reflect the structural dynamics of spoken language to be recognized (Young, 1996; Huang *et al.*, 2001; Juang *et al.*, 2004).

There are several methods of creating robust language models, including the use of rule-based systems (i.e., deterministic grammars that are knowledge driven), and statistical methods which compute an estimate of word probabilities from large training sets of textual material (Juang *et al.*, 2004). The most convenient way of creating robust language models is to use statistical *n*-grams – which are constructed from a large training set of text. An *n*-gram language model assumes that w_n depends only on the preceding $n-1$ words, that is,

$$P(w_n|W_1^{n-1}) = P(w_n|W_{n-k+1}^{n-1}) \tag{2.4}$$

The *n*-gram probability distributions can be computed directly from text data using counting methods and hence there is no requirement to have explicit linguistic rules such as a formal grammar of the language (Young, 1996). To estimate a trigram ($n = 3$) probability – which is the probability that a word w_n was preceded by the pair of words (w_{n-1}, w_{n-2}), the quantity can be computed as (Jelinek, 1991; Huang *et al*., 2001):

$$P(w_n|w_{n-1}, w_{n-2}) = \frac{C(w_{n-2}, w_{n-1}, w_n)}{C(w_{n-2}, w_{n-1})} \tag{2.5}$$

From equation (2.5), $C(w_{n-2}, w_{n-1}, w_n)$ is the frequency count of the word triplet consisting of a sequence of words (w_{n-2}, w_{n-1}, w_n) that occurred in the training set, and

$C(w_{n-2}, w_{n-1})$ is the frequency count of the word duplet (bigram) consisting of a sequence (w_{n-2}, w_{n-1}) that occurred in the training set.

Training *n*-gram language models generally works very well and is used in the development of state-of-the-art ASR systems (Young, 1996). However, they do have limitations (Jelinek, 1991). One of the problems raised by *n*-grams is that for a vocabulary of X words, there are X^3 possible trigrams. This creates an acute *data sparsity problem* in the training data set as a result of a large number of potential trigrams even for small vocabularies (e.g., 5000 words imply $5000^3 = 125\,000\,000\,000$ possible trigrams). As a result, many trigrams may not appear in the training data and many others will only appear few times (once or twice). In this case, equation (2.5) computes a very poor estimate of the trigram.

Some solutions to the training *data sparsity problem* include using a combination of *discounting* and *back-off* (Katz, 1987). Moreover, when estimating trigrams (or any *n*-gram where *n* is more than 3), a smoothing algorithm (Bahl *et al.*, 1983) can be applied by interpolating trigram, bigram and unigram relative frequencies, i.e.,

$$\hat{P}(w_n|w_{n-1}, w_{n-2}) = p3\frac{C(w_{n-2},w_{n-1},w_n)}{C(w_{n-2},w_{n-1})} + p2\frac{C(w_{n-1},w_n)}{C(w_{n-1})} + p1\frac{C(w_n)}{\sum_n C(w_n)} \qquad (2.6)$$

$$p3 + p2 + p1 = 1$$

$\sum_n C(w_n)$ = the size of text training corpus

where the smoothing probabilities, $p3, p2, p1$ are obtained by applying the principle of cross-validation (Bahl *et al.*, 1983; Huang *et al.*, 2001).

Lexical modelling involves the development of a lexicon (or a pronunciation dictionary) that must provide the pronunciation of each word the task vocabulary. Through lexical modelling, various combinations of phonemes, syllables or graphemes (depending on the choice of sub-word units) are defined to give syntactically valid words for the speech recognition process. This is necessary because the same word can be pronounced

differently by people with different accents, or because the word has multiple meanings that change the pronunciation due to the context of its use, known as pronunciation variants.

2.4.5. Search/Decoding

The role of the decoding module (or simply, the decoder) is to combine the probabilities obtained from the preceding components – acoustic models, language models and the lexical models, and use them to perform the actual recognition process by finding an *optimal* sequence of words *W** that maximises $P(W|X)$ in equation (2.5). An *optimal* word sequence *W** is that which is consistent with the language model and that has the highest probability among all the potential word sequences in the language, i.e., *W** must be the best match of the spectral feature vectors of the input signal (Juang *et al.*, 2004).

The primary task of the decoder – which a basically a pattern matching system, is to find the solution to the search problem, and to achieve the goal it searches through all potential word sequences and assigns probability scores to each of them using a pattern matching breadth-first search algorithm such as the Viterbi decoding algorithm (Huang *et al.*, 2001; Young, 2008) or its variants, commonly used by the *stack-decoders* or *A*-decoders* (Paul, 1991; Kenny, 1991).

The challenge for the decoder is to build an efficient structure to search the presumably large lexicon and the complex language model for a range of plausible speech recognition tasks. The efficient structure is commonly built using an appropriate finite state machine (FSM) (Mohri, 1997) that represents the cross product of the acoustic features (from the input signal), the HMM states and units for each sound, the sound for each word, the words for each sentence, and the sentences – which are valid within the syntax and semantics of the language and task at hand (Juang *et al.*, 2004). For large vocabulary and high perplexity speech recognition tasks, the size of the recognition network can become astronomically large and prohibitive that they cannot be

exhaustively searched by any known method or machine. Fortunately, FSM methods such as dynamic programming (Jing *et al.*, 2010) can compile such large networks and reduce the size of the vocabulary significantly due to inherent redundancies and overlaps across each of the levels of the recognition network.

2.5. Classifications of ASR Systems

Speech recognition systems can be divided into three major categories (Huang *et al.*, 1993), namely, (1) speaker-dependent – a speech recognition system is said to be speaker-dependent if it needs to be tuned, or trained, for a specific speaker. In order to enable such a system to recognise speech of different speakers, it must be trained for all the new speakers, (2) speaker-independent – ASR systems that can recognise speech from many users without each user having to undergo a training phase and (3) speaker-adaptive – these systems can be trained, initially, for a set of users to provide some level of speaker independence, but be adaptable enough to provide speaker-dependent operation after training. Unfortunately, it is much more difficult to develop a speaker-independent system than a speaker-dependent one due to the large volume of training data required. Speaker-dependent systems can provide a significant word error rate (WER) reduction in comparison to speaker-independent systems if a large amount of speaker-dependent training data exists.

Besides being speaker-dependent, speaker-independent or speaker-adaptive, speech recognition systems can be classified according to the *continuousness* of their speech input (Whittaker *et al.*, 2001; Vimala and Radha, 2012), namely, (1) Isolated Speech Recognition — this is the simplest and least resource hungry mode a speech recognition engine can operate in. Each word is assumed to be preceded and succeeded by silence, i.e., both sides of a word must have no audio input, making word boundaries easy to detect and construct and (2) Continuous Speech Recognition — this mode allows the recognition of several words uttered continuously without pauses between them. Special methods must be used in order to determine word and phrase boundaries.

Whittaker *et al.* (2001) also demonstrated that the size of the vocabulary can be used to classify ASR systems into the following categories, (1) small – with a maximum of a thousand words, (2) medium – a minimum of 1K and a maximum of 10K words, (3) large – from 10K to about 100K words, (4) extra-large – more than 100K words, and (5) unlimited – which attempts to model all possible (permissible) words in a language.

Furthermore, speech recognition systems can be classified according to the *size* of their linguistic recognition units (Huang *et al.*, 1993; Huang *et al.*, 2001) as follows:

i. Word-based speech recognition uses a single word as a recognition unit. The recognition accuracy is very high because the system is free from negative side effects of co-articulation and word boundary detection. However, for continuous speech recognition, transition effects between words again cause recognition problems. Moreover, for a word-based recognition system, processing time and memory requirements are very high because there are many words in a language, which are the basis of the reference patterns.
ii. Phoneme-based speech recognition uses phonemes as the recognition units. While recognition accuracy decreases when using this approach, it is possible to apply error-correction using the ability to produce fast results with only a finite number of phonemes. There can be several speech recognition systems that make use of sub-word units based on monophones, diphones, triphones, and syllables.

2.6. Approaches to ASR

Klatt (1977) outlined the two general approaches to ASR: "knowledge-based" and "self-organizing" approach. The former refers to systems which are based on explicit formulation of knowledge about the characteristics of different speech sounds while the latter refers to systems where a much more general framework is used and the parameters are learned from training data.

2.6.1. Knowledge-based Approaches

In the early 1970s, the Advanced Research Projects Agency (ARPA) initiated the idea of developing ASR systems based on the explicit use of speech knowledge. This was done within the framework of the speech understanding project (Klatt, 1977). The project resulted in the development of a number of ASR systems. Several *artificial intelligence* techniques were applied to use *higher knowledge* such as lexical and syntactic knowledge or semantics and pragmatics to obtain an *acceptable* recognition rate. The resulting systems produced a very poor recognition rate, needed a lot of computational resources and were limited to the specific task to which they were designed. A fundamental deficiency with this kind of approach is that it is limited by the accuracy of the acoustic phonetic decoding.

Within the same context of knowledge-based approaches, several ASR systems have been developed on *expert systems* modelling of the humans' ability to interpret spectrograms or other visual representations of the speech signal (O'Brien, 1993). These kinds of systems separate the knowledge that is to be used in a reasoning process from the reasoning mechanism which operates on that knowledge. The knowledge is usually entered manually and is based on the existence of particular features such as "a silence followed by a burst followed by noise" for an aspirated voiceless stop. Using this kind of approach triggers the need for a vast amount of knowledge for speaker-independent continuous speech recognition of large vocabularies (Mariani, 1991). However, the large set of rules makes it difficult to imagine all of the ways in which the rules are interdependent.

2.6.2. Self-organizing Approaches

An alternative to the *knowledge-based approach* is the *self-organizing* which provides a general structure and allows the system to learn the parameters from a set of training data. The three most common self-organizing approaches to ASR are, namely,

Template Matching, Artificial Neural Networks (ANNs) and the most commonly used HMMs (Klatt, 1977; Vimala and Radha, 2012).

Template Matching

This is one of the simplest approaches to developing ASR systems. In a typical template matching approach, a template is generated for each word in the vocabulary to be recognized. The generated template is based on one or more examples of that word. The recognition process then proceeds by comparing an unknown input with each template using a suitable spectral *distance measure* (Rabiner and Gold, 1975; Klatt, 1977). The template with the smallest distance is output as the recognized word.

Artificial Neural Networks

One of the most commonly used examples of ANNs is the multi-layer perceptrons (MLPs) (Lippmann, 1989). An MLP consists of a network of interconnecting units, with two layers for input and output, and one or more hidden layers. A set of speech units to be recognized is represented by the output units and the recognition process relies on the weights of the connections between the units. The connection weights are trained in a procedure whereby input patterns are associated with output labels. The MLPs are therefore learning machines in the same way that HMMs. However, they provide the *advantage* that the learning process maximizes discrimination ability, unlike just accurately modelling each class separately (Trentin *et al.*, 2001). However, MLPs have a *disadvantage* in that, unlike HMMs, they are unable to deal easily with the time-sequential nature of speech. The problem with this approach is that it does not generalize to connected speech or to any task which requires finding the best explanation of an input pattern in terms of a sequence of output classes (Klatt, 1977; Vimala and Radha, 2012).

Hidden Markov Models

A hidden Markov model can be defined as a set of probabilistic functions of a Markov chain which involves two nested distributions, one pertaining to the Markov chain and the other to a set of the probability distributions, each associated with a state of the Markov chain (Wilpon *et al.*, 1990). The HMMs attempt to model the characteristics of a probabilistic sequence of observations that may not be a fixed function but instead change according to a Markov chain. The theory of HMMs has been extensively developed to create efficient algorithms for training (Expectation-Maximization, Baum-Welch re-estimation) and recognition (Viterbi, Forward-Backward) (Juang *et al.*, 1991). The HMMs are currently the predominant methodology for state-of-the-art speech recognition (Vimala and Radha, 2012; Besacier *et al.*, 2014).

A typical HMM is defined as follows (Wilpon *et al.*, 1990; Juang *et al.*, 1991; Huang *et al.*, 2001):

- $\boldsymbol{O} = \{O_1, O_2, \dots, O_N\}$ – An output observation alphabet. The observation symbols correspond to the physical output of the system being modelled.
- $\boldsymbol{S} = \{S_1, S_2, \dots, S_N\}$ – A set of all *N* states.
- $\boldsymbol{A} = \{a_{ij}\}$ – A transition probability matrix. An entry a_{ij} of A stands for the probability $P(S_i|S_j)$ that given state S_i, state S_j follows.
- $\boldsymbol{\pi} = \{\pi_i\}$ – An initial state distribution. Each state S_i has a certain probability $\pi_i = P(q_i = S_i)$ to be the starting state of a state sequence.
- $\boldsymbol{B} = \{b_i(x)\}$ – An output probability matrix. $b_i(x)$ is the probability that x is observed in state S_i.

The calculations are simplified by ensuring that state transitions only depend on the directly preceding states hence the name *Markov Models*. According to the start probabilities π_i a starting state S_1 is selected. With the probability a_{ij} the system changes from the current state S_i, to S_j. In each state S_i, the emission probabilities $b_i(x)$ are produced by some *hidden* random process according to which the most likely

observed feature is selected. The random process is *hidden* in the sense that only the output of $b_i(x)$ is observable, not the process producing it.

HMMs can be classified as either *discreet* or *continuous* (Huang *et al.*, 2001). *Discrete* HMMs have a discrete feature vector space. In this case, the emission probabilities $b_i(x)$ are given by probability tables over the discrete observable features V. For *continuous* HMMs, the feature vector space is continuous and the emission probabilities are now probability densities (Huang *et al.*, 2001). Usually the emission probabilities $b_i(x)$ are approximated by a Gaussian distribution with a mean value vector μ and the covariance matrix $\sum$ (Huang *et al.*, 2001):

$$b_i(x) = \sum_{i=0}^{L_i} c_{il} \,.\, Gauss(x|\mu_{il}, \sum_{il}) \tag{2.7}$$

$$\sum_{i=0}^{L_i} c_{il} = 1 \tag{2.8}$$

- L_i is the number of mixture distributions made is used in state S_i.
- c_{il} are the weight coefficients, called *Mixture Components*.

The Gaussian mixture distribution is defined as follows:

$$Gauss(x|\mu_{il}, \sum_{il}) = \frac{1}{\sqrt{(2\pi)^d|\sum|}} . e^{\frac{1}{2}(x-\mu)^T . \sum^{-1}(x-\mu)} \tag{2.9}$$

- d defines the dimensionality of the feature vector space.
- μ_{il} is the mean value vector and $\sum_{il}$ is the covariance, they are both referred to as the *Codebook* of the model

Solving the speech recognition problem with the HMM model can be summarised with three fundamental algorithms (Juang *et al.*, 1991; Huang *et al.*, 2001) as follows:

- The evaluation problem: focuses on the calculation of $P(O|\beta)$ – the probability that an observed feature sequence $O = o_1, o_2, \ldots, o_T$ was produced by the HMM model β. This problem can be tackled by the *Forward Algorithm*.
- The decoding problem – using the *Viterbi Algorithm*, the goal is to identify the most likely path $\mathbb{q}$ that produces the observed feature sequence O.
- With the optimization problem, the goal is to find the parameters β^* that maximises the probability to produce O given the model $\beta = (A, B, \pi)$, this is achieved by the *Baum-Welch Algorithm*, also known as the *Forward-Backward Algorithm*.

2.7. Modelling Units for Speech Recognition

Within the context of automatic speech recognition process, words are traditionally represented as a sequence of acoustic sub-word units such as phonemes (Killer *et al.,* 2003). The mappings from these sub-word units to words are usually contained in a pronunciation dictionary. The pronunciation dictionary provides a mapping to a sequence of sub-word units for each entry in the vocabulary (Stuker *et al.,* 2004). Phonemes are the most commonly used units for acoustic modelling of speech recognition systems (Stuker *et al.,* 2004; Kanthak *et al.,* 2003). The overall performance of ASR systems is strongly dependent on the accuracy of the pronunciation dictionary and best results are usually obtained with hand-crafted dictionaries (Kanthak *et al.,* 2003). Before the era of continuous speech recognition, words or morphemes were commonly used as sub-word units (Killer *et al.,* 2003).

Morphemes are meaningful linguistic units consisting of a word or a word element that cannot be divided into smaller meaningful parts and that are well fitted for a single word recogniser (Killer *et al.,* 2003). In continuous speech, there is a large amount of possible words and word combinations. It gets infeasible to write down all possible morphemes and it is not possible anymore to find enough training data for each such unit (Gorin *et al.,* 1996; Huang *et al.,* 1993; Killer *et al.,* 2003). The simplest way to split up words is to decompose them into their syllables (Huang *et al.,* 1993; Killer *et al.,* 2003).

Syllables model co-articulation effects between phonemes and capture the accentuation of a language (Gorin *et al.,* 1996; Killer *et al.,* 2003). Although syllables are limited in number, they are still too many to cause training problems (Killer *et al.,* 2003). The number of phonemes in a language is well below the number of possible syllables, usually ranging between 30 and 50 phonemes (Killer *et al.,* 2003). Phonemes are easily trainable and offer the advantage that a new word can be added very simply to the vocabulary (Gorin *et al.,* 1996; Killer *et al.,* 2003).

Most speech recognition systems are improved by looking at phonemes in their various contexts. Triphones are used when only the immediate left and right neighbours are considered (Besling*,* 1994, Gorin *et al.,* 1996; Black *et al.,* 1998; Killer *et al.,* 2003). Polyphones are used to model unspecified neighbourhood (Killer *et al.,* 2003).

2.8. Development of state-of-the-art Automatic Speech Recognition Systems

The state-of-the-art ASR systems generally use a standard HMM-based approach and involve two major phases, namely, model training phase and decoding phase (Young, 2008; Besacier *et al.*, 2014). Modern ASR systems commonly incorporate the HMM technique with a variety of other techniques to enhance the recognition accuracy and reduce recognition error rates. Such techniques include, Dynamic Bayesian Networks (DBN) (Stephenson *et al.*, 2002), Support Vector Machines (SVM) (Solera-Urena *et al.*, 2007), Dynamic Time Wrapping (DTW) (Jing *et al.*, 2010) and ANN (Seide *et al.*, 2011; Mohamed *et al.*, 2012). Figure 2.2 outlines a typical state-of-the-art ASR system. A large number of recruited speakers for doing speech recordings are usually required to create and improve acoustic models for large vocabulary speaker-independent ASR systems in the model training phase.

The model training phase involves training both acoustic and language models. Robust acoustic models must take into account speech variability with respect to environment, speakers and channel (Huang *et al.*, 2001). The LVCSR systems require a large textual data to generate robust language models. This is because statistical language models

are based on the empirical fact that a good estimation of the probability of a lexical unit can be obtained by observing it on large text data (Besacier *et al.*, 2014).

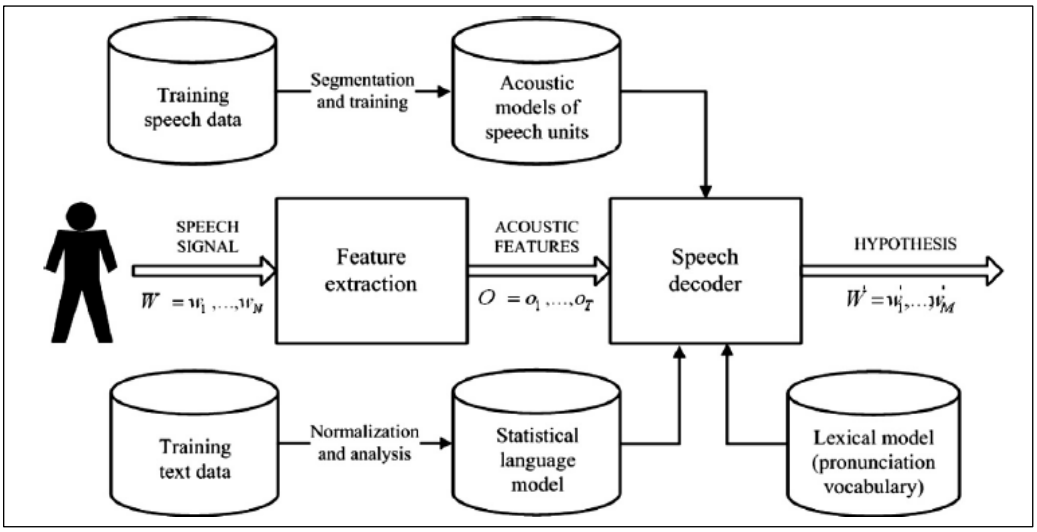

Figure 2.2: Components of a typical state-of-the-art ASR system (Besacier *et al*., 2014)

The decoding phase of state-of-the-art ASR systems integrates a speech decoder that is capable of generating *N*-best lists of words (or phonemes) as a compact representation of the recognition hypotheses and then to re-score them using robust statistical language models to output the best recognition hypothesis (Besacier *et al.*, 2014). At present, several state-of-the-art ASR decoders exist under open-source licences and can easily be adapted to any language of interest. Such decoders include: HTK, KALDI, Julius, RASR, and Sphinx, etc. (Besacier *et al.*, 2014).

2.9. Developing HMM-Based Speech Recognition Systems

There is a wide variety of approaches, techniques and toolkits for developing speech recognition engines. The discussion of all the different techniques is beyond the scope of this research study. We therefore discuss an overview of the most commonly used toolkit for developing HMM-based speech recognition engines, the Hidden Markov Model Toolkit (HTK) (Young *et al.,* 2006). The HTK is the toolkit used in all the training and recognition experiments in this research study.

2.9.1. An Overview of HMM-based Speech Recognition Engines

HMM-based speech recognition engines use HTK with a variety of configuration details to perform training and decoding/recognition of ASR systems. Generally, HMM-based speech recognition engines comprise of two major processing phases:

- **Training Phase** – this phase involves using the training tools to estimate the parameters of a set of HMMs using a set audio files and their associated transcriptions.
- **Recognition/Testing Phase** – HTK recognition tools are used to transcribe (generate text for) unknown utterances.

2.9.2. The Hidden Markov Model Toolkit

The HTK is a toolkit for building HMMs. It is primarily designed for building HMM-based speech processing tools, particularly, speech recognizers (Young *et al.,* 2006). The HTK is an open-source research toolkit that consists of command-line tools written in C language to construct various components of speech recognition systems. The HTK is very flexible and complete (always updated). Besides the tools provided for training and decoding, the toolkit also provides tools designed for data preparation and analysis.

2.9.3. Feature Extraction

The HTK provides a variety of feature extraction parameters (Young *et al.,* 2006). We name a few that are commonly used by most ASR researchers for task-appropriate recognition accuracy:

- LPC : linear prediction filter coefficients
- LPCEPSTRA : LPC cepstral coefficients
- MFCC : mel-frequency cepstral coefficients
- FBANK : log mel-filter bank channel outputs
- PLP : PLP cepstral coefficients

Each of these parameters can have additional qualifiers which are very well understood by HTK. The use of different qualifiers provides the privilege to extract different features which then yields varying recognition accuracies and correctness. It is a researcher's responsibility to try different combinations of these qualifiers to achieve better results. The possible qualifiers interpreted by HTK are (Young *et al.,* 2006):

- _E : has energy
- _N : absolute energy suppressed
- _D : has delta coefficients
- _A : has acceleration coefficients
- _C : is compressed
- _Z : has zero mean static coefficients
- _K : has CRC checksum
- _O : has 0^{th} cepstral coefficients
- _V : has VQ data
- _T : has third differential coefficients

2.10. Multilingual Speech Recognition

Multilingual speech recognition is a topic beyond the scope of this study, therefore only a summary with regards to the challenges and approaches is discussed. There are several successful approaches to multilingual speech recognition (Ulla, 2001), the different approaches depend on the goal of the application. Ulla (2001) clustered the approaches in the following three groups:

- Porting – this approach involves the porting of an ASR system designed for a specific language to another language. In this case, the ASR system is the same

for the target language and the training data are only of the new language. The original (source) system must be optimised for the new (target) language. Most of the source language algorithms must be adapted to conform to the target language. The ASR system for the target language is trained with data from the source language. The porting approach assumes that there is enough training data in the target language to establish a complete system.

- Cross-lingual recognition – unlike the porting approach, cross-lingual assumes that there's insufficient training data available for training the ASR system in the target language. Therefore, techniques are needed to allow the use of training material from a source language to model acoustics parameters of the target language. Occasionally, an adaption with few data from target language could take place (Ulla, 2001). The first step is to find the possible source language(s) to harvest the training material for the target language. An optimal language, the language yielding best recognition performance on the target language, must be identified. A relation between the source language(s) and the target language must also be identified. One such relation must be the most suitable acoustic units of the source and target language(s). The main problem is to determine the identical acoustic units or to model the existing acoustic units in a way that satisfactory recognition accuracies can be achieved (Ulla, 2001).
- Simultaneous multilingual speech recognition – this very different approach allows the recognition of utterances of different languages at the same time. The system, basically, does not know the language of an utterance. Training data is available in all languages and all languages are decoded by a single recognizer.

Research in the domain of ASR for under-resourced languages has focused on the efficient development of multilingual and cross-lingual grapheme-based ASR approaches that can make use of resources available in other languages. The use of multilingual grapheme models for rapid bootstrapping of acoustic models to new languages was investigated by Stuker (2008a; 2008b). Data driven mapping of grapheme sub-word units across languages was studied by Stuker (2008a). Stuker

(2008b), applied polyphone decision-tree based tying for porting decision trees to a new language for grapheme models. The study focused specifically on porting multilingual grapheme models to German and it was found to be beneficial compared to monolingual grapheme models when limited adaptation speech data for training is available.

Kanthak and Ney (2002) demonstrated that grapheme-based acoustic units in combination with decision tree state tying may reach the performance of phoneme-based units for at least a couple of European languages. The approach is driven by the acoustic data and does not require any linguistic or phonetic knowledge. Grapheme-based multilingual acoustic modelling already provides a globally consistent representation of acoustic unit set by definition (Kanthak *et al.,* 2003). Global phoneme representation sets such as Speech Assessment Method Phonetic Languages (SAMPA) or the International Phonetic Alphabet (IPA) (Schultz, 1998) may be used to express similarities between languages when using phoneme-based acoustic sub-word units. However, the use of context-dependent grapheme-based sub-word units eliminates the need to find common sets of acoustic sub-word units.

2.11. Robustness of Automatic Speech Recognition Systems

The recognition accuracy of ASR systems rapidly degrades when deployed in acoustical environments different than those used in training (Acero, 1993). The main cause is the mismatch between training and recognition spaces, which could result in the speech recognizer becoming completely unusable (Acero, 1993).

The training-testing mismatch is commonly caused by two major factors: additive noises and convolutional noises (Juang, 1991; 1992; Acero, 1993; Moreno, 1996). A great deal of attention has previously been paid to this problem in an effort to successfully deploy the technology in speech-enabled applications (Gales, 1992; Acero, 1993). Many approaches have been considered to enhance robustness in speech recognition systems. These includes techniques based on the use of special distortion measures,

autoregressive analysis, the use of auditory models, and the use of microphone arrays, among many other approaches (Juang, 1991; Gales, 1992; Acero, 1993).

There are two main ways to achieve robust speech recognition (Juang, 1992; Acero, 1993; Moreno, 1996):

- Acoustic model adaptation methods, which map acoustic models from training space to recognition space.
- Feature vector normalization methods, which map recognition space feature vectors to the training space.

The choice of a robustness technique depends on the characteristics of the application in each situation. In general, acoustic model adaptation methods produce the best results because they can reasonably model the uncertainty caused by the noise statistics (Neumeyer and Weintraub, 1995).

Well-known successful acoustic model adaptation methods include Maximum A Posteriori (MAP) (Gauvain, 1994), Maximum Likelihood Linear Regression (MLLR) (Leggeter, 1995), Parallel Model Combination (PMC) (Gales and Young, 1995), and Vector Taylor Series (VTS) (Moreno, 1996). However these methods require more training data and computing time than the feature vector normalization methods.

Most common and successful feature vector normalization method is known as, Cepstral Mean Normalization (CMN) (Liu, *et al.,* 1993). The CMN has been successfully used as a simple yet effective way of normalizing the feature space. It provides an error rate reduction under mismatched conditions and has also been shown to yield a small decrease in error rates under matched conditions. These benefits, together with the fact that it is very simple to implement, have seen many current systems adopting it (Manaileng and Manamela, 2013).

2.12. Conclusion

This chapter discussed the theoretical background of ASR. The historical background of the field was elaborated and the statistical framework of the technologies was thoroughly discussed. We further discussed the individual components that make up a typical ASR system. The various classifications of and approaches to ASR systems were also discussed. We further elaborated the approach that is commonly followed to develop HMM-based and state-of-the-art speech recognition systems.

3. RELATED BACKGROUND STUDY

3.1. Introduction

As previously stated, there is little documented knowledge and information on most of under-resourced languages and hence they lack advanced modern linguistic and computational tools. The speech processing research community has been concerned with porting, adapting, or creating written and spoken resources or even models for low-resourced languages (Besacier *et al.*, 2014). Besacier *et al.* (2014) also notes the several adaptation methods that have been proposed and experimented with, and also the workshops and special sessions that have been organized on this issue.

3.2. Previous Studies on Grapheme-based Speech Recognition

In cases were no expert knowledge is available or affordable for hand-crafting a pronunciation dictionary, new methods are needed to automate the process of creating the pronunciation dictionary. However, even the automatic tools often require hand-labelled training materials and rely on manual revision and verification.

There are several different methods to automate the process of creating the pronunciation dictionary that have been introduced in the past (Besling, 1994; Black *et al.*, 1998; Singh *et al.*, 2002; Kanthak and Ney, 2003). Most of the time these methods are based on finding rules for the conversion of the written form of a word to a phonetic transcription, either by applying rules (Black *et al.*, 1998) or by using statistical approaches (Besling, 1994). Some of the methods have been investigated in the field of ASR (Singh *et al.*, 2002; Kanthak and Ney, 2002).

Recently, the use of graphemes as modelling units – instead of phonemes, has been increasingly studied. Graphemes have the advantage over phonemes in that they make the creation of the pronunciation dictionary a trivial task. Creating grapheme-based dictionaries does not require any linguistic knowledge (Stuker and Schultz, 2004). However, graphemes have a generally looser relation to pronunciation, i.e.,

pronunciation is not immediately related to orthography. As such, it becomes important to use context-dependent acoustic modelling techniques and parameter sharing for different models (Kanthak and Ney, 2002; Stuker and Schultz, 2004).

The quality of grapheme-based ASR systems depends significantly on the grapheme-to-phoneme relation of the language, that is, the degree of relatedness between how words are pronounced (articulation) and how they are written (orthography) (Kanthak and Ney, 2003; Killer *et al.*, 2003). This has been demonstrated by prior experiments (Schukat-Talamazzini *et al.*, 1993; Kanthak and Ney, 2002; Black and Llitjos, 2002).

Schukat-Talamazzini *et al*. (1993) and Kanthak and Ney (2002) were some of the first researchers to present results for speech recognition systems based on the orthography of a word. Kanthak and Ney (2002) further suggested the use of decision trees for context-dependent acoustic modelling. Black and Llitjos (2002) successfully relied on graphemes for text-to-speech systems in minority languages. Kanthak and Ney (2003) and Killer *et al.* (2003) investigated the use of graphemes for languages with phoneme-grapheme relations of differing closeness and in the context of multilingual speech recognition. Sirum and Sanches (2010) studied the effect on WER for Portuguese when the acoustic units based in phonemes and graphemes are compared. Whereas, Basson and Davel (2013) investigated the strengths and weaknesses of grapheme-based and phoneme-based acoustic sub-word units using the Afrikaans language as a case study. They developed a grapheme-based ASR system alongside a phoneme-based ASR system using the same standardised approach, except that in the one case they used tied-state triphones and the other, tied-state trigraphemes.

All these experiments have shown that graphemes may be suitable modelling units for speech recognition of some languages and not others. However, the use of grapheme-based pronunciation dictionaries does not yield any pronunciation variants. Therefore, the variations in pronunciation of the same word have to be modelled implicitly in the parameters of the units used, as it is the case with the differences in pronunciation of

the different graphemes depending on their orthographic context (Kanthak and Ney, 2003).

3.3. Related Work in ASR for Under-resourced Languages

3.3.1. Definition of Under-resourced Languages

Krauwer (2003) was one of the first people to introduce the concept of “under-resourced languages”. He referred to them as languages with some of (if not all) the following aspects: lack of a unique writing system or stable orthography, limited presence on the world wide web, lack of linguistic expertise, lack of speech and language processing electronic resources, such as monolingual corpora, bilingual electronic dictionaries, transcribed speech data, pronunciation dictionaries, vocabulary lists, etc. The term is often used interchangeably with: resource-poor languages, less-resourced languages, low-data languages or low-density languages.

The concept of under-resourced languages should not be confused with that of minority languages - which are languages spoken by a minority of the population of a territory (Krauwer, 2003). Some under-resourced languages are actually official languages of their country of origin and are spoken by a very large population. However, some minority languages can often be considered as rather well-resourced. Consequently, under-resourced languages are not necessarily endangered, whereas minority languages may be endangered (Crystal, 2000).

3.3.2. Languages of South Africa

South Africa is a highly linguistically diversified country with eleven official languages, four official race groups and very wide social and cultural disparities. Figure 3.1 shows the distribution of languages across the population, the bold-font languages are those of focus in the Telkom Centre of Excellence for Speech Technology (TCoE4ST) at the University of Limpopo.

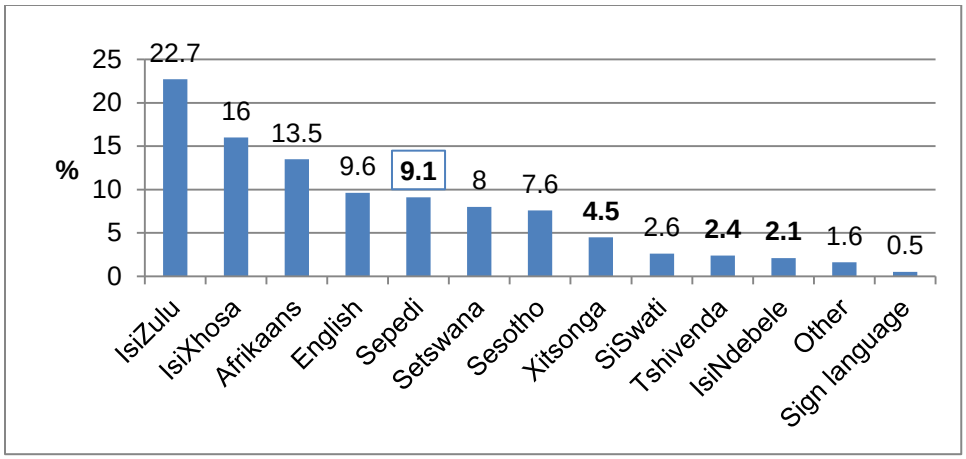

Figure 3.1: South African languages and focus area

Various speech corpora for South African languages have been released in recent years, including the LWAZI telephone speech corpora (Barnard *et al.*, 2009) and National Centre for Human Language Technologies (NCHLT) (De Vries *et al.,* 2013) speech corpora – a substantially large set of broadband speech corpora. These corpora are all focused on speech data from the eleven official languages of South Africa. In recent years, speech and language technology projects that attempt to bridge language barriers while also addressing socio-linguistic issues, have achieved substantial attention and developments mileage in South Africa (Barnard *et al.,* 2010).

All eleven official languages of South Africa do occur in Limpopo Province – most of them being spoken by a few people (Stats SA, 2010). Most languages in this province are considered under-resourced due to the scarcity of speech processing tools such as pronunciation dictionaries and computational linguistic experts, in most cases. Although Barnard *et al.* (2009) and De Vries *et al.* (2013) accounts for all eleven languages in terms of speech corpora, researchers often encounter problems regarding language dialects and perfectly hand-crafted pronunciation dictionaries.

3.3.3. Approaches to ASR for Under-resourced Languages

Feature extraction is one of the most important components of ASR systems. Acoustic features must be robust against environmental and speaker variations, and to some extent, be language independent. The kind of features to be extracted becomes

immediately important in the context of ASR for under-resourced languages because only small amounts of data are generally available and at times speech data must be shared across multiple languages for efficient bootstrapping of systems in unseen languages (Besacier *et al.*, 2014). Studies suggest that multilayer perceptrons (MLP) features extracted from one or multiple languages can be successfully applied to other languages (Stolcke *et al.*, 2006; Toth *et al.*, 2008; Plahl *et al.*, 2011). Thomas *et al.* (2012) and Vesely *et al.* (2012) also demonstrated that the use of data from multiple languages to extract features for an under-resourced language can improve ASR performance.

Due to the difficulty usually encountered in transcribing speech from under-resourced languages, researchers have proposed lightly-supervised and unsupervised approaches for this task. An unsupervised adaption technique was proposed for the development of an isolated word recognizer for Tahil (Cetin, 2008). Several other similar and extended techniques have been explored for a variety of languages, such as Polish (Loof *et al.*, 2009) and Vietnamese (Vu *et al.*, 2011). These kinds of techniques have proven to be useful in saving time and costs required to build ASR systems for unsupported languages if prior information such as pronunciation dictionaries and language models about the target languages exist. Vu *et al.* (2011) demonstrated that such techniques are useful even when the target language is unrelated to the source language.

The development of ASR systems for under-resourced languages commonly uses similar techniques as those in well-resourced languages such as, the use of context-dependent HMMs to model the phonemes of a language. However, this approach raises interesting challenges in the context of under-resourced languages. For instance, Wissing and Barnard (2008) suggested that defining an appropriate phone set to model is a non-trivial task since even when such sets have been defined they often do not have an empirical foundation. Also, putative phonemes such as affricates, diphthongs and click sounds may be modelled as either single units or sequences, while allophones, which are acoustically too distinct may be modelled separately (Besacier *et*

al., 2014). However, solutions to these issues can be tackled with guidance from the choices made in closely related languages. For instance, when a closely related well-resourced source language is available, it is often possible to use data from that language in developing acoustic models for an under-resourced target language.

A variety of approaches have been used in this regard, such as bootstrapping from source-model alignments (Schultz and Waibel, 2001; Le and Besacier, 2009), pooling data across languages (van Heerden *et al.*, 2010) and phone mapping for recognition with the source models (Chan *et al.*, 2012). Some investigators proposed the use of some variation of the standard context-dependent HMMs by using HMMs to rather model syllables instead of phonemes (Tachbelie *et al.*, 2012, 2013). This approach reduces model parameters because context dependencies are generally less important for syllable models. Some researchers however, have proposed the use of alternative phoneme modelling techniques all together. For example, Gemmeke (2011) used exemplar-based speech recognition where the representations of acoustic units (words, phonemes) are expressed as vectors of weighted examples. Siniscalchi *et al.* (2013) proposed to describe any spoken language with a common set of fundamental units that can be defined "universally" across all spoken languages. In this case, speech attributes such as manner and place of articulation are chosen to form this unit inventory and used to build a set of language-universal attribute models derived from IPA (Stuker *et al.*, 2003) or with data-driven modelling techniques. The latter work proposed by Siniscalchi *et al.* (2013) is well suited for deep neural network architectures for ASR (Yu *et al.*, 2012).

3.3.4. Collecting or Accessing Speech Data for Under-resourced Languages

The current development of most ASR systems for well-resourced languages uses statistical modelling techniques which require enormous amounts of data (both speech and text) to build pronunciation dictionaries and robust acoustic and language models (Besacier *et al.*, 2014). However, most under-resourced languages have no existing speech corpora and hence data collection is the most important part of ASR

development. The speech data collection process for under-resourced languages is inarguably a very difficult task. Various approaches for speech data collection in under-resourced language have been explored, two most common being the use of existing audio resources and the recording of audio data from scratch (Besacier *et al.*, 2014).

The use of existing audio sources involves collecting speech data from a variety of sources such as, recordings of lectures, parliamentary speeches, radio broadcasts (news), etc. The main challenge with this approach is the transcription of the recordings so that they are rendered useful for ASR development. However, due to the scarcity of linguistic experts in most under-resourced languages, the difficult manual transcription becomes inevitable. Also, many under-resourced languages do not have well-standardized writing systems (Crystal, 2000). Alternative transcription approaches such as crowd-sourcing have been used successfully (Parent and Eskenazi, 2010). However, for most under-resourced languages, the number of readily available transcribing workers is limited (Gelas *et al.,* 2011). Furthermore, existing sources are generally dominated by fewer speakers while a typical *speaker-independent* ASR corpus requires at least fifty different speakers (Barnard *et al.*, 2009).

In contrast, speech data can be recorded from scratch. This approach can significantly simplify the transcription process since pre-defined prompts can be used. The challenge however, is finding potential speakers and recording them. A text corpus must first be collected. This process assumes a standardized writing system for the language. Prompts may be extracted from the text corpus and systematically and conveniently presented to particular speakers (preferably first language speakers) for recording purposes. Manual verification is often required to ensure that the desired words have been spoken. However, alternative automated methods have been used successfully and efficiently. For example, Davel *et al.*, (2011) used a raw corpus to bootstrap an ASR system, with an assumption that all prompts have been correctly recorded, and used to iteratively identify misspoken utterances and improve the accuracy of the ASR system. The recording process often involves the use of menu-driven telephony services, such as Interactive Voice Response (IVR) systems (Muthusamy and Cole, 1992).

Alternatively, with the use of a tape recorder or a personal computer, recordings can be obtained during a face-to-face recording session where a field worker can provide instructions in person (Schultz, 2002). With the widespread availability of smartphones, researchers have continually developed smartphone applications for a much more flexible speech recording task (Hughes *et al.*, 2010; De Vries *et al.*, 2013).

Although, spontaneous speech can also be collected using most of these platforms (Godfrey *et al.*, 1992), such speech corpora are usually less useful for the development of baseline ASR systems for under-resourced languages (Besacier *et al.*, 2014). This is normally due to resource constraints, small corpora are generally created for under-resourced languages and clear pronunciation of prompted speech is required for such corpora.

3.3.5. Challenges of ASR for Under-resourced Languages

Developing HLT systems for under-resourced languages is indeed a mammoth task with multi-disciplinary challenges. Resource acquisition requires innovative methods (such as those mentioned in the previous, e.g., crowd-sourcing) and/or models which allows the sharing of acoustic information across languages as in multilingual acoustic modelling (Schultz and Waibel, 2001; Schultz, 2006; Le and Besacier, 2009). Porting an HLT system to an under-resourced language requires much more complicated techniques than just the basic re-training of models. Some of the new challenges that arise involve word segmentation problems, unwritten languages, fuzzy grammatical structure, etc. (Besacier *et al.*, 2014). Moreover, the target languages usually introduce some socio-linguistic issues such as dialects, code-switching, non-native speakers, etc.

Another major challenge is finding and accessing both the target language experts (speakers and practitioners) and speech processing technology experts. It is very unlikely in under-resourced languages to find native language speakers with required technical skills for ASR development. Furthermore, under-resourced languages very often do not have sufficient linguistic literature. Thus, system bootstrapping requires

borrowing linguistic resources and knowledge from similar languages. Such a task can be achieved with the help of dialects experts and phoneticians (to map phonetic inventories between target (under-resourced) language and the source (well-resourced) language).

3.4. Conclusion

This chapter gave a brief discussion of the background of ASR for under-resourced languages in relation to our study. We have explored the previous studies on grapheme-based speech recognition. We discussed the common approaches to and the challenges facing ASR research for under-resourced languages. We also discussed the common methods used to collect training and testing data for ASR in under-resourced language scenarios.

4. RESEARCH DESIGN AND METHODOLOGY

4.1. Introduction

An overview of the technologies used to develop state-of-the-art ASR systems were given in the previous chapters. The basic components of ASR systems were also discussed. The acoustic modelling component of ASR systems and alternative acoustic modelling units were explored. Some of the methods for collecting and/or accessing existing speech data and the approaches to developing ASR systems for under-resourced languages were also discussed. This chapter provides a framework on which the study is based. We briefly overview the research approach and the design that seeks to enable this research framework.

In this study, we follow the approach of using alternative acoustic modelling units, graphemes, instead of using the existing ones, phonemes. That is, we use graphemes as acoustic sub-word units instead of phonemes, for both pronunciation and acoustic modelling. We also use existing speech corpora as opposed to collecting speech data from scratch. Collecting speech data from scratch becomes redundant and inefficient if there is an existing corpus for the language of interest.

4.2. Experimental Design

Our proposed research approach was designed to explore the methods of minimising the cost, time and complexity of creating hand-crafted pronunciation dictionaries for ASR systems. The overall experiments involve two competing linguistic units used for acoustic modelling, namely, phonemes and graphemes. Complete and functional ASR systems were developed for each of the three selected under-resourced languages. For each language, two ASR systems were developed, each with two recognition experiments. The script in Appendix A was used to conduct all the recognition experiments.

The first ASR systems for each language were developed using the Lwazi ASR speech corpus (Meraka-Institute, 2009; Barnard *et al.*, 2009; van Heerden *et al.*, 2009) and the second ones using the NCHLT ASR speech corpus (van Heerden *et al.*, 2013; Barnard *et al.*, 2014). Furthermore, each ASR system had two experiments, the phoneme-based experiment (ExpPho), and the grapheme-based experiment (ExpGra). The purpose of both experiments is to attain superior recognition accuracies, and a significantly reduce the WER. The ExpPho used the phoneme-based pronunciation dictionaries, since it uses phonemes as acoustic sub-word modelling units. In contrast, the ExpGra used the grapheme-based dictionaries, using the phonemes counterpart, namely, graphemes as modelling units. Part of the research objectives was to train a multilingual ASR system for the three languages using the two approaches. However, for the purpose of a reasonable (scalable) project scope, a multilingual system was developed using only the Lwazi corpora and consequently adding two more experiments. The ultimate number of experiments is 14, i.e., 3 ExpPho for the Lwazi monolingual corpora, 1 for Lwazi multilingual speech corpus and 3 for NCHLT speech corpora with each ExpPho having its corresponding ExpGra counterpart. The primary purpose of using two different sets of speech corpora was to verify the results they produce and also validate the research hypothesis.

4.2.1. Speech Data Preparation

As previously alluded to, speech data collection for under-resourced languages can be a very cumbersome task. Recording quality speech data from scratch is very time-consuming and can be costly. The task can be a big research project on its own. It is therefore recommended that in the absence of a new corpus, which is often the case in under-resourced languages, researchers should use existing speech corpora, or at least use alternative existing speech data sources, such as parliamentary speeches, radio broadcasts (news), etc. It is for this reason that existing ASR speech corpora were used in this study, namely, Lwazi ASR speech corpus and the NCHLT ASR speech corpus.

The Lwazi ASR Speech Corpus

Lwazi is a HLT project commissioned by the South African national Department of Arts and Culture whose objectives included, amongst others, the development of core HLT resources for all the official languages of South Africa (Badenhorst *et al.*, 2011). The core HLT resources required for the development of ASR and TTS systems were developed for all the eleven official languages. Most of these languages had no prior HLT resources available. For each language, phone sets, new pronunciation dictionaries, and speech and text corpora were developed (van Heerden *et al.*, 2009; Badenhorst *et al.*, 2011). The speech and text data sets obtained from Lwazi (Meraka-Institute, 2009) are presented in Table 4.1.

TABLE 4.1: THE TRAINING AND EVALUATION DATA SETS FROM THE LWAZI CORPORA

Language	# of Speaker		# of Utterances		Duration (Hours)	
	Train	Test	Train	Test	Train	Test
SEPEDI	120	20	4512	1128	6.96	1.74
ISINDEBELE	119	20	4810	1203	7.52	1.88
TSHIVENDA	120	39	4751	1188	5.1	1.2

The data was partitioned into training and testing sets using the 80:20 ratio. This was achieved by using cross-lingual data sharing. Both phonetic and acoustic data was shared across the languages and the performance of the two approaches was investigated. The individual data sets were combined to create a multilingual corpus, outlined in Table 4.2. As a result, the pronunciation dictionaries were combined and pronunciation variants were retained form each language. We used the same phone representation notation, X-SAMPA, as the original pronunciation dictionaries in Davel (2009).

Table 4.2: THE TRAINING AND EVALUATION DATA SETS OF THE MULTILINGUAL CORPUS

	Train	Test	Total
# of Speaker	359	79	438
# of Utterances	14073	3519	17592
Duration (Hours)	19.58	4.85	24.43

The NCHLT ASR Speech Corpus

The NCHLT project is an extension of the Lwazi project. The project intended to support the development of practically useful large-vocabulary speech recognition systems (Barnard *et al.*, 2014). The corpora contains wide-band recordings of read speech made from a close-talking microphone, along with lexicons and significant text corpora, which are suitable for statistical language modelling (Barnard *et al.*, 2014). The data is also available for all eleven official languages of South Africa. Each language has above 56 hours of speech data. The speech data is available with the associated XML-transcriptions files partitioned as training and an evaluation set with 8 speakers for all languages (4 males and 4 females). The script for extracting the transcriptions is presented in Appendix B.

The training and evaluation data for the three languages was obtained from the NCHLT corpora. The experimental speech data setup for each language is outlined in Table 4.3.

Table 4.3: THE TRAINING AND EVALUATION DATA SETS FROM THE NCHLT CORPORA

Language	**# of Speaker**		**# of Utterances**		**Duration (Hours)**	
	Train	Test	Train	Test	Train	Test
SEPEDI	202	08	56284	2829	46.3	2.5
ISINDEBELE	140	08	39415	3108	46.5	4.2
TSHIVENDA	110	08	33327	2805	33.1	2.7

4.2.2. Pronunciation Dictionaries

As in the case of speech data collection, hand-crafting pronunciation dictionaries for under-resourced languages can be a cumbersome task. The linguistic expertise is not always available and/or it's very expensive. This is what necessitates the use of graphemes, in substitution of phonemes, as sub-word acoustic modelling units. Fortunately, the speech corpora used were released with their respective pronunciation dictionaries. Therefore, our part was to obtain the available pronunciation dictionaries for the respective languages and perform appropriate modifications as discussed in the following sections.

Lwazi Pronunciation Dictionaries

The phoneme-based pronunciation dictionaries were also obtained from the LWAZI project. All the words in the pronunciation dictionaries were manually verified and correctly checked for phoneme representation redundancies. The dictionaries used are the original versions of the Lwazi pronunciation dictionaries (Davel, 2009) which contain no pronunciation variants. For the multilingual experiments, the monolingual dictionaries were combined and duplicate words with identical pronunciations were removed by simply sorting the dictionary into unique words. Multilingual speech recognition was not a central focus of this research and thus the techniques and approaches of generating multilingual pronunciation dictionaries were not thoroughly explored. The resulting pronunciation dictionaries are indicated in Table 4.4. The bottom row indicates the multilingual system, abbreviated MULTI-LING.

TABLE 4.4: THE LWAZI PRONCIATION DICTIONARY SETUP PER LANGUAGE

Language	Unique Words	Mono-phones	Mono-graphemes
SEPEDI	3317	43	27
ISINDEBELE	4754	48	27
TSHIVENDA	2490	41	30
MULTI-LING	10184	55	32

There are also more letters shared across the three languages than there are phonemes, noted from MULTI-LING. It can be noted that about 22 of the total 32 graphemes are shared across the languages. Moreover, about 69% of the total graphemes are uniformly distributed across the languages, i.e., 69% of the graphemes appear in all the languages. Conversely, only 24 of the total 55 monophones are shared across the languages.

This is an encouraging distribution and it is what makes graphemes much easier and less costly to use as sub-word acoustic modelling units than phonemes in the selected languages. This uniform distribution of graphemes across languages is one of the motivating reasons to use context-dependent grapheme-based sub-word units for multilingual acoustic modelling. However, this graphemic data sharing approach will

only hold for phonetically related languages. This is because languages with a similar phonetic structure also have a similar syntactic structure and thus have a similar grapheme set.

NCHLT Pronunciation Dictionaries

The used NCHLT phoneme-based pronunciation dictionaries were also obtained from the NCHLT. The dictionaries used are also the original versions by Davel *et al.* (2013) and contains no pronunciation variants. The NCHLT pronunciation dictionaries do not contain all the words appearing in the NCHLT ASR corpus transcriptions. Consequently, the missing words were manually added to the dictionary and the pronunciations were modelled following the NCHLT phone sets.

The details of the pronunciation dictionaries are outlined in Table 4.5. Details regarding the development of the phoneme-based dictionary can be found in (Davel *et al.*, 2013).

TABLE 4.5: THE NCHLT PRONUNCIATION DICTIONARY SETUP PER LANGUAGE

Language	Unique Words	Mono-phones	Mono-graphemes
SEPEDI	11721	45	27
ISINDEBELE	16250	51	26
TSHIVENDA	18073	39	32

Generating Grapheme-based Pronunciation Dictionaries

All existing phoneme-based pronunciation dictionaries were converted to grapheme-based dictionaries. To ensure the minimal time, linguistic knowledge and cost required for generating the dictionaries, the conversion did not follow any predetermined rules. We strictly used the most straightforward method of generating pronunciation dictionaries as words with their sequences of graphemes and thus directly using orthographic sub-word units as acoustic models (Killer *et al.,* 2003; Basson and Davel, 2013).

The wordlists were obtained from the existing phoneme-based pronunciation dictionaries. An alternative method would be to derive lists of words directly from transcriptions, but we wanted to guarantee the same size of vocabulary in both (phoneme and grapheme) dictionaries. The simple procedure to generate the grapheme-based dictionaries was as follows:

i. extract all words from a given pronunciation dictionary,
ii. append all words to a list,
iii. for every line in the list, segment the word into its constituent letters to serve as acoustic realization (pronunciation),
iv. write the list to a file, and
v. sort the file and retain only unique words to remove redundancies.

The final generated file is the actual dictionary. Just like the phoneme-based dictionaries, the grapheme-based dictionaries also do not cater for any pronunciation variants. The scripts for generating the wordlists and creating the grapheme-based pronunciation dictionaries are given in Appendix D and E, respectively.

Handling Foreign Words

Both the NCHLT and the Lwazi corpora contain some English words which are not in the dictionaries. Furthermore, the three languages have words originally borrowed from other languages (loan words), which are now generally used as primary words. More often, such words mixes the spelling and pronunciation conventions of the primary language with the other. For example in Sepedi, the word *Janaware* was originally loaned from the English *January*. *January* translates to *Pherekgong* in Sepedi but speakers still prefer using *Janaware* instead. The same example follows in Tshivenda, *January* translates to *Phando* but speakers prefer to use *Januwari* instead. In addition, it also translates to *Tjhirhweni* in IsiNdebele, but speakers use *Janibari* instead. Moreover, there are loan words which do not have their indigenous counterpart, e.g. *airtime*.

It is generally very difficult to model loans words in any typical phoneme-based monolingual and/or multilingual speech recognition task. In this research study, loans words were modelled using the primary language letter-to-sound rules, i.e., for each language, loans words are dealt with as if they belong to the language. This means that for all languages, the letter-to-sound rules of primary language were applied on the wordlist to predict pronunciations. The rules were developed at Meraka Institute of the CSIR and are available with every pronunciation dictionary for each language. Modipa and Davel (2010) showed that using this approach can achieve better recognition performance when dealing with English and Sepedi.

The grapheme-based pronunciation modelling of loan words is fairly simple since all words are simply separated into their constituent letters. For example, *airtime:* is modelled as *a i r t i m e* and *american* as *a m e r I c a n*. However, this is a disadvantage to the grapheme-based system since graphemes are generally not the ideal acoustic modelling units for most non-phonetic languages, such as English (Killer *et al.*, 2003; Janda, 2012). English words are very problematic when using graphemes as acoustic modelling units, for example the word: *address* is phonetically modelled as *E D r E s*, to provide the acoustic realization of the consecutive letters *dd* as phone *D* and *ss* as *S*. However, graphemes do not provide the acoustic variability between *s* and *ss*. This is a good example of why graphemes are not suited for acoustic modelling of non-phonetic languages.

The number of monophones and monographemes exclude *sil*, the silence phone. As previously alluded, the phoneme-based dictionaries contain a number of foreign (South African English and others) words that are commonly borrowed and used (code-switched) with these languages. Examples of such words include: first and/or second names, street names, names of places, time and dates, months, numbers and some general English words. This resulted in unique foreign graphemes which then increase the number of fundamental graphemes for each language.

4.2.3. Extracting Acoustic Features

The final stage of data preparation involved the process of acoustic feature extraction from the speech waveform. The feature extraction process was aimed to find a set of properties of an utterance that have acoustic correlations to the original speech signal, that is, parameters that can somehow be computed through processing of the signal waveform to estimate the original speech signal. The process is expected to ignore information that is irrelevant to the task and only keeping the useful information. It includes the process of measuring some important characteristic of the signal such as energy or frequency response, augmenting these measurements with some perceptually meaningful measurements (i.e., signal parameterization), and statically conditioning these numbers to form observations (Huang *et al.*, 2001).

For acoustic features, we extracted commonly used Mel-frequency cepstral coefficients (MFCCs) and compute delta features. These feature extraction configurations are reflected in Figure 4.1.

```
CEPLIFTER       =   22
ENORMALISE      =   FALSE
NUMCEPS         =   12
NUMCHANS        =   26
PREEMCOEF       =   0.97
SAVECOMPRESSED  =   FALSE
SAVEWITHCRC     =   FALSE
SOURCEFORMAT    =   WAVE
TARGETKIND      =   MFCC_0_D_A_Z
TARGETRATE      =   100000.0
USEHAMMING      =   TRUE
WINDOWSIZE      =   250000.0
ZMEANSOURCE     =   TRUE
LOFREQ          =   150
HIFREO          =   4000
```

Figure 4.1: The configuration file of the standard MFCC feature extraction technique

The features in all experiments were extracted with the same TARGETKINDS = MFCC_0_D_A_Z. Each feature vector has size 12 MFCC coefficients, one zeroth cepstral coefficients (_0), 13 delta coefficients (_D), 13 acceleration coefficients (_A), and zero mean static coefficients (_Z). The total number of coefficients amounted to 39

per feature vector. The standard feature extraction technique was enhanced with Cepstral Mean Variance Normalisation (CMVN) (Liu *et al.*, 1993; Viikki *et al.*, 1998). Figure 4.2 below outlines the resulting configuration file.

```
CEPLIFTER           =   22
ENORMALISE          =   FALSE
NUMCEPS             =   12
NUMCHANS            =   26
PREEMCOEF           =   0.97
SAVECOMPRESSED      =   FALSE
SAVEWITHCRC         =   FALSE
SOURCEFORMAT        =   WAVE
TARGETKIND          =   MFCC_0_D_A_Z
TARGETRATE          =   100000.0
USEHAMMING          =   TRUE
WINDOWSIZE          =   250000.0
ZMEANSOURCE         =   TRUE
LOFREQ              =   150
HIFREQ              =   4000
HPARM:CMEANDIR      =    'cmn_vectors'
HPARM:CMEANMASK     =   'audio/???/?????_???_%%%%_%%%%.wav'
HPARM:VARSCALEDIR   =   'cvn_vectors'
HPARM:VARSCALEMASK  =   'audio/???/?????_???_%%%%_%%%%.wav'
HPARM:VARSCALEFN    =   'cvn_vectors/globvariance'
```

Figure 4.2: The configuration file of the CMVN feature extraction technique

The CMVN technique is a combination of two robustness techniques, namely, Cepstral Mean Normalisation (CMN) (Liu *et al.*, 1993) and Cepstral Variance Normalisation (CVN) (Viikki *et al.*, 1998). Using the CMVN technique, we performed normalisation by first, (i) extracting features the normal way, (ii) estimating the cluster-means (CMN) and cluster-variances (CVN), and then (iii) extracting features again with normalisation given CMN and CVN, hence CMVN. The CMVN method, unlike the normal MFCCs, produces features that guarantee robust speech recognition (Manaileng and Manamela, 2013).

4.2.4. Model Training: Generating HMM-based Acoustic Models

Robust acoustic models were generated with every individual experiment; triphone acoustic models were generated for the phoneme-based ASR systems and trigrapheme models were generated for the grapheme-based systems. For the purpose of a clear discussion, we discuss the procedure of training a trigrapheme system rather than the

procedure for the two approaches. The procedure is almost identical to that of training a triphone system, except it models graphemes instead of phonemes.

Generating Decision Trees

For the phoneme-based ASR systems, HMMs were generated by using phonetic decision trees to perform clustering of tied-state triphones for continuous density mixture Gaussians. The grapheme-based systems used graphemic decision trees to perform clustering of tied-state trigraphemes. This was achieved by directly applying decision-tree based state-tying to the orthographic representation of words (Kanthak and Ney, 2003). The estimation of decision trees takes into account the complete acoustic training data as well as a list of possible questions to control splitting of tree nodes (Beulen *et al.*, 1997; Kanthak and Ney, 2003).

Since we are using grapheme-based sub-word units, we simply ask graphemes the questions, i.e., questions are asked about the left and right contexts of each trigrapheme, as shown in Example 4.1, and estimate a graphemic decision tree. Appendix C outlines the script used to generate the question files used to create the decision trees.

(4.1.)
```
QS  “R_a”       { *+a }
QS  “R_b”       { *+b }
QS  “R_c”       { *+c }
……
QS  “L_a”       { a-* }
QS  “L_b”       { b-* }
QS  “L_c”       { c-* }
```

This procedure is similar to that of phonetic sub-word units which asks the phonemes the questions, outlined in Example 4.2, and then estimates the phonetic decision tree.

(4.2.)
```
QS  “R_B”       { *+B }
QS  “R_BZ”      { *+BZ }
QS  “R_D”       { *+D }
QS  “R_E”       { *+E }
……
QS  “L_B”       { B-* }
QS  “L_BZ”      { BZ-* }
QS  “L_D”       { D-* }
```

QS “L_E” { E-* }

The phoneme-based approach, by definition, may at times require the assistance of an expert phonetic knowledge to define the question sets used to estimate the phonetic decision tree. Conversely, the grapheme-based approach requires no phonetic expertise for definition of the question sets. The resulting trees are automatically generated by learning the questions from the acoustic training data. The need for phonetic knowledge becomes completely trivial. One major advantage of using decision tree clustering is that it allows the recognition of previously unseen triphones and/or trigraphemes. Furthermore, context-dependent acoustic sub-word units in combination with decision tree state-tying guarantees detailed acoustic models which improved recognition performance.

The Procedure for Generating HMM-based Acoustic Model

The model generation procedure is identical for the two approaches, with the only difference being the sub-word units being used, e.g., monophones are used for the phoneme-based approach whereas monographemes are used for the grapheme-based approach. We therefore discuss only the phoneme-based approach to avoid repetitions.

The first step of the procedure is to define a prototype model with initial guesses of the parameters. The purpose of the prototype model is to define the model topology, which is a 3-state left-right with no skips. In summary, the HTK tool *HcompV* is used to scan all training data files, compute the global mean and variances and then sets all the Gaussians in the prototype model to have the same mean and variance. This will create a new version of the prototype model and store it in the *hmm_0* directory. It is from this prototype model that the initial parameters of all the monophone HMMs (including *sil*) are estimated.

The next step is to use the *Baum-Welch* re-estimation algorithm to re-estimate the flat start monophones. This is achieved by invoking the HTK-embedded re-estimation tool *HERest* as indicated in Example 4.3:

(4.3.) HERest –A –D –T 1 –V –S audio_trn.lst –t 250.0 150.0 1000.0 –H hmm_0/macros –H hmm_0/hmmDefs.mmf –M hmm_1 –s stats monophones.lst

This serves to load all the models contained in the *hmmDefs.mmf* file which are listed in the model list (*monophones.lst*), excluding the short pause (*sp*) model. The loaded models are then re-estimated using the training data listed in *audio_trn.lst* to create a new model set stored in the directory *hmm_1*. The re-estimation was performed with three iterations until the final sets of initialised HMMs were stored in the third HMM directory (*hmm_3*).

The next step was to create the short pause (*sp*) model, which was excluded in the preceding steps. The model was stored in the fourth HMM directory (*hmm_4*). The emitting state of the *sp* model was then tied to the centre state of the silence (*sil*) model. This was achieved by invoking the *HHEd* tool in Example 4.4.

(4.4.) HHEd –T 1 –H hmm_4/macros –H hmm_4/hmmDefs.mmf –M hmm_5 sil.hed monophones_sp.lst

This extended the initial monophone list (*monophones.lst*) with the new *sp* model and stores them in *monophones_sp.lst*. Re-estimation was performed twice, this time including the *sp* model. The latest models are used to realign and select best pronunciations for both the training and testing data. Re-estimation was then performed twice on the latest models with the aligned data.

The succeeding stage of the model generation procedure was to use the monophone HMMs to create context-dependent triphone HMMs. To achieve this, we first had to convert the monophone transcriptions to triphone transcriptions and then create a set of triphone models by cloning the monophones and then re-estimating them using the triphone transcriptions. Secondly, similar acoustic states of these triphones were tied to ensure that all state distributions can be robustly estimated (Young *et al.*, 2006).

The *HLEd* tool was invoked to convert the aligned monophone transcriptions to their equivalent triphone transcriptions. The generated triphones must have at least one

example in the training data. For example, the monophones in Example 4.5 will become the triphones in Example 4.6.

(4.5.) sil B O u t_> O sp j a sp B O n tS_> I sp BZ a sp m a l O k_> O sp sil

(4.6.) sil sil-B+O B-O+u O-u+t_> u-t_>+O t_>-O+j sp O-j+a j-a+B sp a-B+O B-O+n O-n+tS_> n-tS_>+I tS_>-i+BZ sp i-BZ+a BZ-a+m sp a-m+a m-a+l a-l+O l-O+k_> O-k_>+O k_>-O+sil sp sil

Conversely, for the grapheme-based system, the monographemes in Example 4.7 became the trigraphemes in Example 4.8.

(4.7.) sil b o u t o sp y a sp b o n t S I sp b j a sp m a l o k o sp sil

(4.8.) sil sil-b+o b-o+u o-u+t u-t+o t-o+y sp o-y+a y-a+b sp a-b+o b-o+n o-n+t n-t+S t-S+I S-i+b sp i-b+j b-j+a j-a+m sp a-m+a m-a+l a-l+o l-o+k o-k+o k-o+sil sp sil

The context-dependent HMMs were cloned using HHEd and the *mktri.hed* script which allows the tying of all the transition matrices in each triphone set. HERest was the used to re-estimate the new triphone sets.

To this point, we had a set of triphone HMMs with all triphones sharing the same transition matrix per phone set. Each HMM state distribution was modelled by shared 16-Gaussian mixtures with a diagonal covariance matrix. The final stage involved tying the states within triphone sets in order to share data and thus be able to make robust parameter estimates (Young *et al.*, 2006). This was done by using decision trees, mentioned in the preceding section, to cluster the states then tie the clusters. HHEd was invoked with the script *tree.hed* to perform decision tree state-tying, as shown in Example 4.9.

(4.9.) HHEd –A –D –T 1 –V –H hmm_12/macros –H hmm_12/hmmDefs.mmf –M hmm_13 trees.hed triphones.lst

Upon completion of state-tying, some of the new models were identical, i.e., they pointed to the same 3 tied-states and transition matrices. Identical models were tied together, this compacted the models to produce a new model set called *tiedlist*. What was left at this stage was to increase the mixtures by cloning the new models and re-

estimating them (Appendix F). The final step was to apply semi-tied transforms and then re-estimate the models further to improve the robustness of the acoustic models.

4.2.5. Language Modelling

The SRILM language modelling toolkit (Stolcke, 2002) was used to train word-level Language Models (LMs) from the sentential transcriptions. SRILM allows two major language modelling operations, estimation and evaluation. Language model estimation refers to the creation of a model from a set of training data, and evaluation refers to the calculation of the probability of the test data, commonly expressed as the test set perplexity (Stolcke, 2002). For each system, a statistical *n*-gram LM was trained and employed in the decoding process. The use of well-trained statistical *n*-gram LMs can attain better speech recognition accuracies (Besling, 1994; Kanthak and Ney, 2003).

To build an LM training corpus, words were extracted from all the sentential transcriptions in the training data set. The generated training corpus was used to train a third order (3-gram) LM for each language in the two corpora. The *ngram-count* tool was used to estimate the word probabilities from the training corpus, as shown in Example 4.10.

(4.10.) ngram-count –text corpus.train –order 3 –lm trigram.lm –interpolate –cdiscount1 0.7 –cdiscount2 0.7 –cdiscount3 0.7

The above-stated tool trained a 3-gram LM *trigram.lm* from the training corpus *corpus.train* using interpolated absolute discounting with a discounting coefficient of 0.7. The LM order, like the discounting coefficient, can be specified arbitrarily by the user. A portion of the LM file is shown in Figure 4.3.

We further build an LM testing corpus by extracting all words from the sentential transcriptions of the testing data sets. The tool *ngram* was invoked with the option *–ppl* to evaluate the trained LM on the test corpus *corpus.test* to compute the test corpus perplexity, as shown in Example 4.11.

(4.11.) ngram –ppl corpus.test –order 3 –lm trigram.lm

Since both the approaches use the same LMs, only one third-order (3-gram) LM was trained and evaluated for each language corpus. Table 4.6, outlines the details of the LMs from the Lwazi ASR corpus and Table 4.7 shows that of the NCHLT ASR corpus.

```
\data\
ngram 1=11086
ngram 2=31807
ngram 3=37768

\1-grams:
-0.7763174  </s>
-99 <s> -1.436939
-1.262546   a   -1.339154
-4.623612   aba -0.7569619
-5.049581   abagana -0.6320232
....
\2-grams:
-1.16631    <s> a   -1.023438
-3.901014   a batalala  -0.6320232
-0.7115493  baswa </s>
....
\3-grams:
-1.699049   <s> a
-3.440158   <s> a ahlolwe
-0.2902525  bile a </s>
...

\end\
```

Figure 4.3: A typical 3-gram LM generated by SRILM

These tables below indicate the total number of words in the LMs, the number of out-of-vocabulary (OOV) words and the test set perplexity. The LM perplexity is used to evaluate the accuracy of the language model. The best language model is the one that best predicts unseen words (OOV) in the test set.

TABLE 4.6: DETAILS OF THE LMS FOR EACH LANGUAGE FROM THE LWAZI CORPORA

	SEPEDI	ISINDEBELE	TSHIVENDA	MULTI-LING
Total Words	45206	37742	36266	111426
# Trigrams	5709	5458	4975	14073
OOVs	291	401	162	816
Test Set Perplexity	9.98132	16.695	13.61	10.2929

The LMs of the Lwazi corpora have a lower perplexity compared to those of the NCHLT corpora. This is due to the significant difference in the amount of training and testing data between the two corpora.

Table 4.7: DETAILS OF THE LMS FOR EACH LANGUAGE FROM THE NCHLT CORPORA

	SEPEDI	ISINDEBELE	TSHIVENDA
Total Words	279997	140871	793589
# Trigrams	37768	30081	26409
OOVs	158	367	648
Test Set Perplexity	12.4	23.958	48.9825

4.2.6. Pattern Classification (Decoding)

Having successfully trained robust context-dependent acoustic models, the next step was to evaluate the recognition performance using the test set. This was achieved by using the Viterbi decoding algorithm. The algorithm uses a list of physical models (tiedlist), the recognition network (grammar), and the pronunciation dictionary to recognise (transcribe) a set of audio files (the test set). The values of the insertion penalty, grammar scale factor and beam-width pruning threshold were optimally set for decoding.

A typical HTK recogniser uses the *Hvite* tool with optimal parameter values and a flat start language model to perform recognition of a test set. However, *Hvite* does not allow high order *n*-gram language models; therefore, the *HDecode* tool was used instead. *HDecode* is an HTK-patch designed for LVCSR tasks. It can handle larger *n*-gram language models, restricted to up to the third-order (Young *et al.*, 2006). The LMs described in the previous section were used for decoding the test sets in the respective experiments. The *HDecode* tool will dump recognition results into a file which can be used later to evaluate the overall system performance.

4.3. Summary

In this chapter, we discussed most important steps of the overall study in details. The speech corpora used for training and evaluation in all experiments was discussed. The pronunciation dictionaries used were also discussed. The procedures for developing the grapheme-based dictionaries and generating decision trees for context-dependent tied-state acoustic models were outlined. Some of the key HTK commands executed at the key experimental steps were also briefly outlined. The following chapter discusses the

results obtained using the framework outlined. The recognition results answers the framed research questions and also answers whether or not the research approach is found to be plausible.

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1. Introduction

The aim of this study is to compare the performance of two acoustic modelling units; graphemes and phonemes. Two speech corpora were employed, the Lwazi ASR corpus and the NCHLT ASR corpus. For each corpus, context-dependent tied-state acoustic models were trained using both units. Two types of pronunciation dictionaries were used for decoding, namely, the grapheme-based and the phoneme-based dictionary. The form of the dictionaries and model generation procedures were discussed in the previous chapter.

This chapter discusses the decoding parameters, the procedure followed to generate the recognition results, presents the speech recognition statistics and errors with a brief analysis thereof. From each of the corpus, the three typically under-resourced languages were selected for ASR experiments. For the Lwazi ASR corpus, six monolingual and two multilingual ASR experiments were conducted, resulting to eight experiments shared equally for grapheme- and phoneme-based units. Only monolingual ASR experiments were conducted on the NCHLT corpus. This is due to the scope and feasibility of the study. The NCHLT corpus was not only used to increase training data and improve results, but also to cross-validate the results obtained with the Lwazi ASR corpus by means of reproducing them.

5.2. ASR Performance Evaluation Metrics

The WER is the most commonly used metric to evaluate the overall recognition performance of ASR systems. To compute the WER, the recognition output (rec file) is compared with the reference (label) file (i.e. the corresponding correct transcriptions). The three typical types of word recognition errors in ASR are (Huang *et al.*, 2001):

- Substitution (S): an incorrect word was substituted for the correct word.
- Deletion (D): a correct word was omitted in the recognized sentence.

- Insertion (I): an extra word was added in the recognized sentence.

The WER is defined as:

$$WER = \frac{S+D+I}{N} * 100\% \quad (5.1)$$

where *N* is total the number of words in the correct sentence.

In some cases the recognition performance of an ASR system can be measured by the phone recognition accuracy, using a metric termed phone error rate (PER). The PER is measured exactly the same way as WER, except individual words are replaced with individual phones (Mabokela, 2014).

Furthermore, the performance of a speech recogniser can be measured according to the accuracy and correctness of word recognition. The word accuracy measures how an ASR system accurately captures the spoken signal as a word, it is defined as follows:

$$Word\ Accuracy = \frac{N-S+D+I}{N} * 100\% \quad (5.2)$$

The word correctness on the other hand, measures the correctness of every recognised word, and it is defined as follows:

$$Word\ Correctness\ = \frac{N-D-S}{N} * 100\% \quad (5.3)$$

5.3. ASR Systems Evaluation with HDecode

The *HVite* decoder is only suitable for systems using bigram language models. As stated previously, we used trigram language models in all experiments. We therefore used the *HDecode* tool to decode/recognise the test (evaluation) sets in all the experiments. The *HDecode* decoder has a number of predefined restrictions, one of them being that it supports *n*-gram LMs up to trigrams (Young *et al.*, 2006).

5.3.1. Optimum Decoding Parameters

HDecode requires four very important parameters to perform decoding; (i) the model definitions – contained in a master macro file with extension .mmf, (ii) a statistical language model – the trigram language models described in the previous sections, (iii) the pronunciation dictionary and, (iv) a list of physical models – the tied-state triphones (trigraphemes) in tiedlist (Young *et al.*, 2006). Of course, a set of acoustic features to be recognised must also be passed as a parameter – this is the test set from which the recognizer must be evaluated. To obtain reliable recognition test results, the test set data must not appear in the training set.

Furthermore, HDecode requires fixed parameter values to control the searching process. The values chosen in our experiments are those that yielded optimum recognition accuracies. The LM scaling factor (-s) was set to -10, the pruning threshold (-t) was set to 240, and the word insertion penalty (-p) was set to -25. These values were chosen by carefully running experiments on the development set to test the optimal value. We first used the default values and iterated the experiments with different values until optimal recognition results were obtained.

The word insertion penalty is a fixed value that is added to the accumulated log likelihood each time a new word is entered during the Viterbi search. It is used to balance the relation between the deletion and insertion errors. The default value of the HTK is 0.0 (Young *et al*., 2006), but the effect of this parameter on the accuracy may be different per language. Therefore, calibrating this value for each experiment is important. As a result, we ran multiple experiments for each language to determine the optimum value. The value was set from the default 0.0 in the range as follows, -5.0, -10.0,…,-30.0. Interestingly, the optimum value for both trigrapheme and triphone experiments was -25 in the three languages. In all the experiments, the performance of the systems began to degrade below -25.0 and above it. Hence this value was selected.

The pronunciation dictionary contains a list of words and their correct pronunciation. It also contains the sentence start and the sentence end tokens, and models them with a silence phoneme/grapheme. *HDecode* does not permit the silence model, *sil*, and the short pauses, *sp*, to appear in the pronunciation dictionary. The silence model *sil* appear as the dictionary entry (or the pronunciation) of both the *sentence start* and the *sentence end* tokens.

HDecode will then store the recognition output in a file. The file contains the estimated transcription of each input file. Each transcription is given as a set of hypotheses – the closest estimates each word in the correct transcription of the input acoustic feature. A typical output recognition file is indicated in Figure 5.1.

The *sentence start* token – the first hypothesis of every input signal, is recognised as *sil*. The actual words are then recognised individually as a single hypothesis. Finaly, the *sentence end* token, also recognised as *sil*, as the last hypothesis of every input. The start point of each hypothesis is given in the first column, the end point in the second, and the estimated acoustic scores in the fourth (last) column. The acoustic score is estimated by the embedded *Viterbi* decoding algorithm

```
File: sepedi_152_02.mfc
CPU time 2.260000  utterance length 2.240000  RT factor 1.008929
Transcription: 1-best hypothesis [1 lists]
 List 1
   1.        0 1000000      <s> -86.840378
   2.  1000000 5200000        e -71.488487
   3.  5200000010900000   masome -78.841263
   4. 10900000014100000     pedi -81.864105
   5. 14100000020500000      nne -77.527817
   6. 20500000022400000     </s> -78.809158
Stats: nTokSet 1376815
Stats: TokPerSet 14.180460
Stats: activePerFrame 3019.540179
Stats: activateNodePerFrame 1550.424107
Stats: deActivateNodePerFrame 1525.968750
```

Figure 5.1: A typical *HDecode* output for an input feature file

5.3.2. Generating Recognition Results

The HTK provides a performance evaluation tool, *HResults*, which computes performance statistics. The *HResults* tool also takes several parameters. We summarise only those which were of the immediate interest of the study. With its invocation, three important files are passed to *HResults*, the reference master label file (MLF), the recognised MLF, and a list of tied-state triphones/trigraphemes.

The reference MLF contains the correct transcriptions of the entire test data set (the lab files), and the recognised MLF contains the recognised transcriptions of the entire test data set (the rec files) as generated by *HDecode*. *HResults* measures the recognition performance by performing optimal string matches, i.e. it compares the reference transcriptions to the recognition hypothesis per input file, as shown in Table 5.1 below.

TABLE 5.1: A TYPICAL HRESULTS OUTPUT OF COMPARING A REC FILE TO A LAB FILE

```
Aligned transcription: sepedi_180_03.lab vs sepedi_180_03.rec
 LAB:     ke monna
 REC: <s> ke monna </s>
```

The recognition statistics are then dumped on the screen or redirected to a file (like we did in this study). It is from the statistics that individual recognition errors can be analysed and recognition error rates can be calculated.

5.4. Baseline Recognition Results of the Lwazi Evaluation Set

For the Lwazi ASR corpora, three monolingual ASR systems were first trained and evaluated independently for the two approaches. A multilingual system was then trained with the three selected languages. We report the results of all the systems and analyse them.

5.4.1. Evaluating the Monolingual Lwazi ASR Systems

For all three monolingual ASR systems, we present a single figure outlining the evaluation results of the two experiments. The phoneme-based experiment (ExpPho) is on the left side of the figure and grapheme-based experiment (ExpGra) on the right side. The results for the languages, IsiNdebele, Sepedi, and Tshivenda are presented in Tables 5.2, 5.3 and 5.4, respectively. The results presented here were directly generated by the *HResults* tool.

The phoneme-based WERs obtained in this study are comparable to those reported by Henselmans *et al.,* (2013). The slight discrepancies can most likely be attributed to the kind of language models used and also the partitioning of the training and evaluation sets. As one would expect, there is a very strong correlation between the LM perplexities of each language and the recognition accuracies.

TABLE 5.2: THE LWAZI ASR RECOGNITION STATISTICS OF THE PHONEME-BASED EXPERIMENT (EXPPHO) VS. THE GRAPHEME-BASED EXPERIMENT (EXPGRA) FOR ISINDEBELE LANGUAGE

```
============ HTK Results Analysis ===========
  Date: Tue Jul 22 13:34:11 2014
  Ref : pho/mlfs/words.tst.mlf
  Rec : results.final.mlf
------------ Overall Results ---------------
SENT: %Correct=0.00 [H=0, S=1320, N=1320]
WORD: %Corr=65.21, Acc=34.77 [H=6714, D=739,
                    S=2843, I=3134, N=10296]
===============================================
```

```
============ HTK Results Analysis ===========
  Date: Tue Jul 22 13:34:21 2014
  Ref : gra/mlfs/words.tst.mlf
  Rec : results.final.mlf
------------ Overall Results ---------------
SENT: %Correct=0.00 [H=0, S=1320, N=1320]
WORD: %Corr=64.01, Acc=34.94 [H=6590, D=790,
                    S=2916, I=2993, N=10296]
===============================================
```

The word LM perplexity of IsiNdebele is the highest of all the three languages and hence it is not surprising that the word recognition accuracy is also the worst of all the languages. That is, the word accuracy is 34.77% for phonemes and 34.94% for graphemes in IsiNdebele as compared to 46.40% and 45.68% for Sepedi and 38.20% and 40.56% for Tshivenda. The IsiNdebele word LM also had the highest OOVs, which is one of the factors that influenced the low accuracies. This is because the OOV rate has a significant impact on the recognition rate.

TABLE 5.3: THE LWAZI ASR RECOGNITION STATISTICS OF EXPPHO VS. EXPGRA FOR SEPEDI LANGUAGE

```
============ HTK Results Analysis =============
  Date: Thu Jul 17 15:02:06 2014
  Ref : pho/mlfs/words.tst.mlf
  Rec : pho/hdecode/results/results.final.mlf
------------------------ Overall Results ------
SENT: %Correct=0.00 [H=0, S=1160, N=1160]
WORD: %Corr=66.03, Acc=46.40 [H=7677, D=805,
                              S=3145, I=2282, N=11627]
================================================
```

```
============ HTK Results Analysis ============
  Date: Thu Jul 17 15:03:14 2014
  Ref : gra/hdecode/final/words.tst.mlf
  Rec : gra/hdecode/final/results.final.mlf
------------------------ Overall Results -----
SENT: %Correct=0.00 [H=0, S=1160, N=1160]
WORD: %Corr=65.89, Acc=45.68 [H=7661, D=725,
                              S=3241, I=2350, N=11627]
================================================
```

Sepedi has a better word recognition accuracy compared to all the three languages, 46.40% for phonemes and 45.68% for graphemes. The word accuracies also correspond very well with word LM perplexity. Since the Sepedi LM had the lowest perplexity and OOV rate, the results are expected to be the highest. This is because the lower the OOVs in the test set the higher the LM estimation of the unseen words.

TABLE 5.4: THE LWAZI ASR RECOGNITION STATISTICS OF EXPPHO VS. EXPGRA FOR TSHIVENDA LANGUAGE

```
============ HTK Results Analysis ===========
  Date: Thu Jul 17 18:07:44 2014
  Ref : pho/mlfs/words.tst.mlf
  Rec : results.pho.mlf
------------- Overall Results ---------------
SENT: %Correct=0.00 [H=0, S=1225, N=1225]
WORD: %Corr=61.16, Acc=38.20 [H=5797, D=933,
                              S=2748, I=2176, N=9478]
================================================
```

```
============ HTK Results Analysis ===========
  Date: Thu Jul 17 15:11:05 2014
  Ref : gra/mlfs/words.tst.mlf
  Rec : results.gra.mlf
------------- Overall Results ---------------
SENT: %Correct=0.00 [H=0, S=1225, N=1225]
WORD: %Corr=64.33, Acc=40.56 [H=6097, D=867,
                              S=2514, I=2253, N=9478]
================================================
```

As noted in Table 4.6, the Tshivenda word LM has the lowest OOV rate (only 162) due to its small vocabulary. However, the word LM perplexity is still significantly higher and hence it is reflected in the word recognition accuracy which is 38.2% and 40.56% for phonemes and graphemes, respectively. The results for all languages are consistent in the two approaches since they both use the same LMs.

5.4.2. Analysis of the Lwazi Monolingual ASR Systems

Having tested both experimental approaches on each language, we obtained the following WERs: 54.32% WER on graphemes and 53.59% on phonemes for Sepedi, 59.44% on graphemes and 61.79% on phonemes for Tshivenda, 65.06% on graphemes

and 65.22% on phonemes for IsiNdebele and 64.59% on graphemes. The WERs are graphically presented in Figure 5.2.

The performance of the two approaches is language-dependent. As outlined in Figure 5.2, graphemes outperformed phonemes with a significant margin for Tshivenda. The grapheme-based sub-word units obtained a WER reduction of above 2.35%, which is indeed significant. For the IsiNdebele language, graphemes also outperformed phonemes, but with a very small margin. The grapheme-based sub-word units reduced the WER with 0.16%. However, for Sepedi, phonemes demonstrate superiority over graphemes.

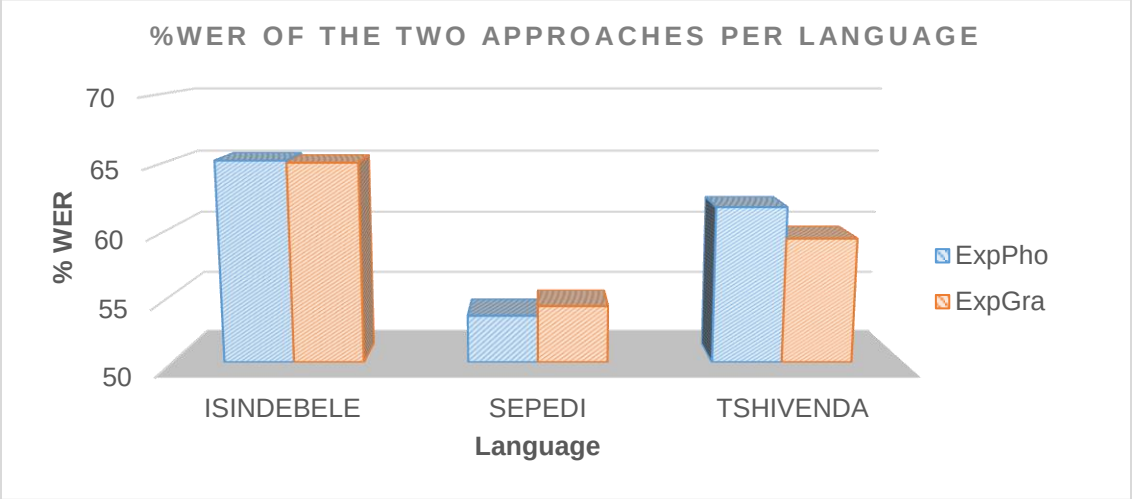

Figure 5.2: Percentage WERs obtained in ExpPho and ExpGra for each of the three languages

The phoneme-based sub-word units are 0.73% more accurate than the grapheme-based units. This is a considerably small margin and thus also demonstrates that indeed graphemes can attain comparable recognition performance for this language.

5.4.3. Evaluating the Multilingual Lwazi ASR System

For the multilingual system, a single figure is also presented, outlining the evaluation results of the two experiments. The results are presented in Table 5.5. ExpPho is also on the left side of the figure and ExpGra on the right side.

Table 5.5 indicates the multilingual system suffers significantly larger recognition errors, with a word recognition accuracy of 37.77% for phonemes and 35.41% for graphemes. This is because the evaluation set (test data) is much bigger since it's a combination of all three evaluation sets from the three languages. The recognition vocabulary is also broad hence the word LM has a higher perplexity and thus the overall recognition performance is expected to degrade.

TABLE 5.5: THE LWAZI RECOGNITION STATISTICS OF EXPPHO VS. EXPGRA FOR THE MULTILINGUAL ASR SYSTEM

```
============ HTK Results Analysis ==============
  Date: Thu Jul 24 15:17:34 2014
  Ref : pho/mlfs/words.tst.mlf
  Rec : results.final.mlf
------------- Overall Results ------------------
SENT: %Correct=0.00 [H=0, S=3706, N=3706]
WORD: %Corr=64.36, Acc=37.77 [H=18739, D=1423,
                    S=8955, I=7741, N=29117]
================================================
```

```
============ HTK Results Analysis ==============
  Date: Thu Jul 24 15:18:33 2014
  Ref : gra/mlfs/words.tst.mlf
  Rec : results.final.mlf
------------- Overall Results ------------------
SENT: %Correct=0.00 [H=0, S=3706, N=3706]
WORD: %Corr=60.97, Acc=35.41 [H=17753, D=1550,
                    S=9814, I=7443, N=29117]
================================================
```

It is observed from the results obtained by the multilingual system, presented in Table 5.5, that a combination of a large recognition vocabulary and a high LM perplexity constitutes low recognition accuracies.

The WERs obtained by the two approaches are graphically presented in Figure 5.4. A WER of 64.59% was attained in the grapheme-based experiment while 62.22% was obtained in the phoneme-based experiment. The multilingual platform, just like the monolingual Sepedi ASR, also sees phoneme-based acoustic sub-word units performing better with a WER reduction of 2.37%.

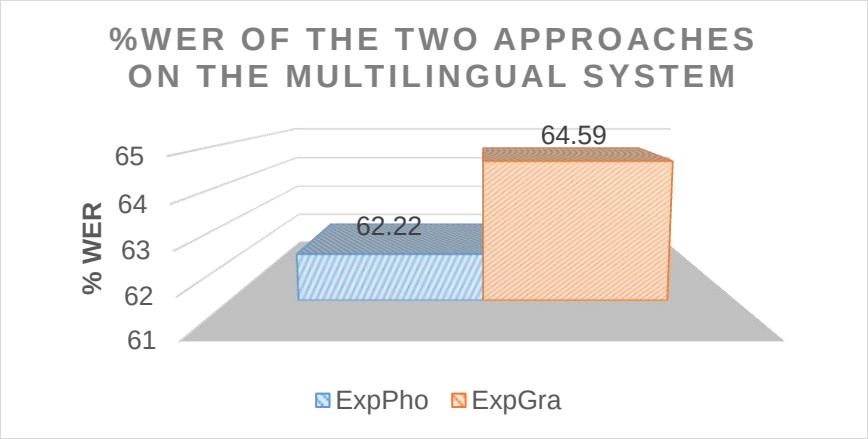

Figure 5.3: Percentage WERs obtained in ExpPho and ExpGra for the multilingual ASR system

Interestingly, the overall cross-lingual acoustic models perform worse than the monolingual models regardless of the data shared across languages. Moreover, for unknown reasons, the grapheme-based models are worse than the phoneme-based regardless of having more graphemes shared across the languages than phonemes. To investigate these interesting observations, each language was tested on the cross-lingual acoustic models. The results are outlined in Figure 5.5.

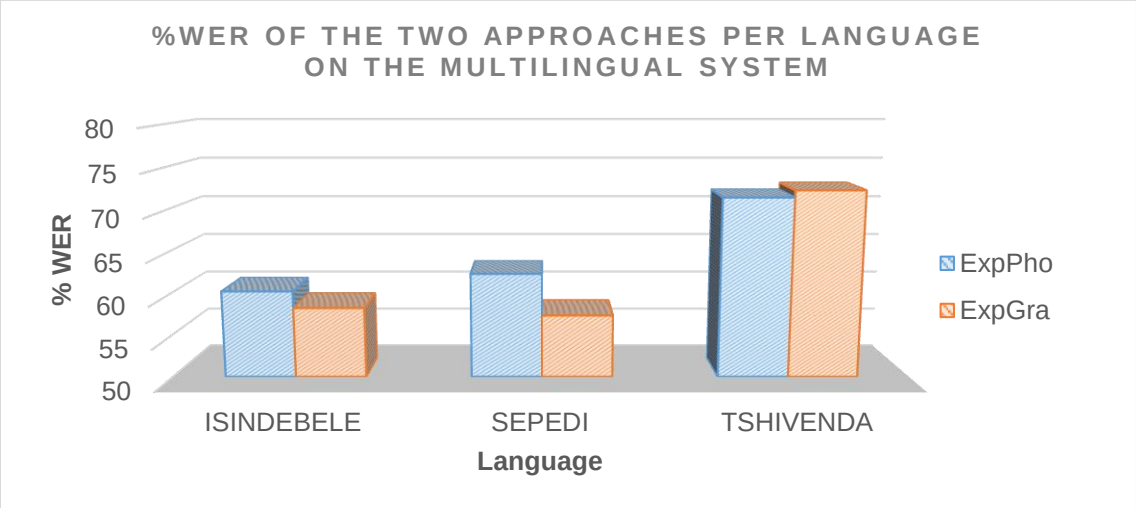

Figure 5.4: Percentage WERs obtained in ExpPho and ExpGra for each language on the multilingual ASR system

As reflected in Figure 5.5, the monolingual acoustic models perform better than the cross-lingual models. The phoneme-based units attain a WER of 60.41% for IsiNdebele, 62.54% for Sepedi and 71.6 for Tshivenda. However, graphemes perform better than

phonemes on the cross-lingual models for IsiNdebele and Sepedi, obtaining a WER of 58.42 and 57.51%, respectively. This implies that there were more graphemes shared across Sepedi and IsiNdebele than Tshivenda and any of the languages. The shared graphemes increase the model training data and since there are more graphemes shared across the language than there are phonemes, the grapheme-based units are expected to perform better. These findings are also supported by those in Manaileng and Manamela (2014). As stated by Manaileng and Manamela (2014), Tshivenda has five graphemes which are unique to the combined grapheme set of the three languages. This means that the shared training data does not account for 15% of the graphemes during model training. Inadequately trained models in a multilingual acoustic model platform can increase recognition errors due to model mismatch (Ulla, 2001).

Although cross-lingual data sharing provides the fundamental advantage of combining training data of multiple languages, sharing data across phonetically unrelated languages can be a disservice for other languages. One of the important observations drawn from the results is that for graphemes to perform better in cross-lingual data sharing, the languages must have common graphemes and they must have a small number of unique graphemes. Otherwise, there may be little data to train language-unique models which would then result in the contamination of the model set. It is therefore evident that Tshivenda is not suitable for sharing data with Sepedi and IsiNdebele, despite the languages' close socio-geographical proximity.

5.5. Recognition Results of the NCHLT Evaluation Set

Unlike with the Lwazi ASR corpora – in which both monolingual and cross-lingual acoustic models were trained, only monolingual acoustic models were trained for the NCHLT corpora. This is due to the scope of the project. We present and analyse recognition results obtained from the two approaches for each language.

5.5.1. Recognition Statistics of each Language

The recognition results are analysed using two recognition metrics in addition to the common metrics WER, namely, word accuracy and word correctness. As previously mentioned, word accuracy measures how an ASR system accurately captures the spoken signal as a word while the word correctness measures the correctness of every recognised word. The word recognition statistics per experiment for each language are outlined in Table 5.6.

TABLE 5.6: PERCENTAGE WORD ACCURACY AND WORD CORRECTNESS OBTAINED IN EXPPHO AND EXPGRA FOR EACH LANGUAGE

Language	Recognition Metric (%)	ExpPho	ExpGra
ISINDEBELE	Word Accuracy	70.58	71.11
	Word Correctness	74.42	75.05
SEPEDI	Word Accuracy	68.75	75.62
	Word Correctness	73.06	79.27
TSHIVENDA	Word Accuracy	70.42	74.70
	Word Correctness	73.70	70.38

As indicated in Table 5.6, the grapheme-based units attain better word recognition accuracy than the phoneme-based ones for all languages. For two languages, IsiNdebele and Tshivenda, the accuracies attained by the two approaches differ in a small margin. For Sepedi however, a slightly larger difference in word recognition accuracies was obtained by the two approaches. Graphemes also performed better in word recognition correctness than phonemes for IsiNdebele.

However, the performance of graphemes degrades in word recognition performance for Tshivenda. It is not obvious what the causes of the degradation are, but the LM perplexity and the little training data can be attributed to this phenomenon. As noted in Table 4.3, Tshivenda has the lowest amount of training data, 33.1 hours compared to the 46.3 hours for Sepedi and 46.5 hours for IsiNdebele. Moreover, Tshivenda has the highest number of graphemes, outlined in Table 4.5. The number of graphemes is very close to that of phonemes, 32 graphemes and 39 phonemes, unlike for the other languages. This means that the amount of speech data available for training each grapheme model nearly equals the amount available to train each phoneme model.

Furthermore, Tshivenda has the highest LM model perplexity, as shown in Table 4.7. These factors collectively contribute to the inferior and odd performance by graphemes for this language.

Table 5.7 presents the WERs obtained in the two experiments for each language. The difference – the right-most column of the table, is used to measure the superiority of one approach over another. The WERs clearly correlates with the word accuracies and word correctness in the previous table.

The WERs are comparable to those obtained in a study by Barnard *et al.* (2014). However, our results cannot be homologous to theirs due to the difference in recognition framework and the employed language models. Furthermore, Barnard *et al.* (2014) used the *Kaldi speech recognition toolkit* for decoding whereas HTK was used in this study.

As noted in Table 5.7, the grapheme-based units perform slightly better than the phoneme-based ones for IsiNdebele by attaining a WER reduction of 0.54%. For Sepedi, graphemes performed better by attaining a significantly higher WER reduction of 6.91%. For Tshivenda however, graphemes perform slightly inferior with phonemes being 0.04% more accurate than graphemes. A very similar study by Basson and Davel (2013) also reported degradation in word recognition accuracy using graphemes for the Afrikaans language. Although the grapheme-based system performed worse than the phoneme-based system, the results are still comparable and the authors successfully identified a set of “problematic categories” as the causes of the under par performance of the grapheme-based acoustic sub-word units.

TABLE 5.7: WERs OBTAINED BY THE TWO APPROACHES AND THEIR DIFFERENCE FOR EACH LANGUAGE

Language	% WER		Difference
	ExpPho	ExpGra	
ISINDEBELE	29.42	28.88	0.54
SEPEDI	31.25	24.34	6.91
TSHIVENDA	29.57	29.61	-0.04

Schukat-Talamazzini *et al.,* (1993) achieved better recognition results with graphemes, obtaining a 1.68% better word-level recognition accuracy. Sirum and Sanches (2010), who studied the effect of WER for Portuguese language when the acoustic units based on phonemes and graphemes are compared, also reported that there is no considerable difference in performance between the phoneme-based speech recognizer and the grapheme-based one when evaluated over Command & Control and Connected digit ASR experiments.

What seems to be interesting however, is that context-dependent grapheme-based sub-word units perform better than the phonemic ones in our study as opposed to the observations made in the study by Kanthak and Ney (2002). The most likely factor may be the phonetic structure of the languages of focus. One other possible factor might be that the quality some of the pronunciation dictionaries is not optimal. Sirum and Sanches (2010) also reported that their grapheme-based speech recognizer performed considerably worse than the phoneme-based over a Spelling ASR experiment.

5.5.2. Number of GMMs vs. the Recognition Performance for Each Experiment

One of the important factors that contribute to recognition accuracy is the number of GMMs per state during model training. We therefore investigated the effect the number of GMMs has on the ultimate WER for each language. We analysed the behaviour of the WER in both approaches when the number of GMMs is altered. The results are presented in Figures 5.10, 5.11, and 5.12 for IsiNdebele, Sepedi and Tshivenda, respectively.

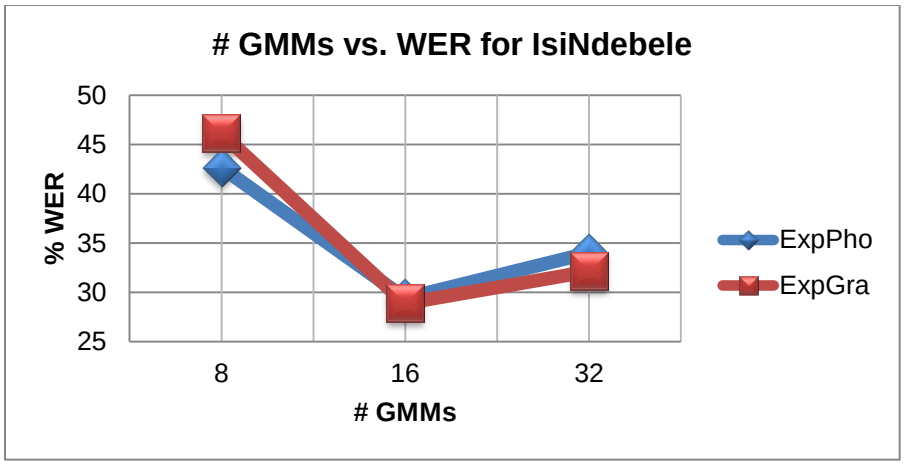

Figure 5.5: Effect of the number of GMMs on WER for IsiNdebele

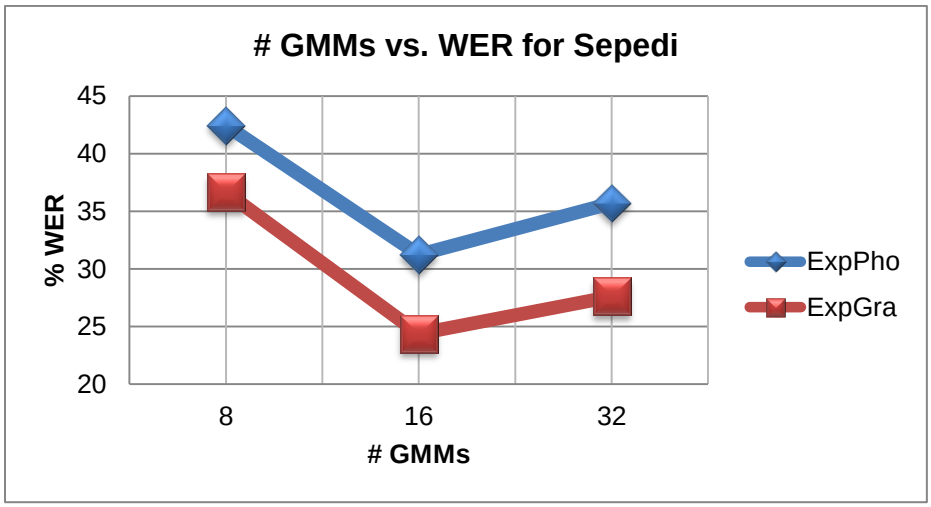

Figure 5.6: Effect of the number of GMMs on WER for Sepedi

It is evident from the diagrams that there exist a strong relationship between the number of GMMs and the recognition performance (WER). Interestingly, the two approaches behave similarly when the number of GMMs is increased. For a number of GMMs, the WER is either increased or decreased in both approaches.

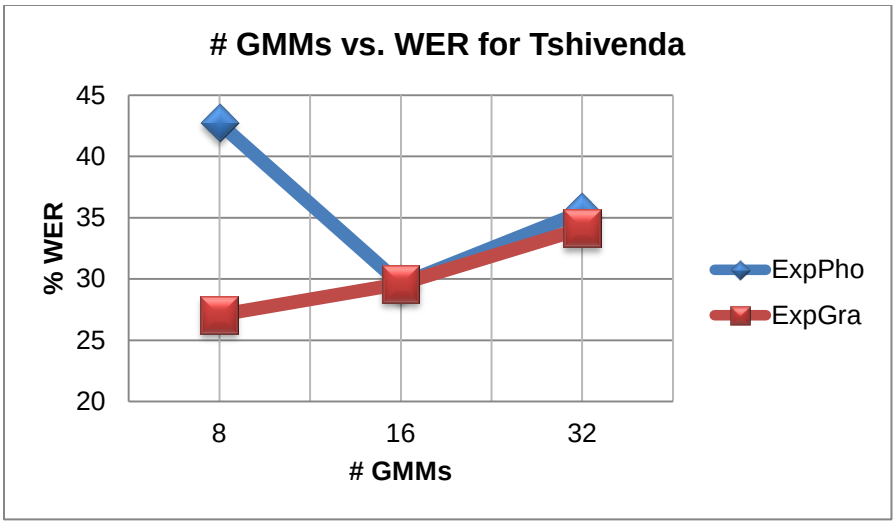

Figure 5.7: Effect of the number of GMMs on WER for Tshivenda

A special case is only in Tshivenda where WER is lowest for the grapheme-based experiment with 8 GMMs whereas it is the highest for the phoneme-based with the same number of GMMs. This can be attributed to the same factors that constituted an odd trend of the WER, as discussed in the previous section. However, we observed that the optimum recognition results are attained with 16 GMMs in the overall systems.

5.5.3. Error Analysis

Since the two approaches attain varying WERs, although the difference is largely small, it is interesting to see, and important to know, which recognition errors each approach suffers. The language structure can be a significant determinant of how a recogniser handles errors during recognition; therefore the investigation must be done for each language. To carry out the investigation, individual recognition errors were analysed for each approach in all the three languages. Figures 5.9, 5.10, and 5.11 highlight the number of individual errors for each ASR experiment in IsiNdebele, Sepedi and Tshivenda, respectively.

As presented in Figure 5.9, the two approaches suffer marginal recognition errors for IsiNdebele. The grapheme-based units suffer slightly lower substitution errors and hence the ultimate recognition performance is slightly better.

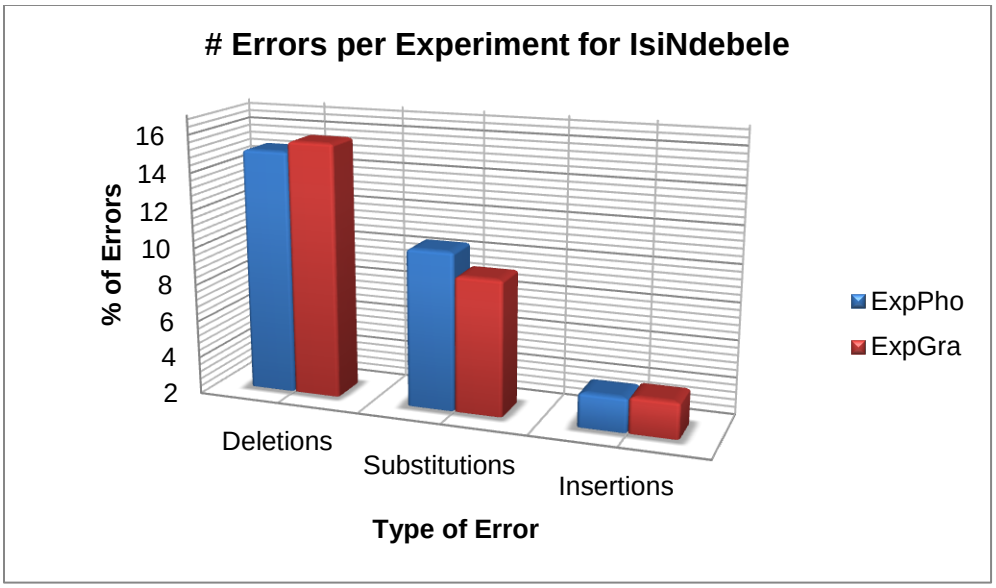

Figure 5.8: Number of recognition errors for each experiment in IsiNdebele

For Sepedi, however, the phoneme-based units uniquely suffer significant substitution errors. The grapheme-based units significantly reduce the number of substitution errors, as outlined in Figure 5.10. This reduction correlates very well with the overall WER reduction of 6.91% to make the grapheme-based units significantly superior to their phoneme-based counterparts.

Tshivenda has a similar trend (Figure 5.11) with IsiNdebele in the sense that both units almost handle the errors the same way. Graphemes handle both deletion and substitution errors slightly better and the insertion errors almost the same for Tshivenda. However, there is a spike in the substitution errors suffered by the phoneme-based units for Sepedi.

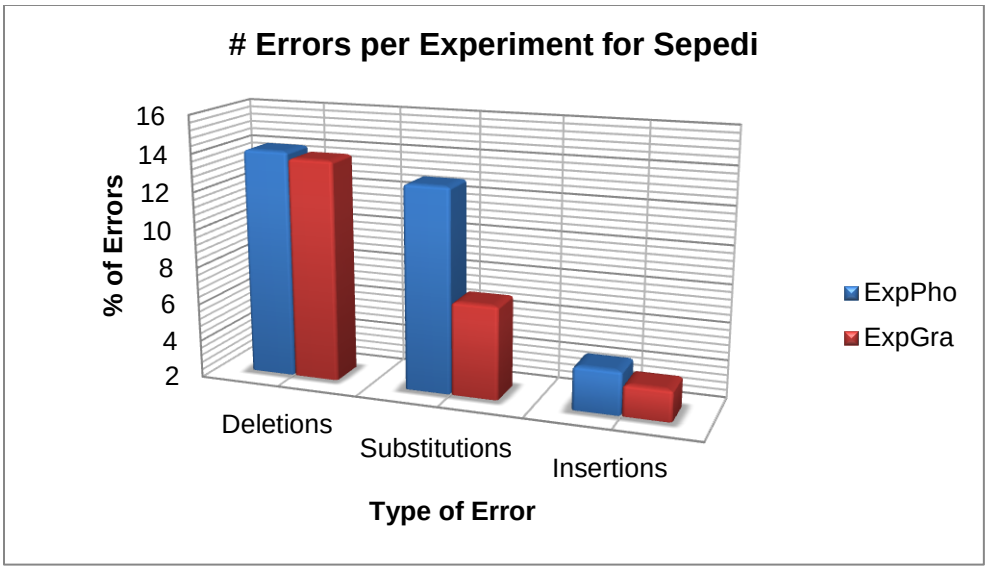

Figure 5.9: Number of recognition errors for each experiment in Sepedi

These discrepancies may be due to the phonetic structure of the languages and/or the pronunciation modelling accuracy. Substitution errors are caused by a phoneme/grapheme being confused with another and thus being wrongfully substituted thereof. Languages having a number of phonemes that sound alike are susceptible to substitution errors since an accurate pronunciation modelling of closely similar phonemes is difficult.

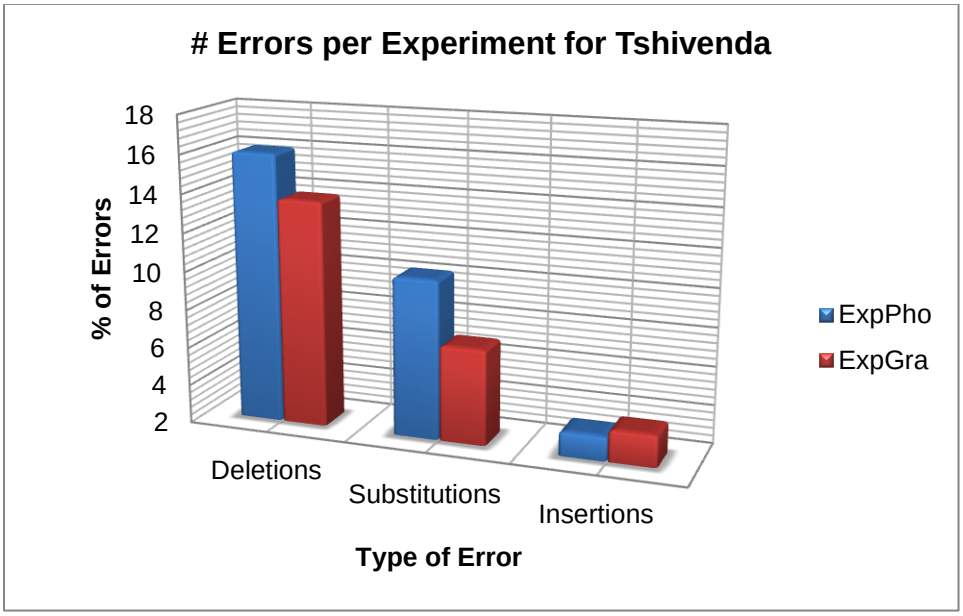

Figure 5.10: Number of recognition errors for each experiment in Tshivenda

Generally, both approaches handle all errors similarly for each language. The two approaches suffer the most deletion errors and the least insertion errors in all languages. The phoneme-based units are superior in recognizing short words and suffer a great number of substitutions in long words. Conversely, the grapheme-based units have a better recognition rate of long words, suffering only a few substitution errors and lots of deletions in short words. It was also noted that the phoneme-based units are superior in recognizing the foreign words, as one would expect.

Given the *unique* trend observed in Figure 5.14, the error handling by the two approaches for Sepedi, an investigation on how the number of GMMs affects the percentage of individual recognition errors was conducted for Sepedi. Moreover, Sepedi is more interesting than the rest of the languages since graphemes attained the highest WER reduction. Moreover, with the Lwazi ASR data (little training data), the grapheme-based units performed worse than the phoneme-based one but performed significantly better with the NCHLT ASR data (medium-sized training data). This observation opens a possibility for research.

The percentage of individual errors is analysed for each number of GMMs in both approaches. The phoneme-based experiment (ExpPho) is outlined in Figure 5.16 and grapheme-based experiment (ExpGra) in Figure 5.17.

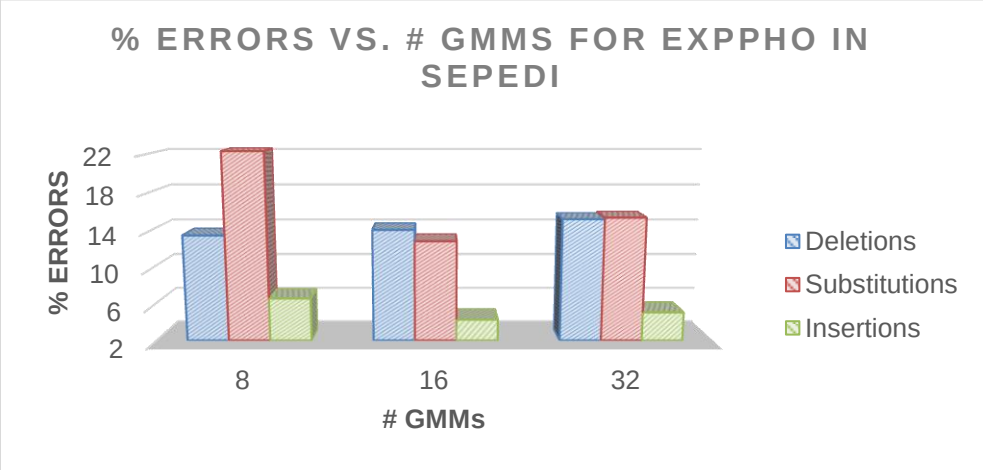

Figure 5.11: The percentage of errors against the number of GMMs for the phoneme-based experiment in Sepedi

As expected, the number of errors is minimal with 16 GMMs in both experiments. Interestingly, there is a spike in the number of substation errors for 8 GMMs. The number of substitution errors declined significantly with an increase in number of GMMs. In both experiments, insertions remained the least committed errors. Unlike the other two errors, deletions are only slightly affected by the increase of GMMs. Also, the number of deletion errors appears slightly uniform in both experiments.

It is evident from the figures that the grapheme-based units had a significant reduction of the substitution errors and hence the ultimate WER reduction was also significant. As reflected in Figure 5.17, the number of substitution errors for the grapheme-based experiment is almost half that of the phoneme-based experiments.

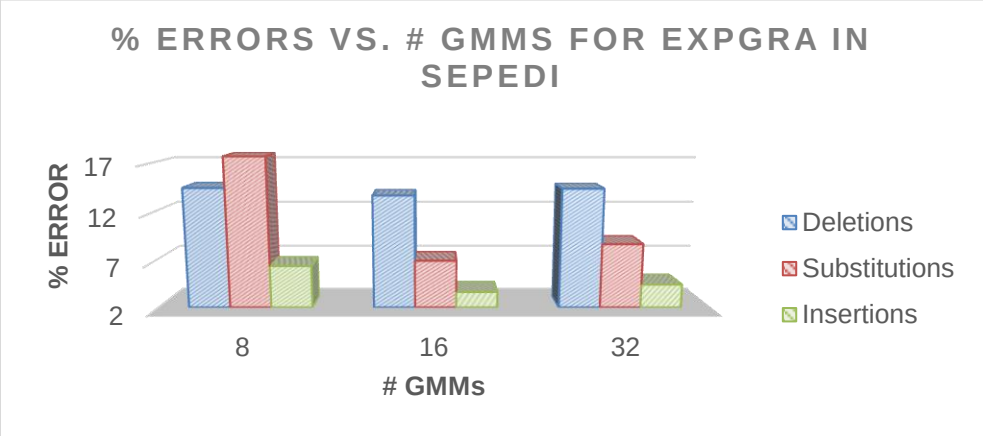

Figure 5.12: The percentage of errors against the number of GMMs for the grapheme-based experiment in Sepedi

Linguistic factors contributing towards the error trends and recognition accuracies were not investigated exhaustively because they are beyond the scope of this study. For example, one of the linguistic factors is the pronunciation modelling of the loan words, words used in a language they do not belong to. We used the primary language letter-to-sound rules to model loan words. Although our decision was guided by the findings in Modipa and Davel (2010), alternative methods to model loan words could achieve better recognition results. For instance, the odd results given by Tshivenda could imply that a different modelling technique for the loan words is required. It therefore remains

interesting to investigate the possible linguistic, technical and non-technical factors contributing to these trends.

5.6. Summary

This chapter briefly outlined some of the most common ASR performance evaluation metrics. We further discussed some of the important decoding parameters, and also described the recognition results generation procedure. The *HDecode* tool was used to evaluate the recognition performance of the context-dependent acoustic models in the two approaches. The grapheme-based and phoneme-based recognition results were presented, compared and analysed. The results obtained in this study were also compared to other studies of the same, and also of different languages. Analysis of the recognition errors was also presented. The next chapter provides a summary of our findings and recommendations for future work.

6. CONCLUSION

6.1. Introduction

In the previous chapter, we compared the performance of grapheme-based acoustic sub-word units to the phoneme-based ones. We showed that the context-dependent grapheme-based acoustic units can attain comparable and in some instances, even attaining better recognition accuracies. This chapter provides a summary of this research study, and also introduces ideas for future research.

6.2. Recognition Results

The ASR results obtained from our research study show that in some languages the grapheme-based acoustic sub-word units achieves acceptable levels of CSR accuracies when compared to phoneme-based units. On the small Lwazi ASR corpus, graphemes obtained better recognition accuracies than phonemes with a word error rate (WER) of 2.35% and 0.16% for Tshivenda and IsiNdebele, respectively. However, phonemes perform slightly better than graphemes for Sepedi, with a WER reduction of 0.73%. Phonemes also performed better than graphemes in the multilingual experiments, attaining a WER reduction of 2.37%.

The medium-sized NCHLT ASR corpus was used to further verify the results. The grapheme-based units remained better for IsiNdebele with a WER reduction of 0.54%. Graphemes also perform better for Sepedi with a WER reduction of 6.91%. However, the phoneme-based units perform slightly better for Tshivenda with a WER reduction of 0.04%, unlike they did on the Lwazi ASR cospus.

6.3. Summary of Findings

The aim of the research study was to investigate the potential of using graphemes, instead of phonemes, as acoustic sub-word units for the automatic speech recognition of the three under-resourced languages of the Limpopo, South Africa. To achieve this,

context-dependent tied-state acoustic models were trained using decision trees in both graphemes and phonemes. The grapheme-based and phoneme-based pronunciation dictionaries were used in the recognition process and the recognition results were compared for each language.

The research hypothesis which was tested in this study was formulated as follows: "*The grapheme-based acoustic sub-word units achieve acceptable levels of CSR accuracies when compared to phoneme-based units*". To validate this hypothesis, grapheme-based and phoneme-based experiments were conducted using two speech corpora: the Lwazi ASR speech corpus and the NCHLT ASR speech corpus. The two ASR experiments that used both speech corpora were identical with the only difference being the pronunciation dictionaries and thus the acoustic sub-word units. The primary evaluation focus of the two approaches was along monolingual speech recognition; however we also trained and tested multilingual acoustic models on the Lwazi ASR corpus. The Lwazi ASR corpus evaluation results were validated by the NCHLT ASR corpus.

For the Lwazi ASR corpus, it was found that the grapheme-based sub-word units performed better than phonemes in IsiNdebele and Tshivenda but 0.73% worse in Sepedi for the monolingual experiments. Furthermore, graphemes generally performed worse than phonemes in the multilingual experiment. However, phonemes did not perform better in the cross-lingual acoustic models for all target languages. Graphemes outperformed phonemes for IsiNdebele and Sepedi when the individual languages were tested on the cross-lingual acoustic models. Since data is shared across languages, Sepedi improved due to the increased training data sourced from the other two languages, namely Tshivenda and IsiNdebele. However, Tshivenda suffered a slight decrease in grapheme accuracy and saw phonemes perform better than they did in the monolingual models. This depends significantly on the structure of the languages being shared, as confirmed by Manaileng and Manamela (2014).

The Lwazi-based ASR results were validated by those obtained from using the NCHLT corpora. For IsiNdebele, the performance grapheme-based sub-word units remained

consistent and outperformed the phoneme-based units, just as it did in both the monolingual and multilingual experiments of the Lwazi corpora. With the increased training data, graphemes performed significantly better than phonemes for Sepedi. This correlates to the recognition results for the multilingual Lwazi corpora experiment, which also had more training data than the monolingual Lwazi corpora experiments. Tshivenda, however, did not improve with an increase of training data. It is suspected that the problem might be its language form and phonetic structure which was not easy to comprehend and manipulate by the researcher. Also, a similar study on the Afrikaans language also reported inferior graphemes performance due to language form and structure (Basson and Davel, 2013).

6.4. Future Work and Recommendations

This section discusses and recommends the potential directions of the future.

6.4.1. Pronunciation Dictionaries

Given the minimal effort required to build pronunciation dictionaries for the grapheme-based systems, as compared to the excessive effort required for the phoneme-based systems, we are confident that the use of graphemes can massively contribute towards the success of developing more pronunciation dictionaries and CSR systems for more under-resourced languages. The efficiency (in terms of cost and time) offered by graphemes also demonstrate their possible preference for under-resourced languages.

Furthermore, the primary language letter-to-sound rules were used to model loan words in the pronunciation dictionaries across all languages. Better recognition results may be obtained using different approaches such as language-specific letter-to-sound rules, data-driven foreign-to-primary language phoneme mappings, automatic language identification, etc.

6.4.2. Grapheme-based ASR Systems for More Under-resourced Languages

We hope to develop speech recognizers for more under-resourced languages of South Africa as these languages currently have few or non-existing speech processing tools. We have demonstrated that graphemes can improve monolingual speech recognition when cross-lingual data is shared across related languages to build multilingual acoustic models. This approach takes advantage of the uniformly distributed graphemes across the indigenous South African languages to increase training data. We therefore hope to use the grapheme-based approach to further train multilingual acoustic models on the related indigenous languages of South Africa.

6.4.3. Can Graphemes Solve the Problem of Language Variants?

One interesting observation that remains to be studied is the effects of using graphemes on language variants, such as dialects and accents. Dialects and accents are only spoken and not written, i.e. dialects or accents of a language are mostly written exactly the same way (or slightly differently) as the language itself. Graphemes are mainly concerned with the orthography of the language and thus it should be interesting to see how grapheme-based acoustic models handle these language variants.

6.4.4. Improved Recognition Accuracies

More work remains to be done to ensure satisfactory and reliable recognition results with significantly reduced recognition error rates so that the local speech processing research community can consider adopting this method to build continuous speech recognition systems for more languages with little or no linguistic resources. This will benefit various speaker communities that use most of these heavily under-resourced languages on daily basis by ensuring the delivery of automatic linguistic tools which may significantly help with language preservation, uplifting and general language-specific e-service provisioning tasks.

6.5. Final Remarks

This research essentially investigated the potential of grapheme-based sub-word units for monolingual and cross-lingual speech recognition for three indigenous South African languages, namely, the Sepedi, Tshivenda and IsiNdebele languages.

We have shown that grapheme-based continuous speech recognition, which copes with the problem of low-quality or unavailable pronunciation dictionaries, is comparable to phoneme-based recognition for these languages in both the monolingual and cross-lingual speech recognition tasks. The study significantly demonstrates that context-dependent grapheme-based sub-word units can be reliable for small and medium-large vocabulary speech recognition tasks for IsiNdebele, Sepedi and Tshivenda, and potentially other official languages of South Africa, as also suggested by Manaileng and Manamela (2014).

We demonstrated that graphemes can attain superior recognition accuracies for some under-resourced languages, preferably phonetic languages such Spedi, Tshivenda and IsiNdebele. This finding implies that for these under-resourced languages, graphemes can be considered alternatives to phonemes as sub-word recognition units to lessen the total effort and cost required in developing perfectly hand-crafted pronunciation dictionaries. This straightforward approach to pronunciation dictionary creation is advantageous especially in situation of under-resourced languages and can be successfully used for building more robust speech recognisers for rare and marginalized languages.

The method of applying graphemes-based acoustic sub-word units is novel for all the three languages. The results reported in our research study forms a baseline for further grapheme-based studies on these three and/or other under-resourced indigenous languages. Moreover, we have shown that the use of grapheme-based units attains better recognition accuracies in closely-related languages, such Sepedi and IsisNdbele, and worse accuracies in unrelated languages, Tshivenda and Sepedi. We further

demonstrated that Tshivenda is not suitable for cross-lingual data sharing with Sepedi and IsiNdebele, despite the languages' close socio-geographical proximity.

REFERENCES

Acero, A. 1993. *Acoustical and Environmental Robustness in Automatic Speech Recognition.* Kluwer Academic Publishers. Boston, MA.

Atal, B.S. and Hanauer, S.L. 1971. *Speech Analysis and Synthesis by Linear Prediction of the Speech Wave*. Journal of the Acoustical Society of America. Vol. 50. No 2. pp. 637-655.

Badenhorst, J. van Heerden, C. Davel, M. and Barnard, E. 2011. *Collecting and evaluating speech recognition corpora for 11 South African languages*. Language Resources and Evaluation. Vol. 45. pp. 289-309.

Bahl, L.R., Jelinek, F., and Mercer, R.L. 1993. *A Maximum Likelihood Approach to Continuous Speech Recognition*. IEEE Transitions on Pattern Analysis & Machine Intelligence (PAMI). Vol. 5. No. 2. pp. 179-190.

Barnard, E. Davel, M. and van Heerden, C. 2009. *ASR corpus design for resource-scarce languages*. In: Proceedings. INTERSPEECH. pp. 2847-2850.

Barnard, E. Davel, M. and van Huyssteen, G.B. 2010. *Speech technology for information access: a South African case study*. In Proceedings of the AAAI Spring Symposium on Artificial Intelligence for Development (AI-D). Palo Alto, California. pp. 8-13.

Barnard, E. Davel, M.H. van Heerden, C. de Wet, F. and Badenhorst, J. 2014. *The NCHLT Speech Corpus of the South African languages*. In Proceedings of SLTU-2014. St. Petersburg, Russia. pp. 194-200.

Basson, W.D. and Davel, M.H. 2012. *Comparing grapheme-based and phoneme-based speech recognition for Afrikaans*. In 23rd Annual Symposium of the Pattern Recognition Association of South Africa. PRASA 2012. CSIR International Convention Centre, Pretoria, pp. 144-148.

Besacier, L. Barnard, E. Karpovc, A. and Schultz, T. 2014. *Automatic speech recognition for under-resourced languages: A survey*. Speech Communication. Vol. 56. pp. 85-100.

Besling, S. 1994. *Heuristical and statistical methods for Grapheme-to-Phoneme conversion*. In Proceedings of Konvens. Wien, Austria, pp. 23-31

Beulen, K. Bransch, E. and Ney. H. 1997. *State tying for context dependent phoneme models*. In Proceedings of European Conf. on Speech Communication and Technology. Rhodos, Greece. pp. 1179-1182.

Black, A. and Llitjos, A. 2002. *Unit Selection Without a Phoneme Set*. In Proceedings of the IEEE TTS Workshop. Santa Monica, CA. pp. 77-80.

Black, A. Lenzo, K. and Pagel, V. 1998. *Issues in building general letter to sound rules*. In Proceedings of the ESCA Workshop on Speech Synthesis. Australia. pp. 77-80.

Cetin, O., 2008. Unsupervised adaptive speech technology for limited resource languages: a case study for Tamil. In: SLTU'08, Hanoi, Vietnam.

Chan, H.Y. and Rosenfeld, R. 2012. *Discriminative pronunciation learning for speech recognition for resource scarce languages*. In: Proceedings of the 2nd ACM Symposium on Computing for Development. Article No. 12.

Crystal, D. 2000. *Language Death*. Cambridge University Press. Cambridge.

Davel, M.H. and Martirosian, O. 2009. *Pronunciation dictionary development in resource-scarce environments*. In Proceedings of INTERSPEECH 2009. Brighton, UK. pp. 2851-2854.

Davel, M.H. Basson, W.D. Barnard, E. and van Heerden, C. 2013. *NCHLT Dictionaries: Project Report*. North-West University, Tech. Rep. [Online]. Accessed on September 2014. https://www.sites.google.com/sites/nchltspeechcorpus/home.

Davel, M.H. van Heerden, C. Kleynhans, N. and Barnard, E. 2011. *Efficient harvesting of Internet audio for resource-scarce ASR*. In: Proceedings. INTERSPEECH. pp. 3153-3156.

Davis, K.H. Biddulph, R. and Balashek, S. 1952. *Automatic Recognition of Spoken Digits*. Journal of the Acoustic Society of America. Vol. 24. No. 6. pp. 627-642.

Davis, S. and Mermelstein, P. 1980. *Comparison of Parametric Representations of Monosyllabic Word Recognition in Continuously Spoken Sentences*. IEEE Transitions on Acoustics Speech and Signal Processing. Vol. 28. No. 4. pp. 357-366.

De Vries, N.J. Davel, M.H. Badenhorst, J. Basson, W.D. de Wet, F. Barnard, E. and De Waal, A. 2013. *A smartphone-based ASR data collection tool for under-resourced languages*. Speech Communication. Vol. 56. pp. 119-131.

Ernst, G. Schukat-Talamazzini, H. Niemann, W. Eckert, T. Kuhn, and Rieck, S. 1993. *Automatic Speech Recognition without Phonemes*. In Proc. European Conf. on Speech Communication and Technology. Berlin. pp. 129-132.

Fry, D.B. and Denes, P. 1959. *The Design and Operation of the Mechanical Speech Recognizer at University College London*. Journal of the British Institute, Radio Engineers. Vol. 19. No. 4. pp. 219-229.

Gales, M. and Young, S. 1992. *An Improved Approach to the Hidden Markov Model Decomposition of Speech and Noise.* In Proc. ICASSP92.

Gales, M.J.F. and Young, S.J. 1995. *Robust speech recognition in additive and convolutional noise using parallel model combination*. Computer Speech and Language. Vol. 9. pp. 289-307.

Gauvain, J.L. and Lee, C.H. 1994. *Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains*. IEEE Transactions on Speech and Audio Processing. Vol. 2. pp. 291-298.

Gelas, H., Teferra, S. Besacier, L. and Pellegrino, F. 2011. *Quality assessment of crowd-sourcing transcriptions for African languages*. In: INTERSPEECH 2011. Florence. Italy. pp. 28-31.

Gemmeke, J.F. van Hamme, H. 2011. *A hierarchical exemplar-based sparse model of speech with an application to ASR.* IEEE ASRU 2011. HI, USA. ISBN: 978-1-4673-0365-1. pp. 101-106.

Godfrey, J.J. Holliman, E.C. and McDaniel, J. 1992. *SWITCHBOARD: telephone speech corpus for research and development*. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 1. pp. 517-520.

Gorin, A.L. Parker, B.A. Sachs, R.M. and Wilpon, 1996. J.G. *How May I Help You?* In Proc. Interactive Voice Technology for Telecommunications Applications (IVTTA), pp. 57-60.

Henselmans, D. Niesler, T. and van Leeuwen, D. 2013. *Baseline Speech Recognition of South African Languages using Lwazi and AST*. In 24th Annual Symposium of the Pattern Recognition Association of South Africa, PRASA 2013, Pretoria, pp. 30-35.

Hermansky, H. 1990. *Perceptual Linear Predictive (PLP) Analysis of Speech*. Journal of the Acoustical Society of America, Vol. 87, No 4. pp. 1738-1752.

Huang, X. Acero, A. and Hong, H. 2001. *Spoken Language Processing – A guide to Theory, Algorithm and System Development*. Prentice Hall, Inc.

Huang, X. and Lee, K.F. 1993. *On Speaker-Independent, Speaker-Dependent, and Speaker-Adaptive Speech Recognition*. IEEE Trans. On Speech and Audio Processing, pp. 150-157.

Hughes, T. Nakajima, K. Ha, L. Moreno, P. and LeBeau, M. 2010. *Building transcribed speech corpora quickly and cheaply for many languages*. In: Proceedings. INTERSPEECH. Makuhari. Japan. pp. 1914-1917.

Itakura, F. 1975. *Minimum Prediction Residual Principle Applied to Speech Recognition*. IEEE Trans. Acoustics, Speech and Signal Proc. Vol. ASSP-23. pp. 57-72.

Janda, M. 2012. *Grapheme Based Speech Recognition*. In Proceeding of the 18th Conference STUDENT EEICT 2012. Brno, CZ. Vol. 3. pp. 441-445.

Jelinek, F. 1991. *Up From Trigrams: the Struggle for Improved Language Models*. In Proceedings of Euro-Speech. Genoa. pp. 1037-1040.

Jelinek, F. Mercer, R.L. and Roukos, S. 1991. *Principles of lexical language modeling for speech recognition*. In Proceedings of Advances in Speech Signal Processing. Furui & Sondhi (eds.). Mercer Dekker, New York. pp. 651-699.

Jing, Z. and Min, Z. 2010. *Speech recognition system based improved DTW algorithm*. In: Proceedings. Int. Conf. on Computer, Mechatronics, Control and, Electronic Engineering CMCE-2010. Vol. 5. pp. 320-323.

Juang, B.H. 1991. *Speech Recognition in Adverse Environments.* Computer Speech and Language. Vol. 5. pp. 275-294.

Juang, B.H. and Rabiner, L.R. 1991. *Hidden Markov Models for Speech Recognition.* Technometrics. Vol. 33. pp. 251-272.

Juang, B.H. and Rabiner, L.R. 2004. *Automatic Speech Recognition – A Brief history of the technology development*. Georgia Institute of Technology, Atlanta.

Juang, B.H. Levinson, S.E. and Sondhi, M. 1986. *Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains*. IEEE Trans. Information Theory. Vol. It-32. No. 2. pp. 307-309.

Kanthak, S. and Ney, H. 2002. *Context-dependent Acoustic Modelling using Graphemes for Large Vocabulary Speech Recognition*. In Proceedings of ICASSP. Orlando, Florida. pp. 845-848.

Kanthak, S. and Ney, H. 2003. "*Multilingual Acoustic Modelling Using Graphemes*". In Proceedings of European Conference on Speech Communication and Technology. Geneva, Switzerland. Vol. 2. pp. 1145-1148.

Katz. S.M. 1987. *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recogniser*. IEEE Transitions on Acoustic Speech and Signal Processing. Vol 35, No. 3. pp. 400-401.

Kenny, P. 1991. *A* Admissible Heuristics for Rapid Lexical Access*. In Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP). Toronto. pp. 689-692.

Killer, M. 2003. *Grapheme Based Speech Recognition*. Interactive Systems Laboratory. Pittsburgh. PA, USA.

Klatt, D.H. 1977. *Review of the ARPA Speech Understanding Project*. Journal of the Acoustical Society of America. Vol. 62. pp. 1345-1366.

Krauwer, S. 2003. *The basic language resource kit (BLARK) as the first milestone for the language resources roadmap*". *In Proceedings of the International Workshop Speech and Computer. SPECOM-2003*. Moscow. Russia. pp. 8-15.

Le, V. and Besacier, L. 2009. *Automatic speech recognition for under-resourced languages: application to Vietnamese language*. IEEE Transactions on Audio, Speech and Language Processing. Vol. 17. No. 8. pp. 1471-1482.

Lee, C.H. Rabiner, L.R. Pieraccini, R. and Wilpon, J.G. 1990. *Acoustic Modelling for large vocabulary speech recognition*. Computer Speech & Language. Vol. 4. pp. 1237-1265.

Leggeter, C. and Woodland, P. 1995. *Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models*. Computer Speech and Language. Vol. 9. pp. 171-185.

Liporace, L.A. 1982. *Maximum Likelihood Estimation for Multivariate Observations of Markov Sources*. IEEE Trans. On Information Theory. Vol. IT-28. No. 5. pp. 729-734.

Lippmann, R.P. 1989. *Review of Neural Networks for Speech Recognition*. Neural Computation. Vol. 1. No. 1. pp. 1-38.

Liu, F. Stern, R. Huang, H. and Acero, A. 1993. *Efficient Cepstral Normalization for Robust Speech Recognition*. In Proceedings of ARPA Human Language Technology Workshop, pp. 69-74.

Loof, J., Gollan, C., Ney, H., 2009. Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a Polish speech recognition system. In: INTERSPEECH 2009. Brighton, UK.

Mabokela, K.R. 2014. *Automatic language identification using word segments on mixed-language speech*. Masters Dissertation. School of Mathematical and Computer Science, Faculty of Science and Agriculture, University of Limpopo (Turfloop Campus), South Africa.

Manaileng, M. and Manamela, M. 2014. *Graphemes and Phonemes as Acoustic Sub-word Units for Continuous Speech Recognition of Under-resourced Languages*. Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2014, ISBN: 978-0-620-61965-3. Nelson Mandela Bay, Eastern Cape, South Africa. pp. 93-98.

Manaileng, M. J. Manamela, M.J.D. 2013. *Connected-digits Recognition for an Under-resourced Language Using Hidden Markov Models*. In Proceedings of ELMAR-2013. Zadar, Croatia. pp. 211-214.

Mariani, J. 1991. *Knowledge-Based Approaches versus Mathematical Model Based Algorithms: the Case of Speech Recognition*. Proceedings of 30th Conference on Decision and Control. Brighton. pp. 841-846.

Meraka-Institute. 2009. *Lwazi ASR corpus*. Online: http://www.meraka.org.za/lwazi

Modipa, T. and Davel, M.H. 2012. *Pronunciation Modelling of Foreign Words for Sepedi ASR*. In 21st Annual Symposium of the Pattern Recognition Association of South Africa, PRASA 2012, Stellenbosch, South Africa, pp. 185-189.

Mohamed, A. Dahl, G.E. and Hinton, G. 2012. *Acoustic modelling using deep belief networks*. IEEE Transactions on Audio, Speech, and Language Processing. Vol. 20. No. 1. pp. 14-22.

Mohri, M. 1997. *Finite-State Transducers*. In Language and Speech Processing, Computational Linguistics. Vol. 23. No. 2. pp. 269-312.

Moreno, P. April 1996. *Speech recognition in noisy environments*. Ph.D. dissertation. ECE Department, Carnegie-Mellon University.

Muthusamy, Y.K. and Cole, R.A. 1992. *Automatic segmentation and identification of ten languages using telephone speech*. In: Second International Conference on Spoken Language Processing. Vol. 2. pp. 1007-1010.

Neumeyer, L. and Weintraub, M. 1995. *Robust Speech Recognition in Noise Using Adaptation and Mapping Techniques.* In Proceedings of ICASSP. Vol. 1, Detroit, USA. pp. 141-144.

O'Brien, S.M. 1993, *Knowledge-based systems in speech recognition: a survey*. International Journal of Man-Machine Studies. Vol. 38. pp. 71-95.

Parent, G. and Eskenazi, M. 2010. *Toward better crowd-sourced transcription: transcription of a year of the Let's Go bus information system data*. In: Proceedings of IEEE Workshop on Spoken Language Technology. Berkeley. California. December 2010. pp. 312-317.

Paul, D.B. 1991. *Algorithms for an Optimal A* Search and Linearizing the Search in the Stack Decoder*. In Proceedings of ICASSP. pp. 693-996. Toronto.

Plahl, C. Schlueter, R. and Ney, H. 2011. *Cross-lingual portability of Chinese and English neural network features for French and German LVCSR*. In: Proceedings. ASRU, USA. pp. 371-376.

Poritz, A.B. 1982. *Linear predictive hidden Markov models and the speech signal*. In Proc. ICASSP-82. Paris, France. pp. 1291-1294.

Rabiner, L.R. 1989. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE. Vol. 77. No. 2. pp. 257-286.

Rabiner, L.R. and Gold, B. 1975. *Theory and Applications of Digital Signal Processing*. Prentice-Hall. Englewood Cliffs. NJ.

Rabiner, L.R. and Juang, B.H. 1985. *An Introduction to Hidden Markov Models*. IEEE Signal Processing Magazine.

Schultz, T. 2002. *GlobalPhone: a multilingual speech and text database developed at Karlsruhe University*. In: ICSLP. pp. 345-348.

Schultz, T. Kirchhoff, K. 2006. *Multilingual speech processing*. Elsevier, Academic Press, ISBN 13: 978-0-12-088501-5.

Schultz, T. and Waibel, A. 1998. *Language independent and language adaptive large vocabulary speech recognition*. In Proc. Int. Conf. on Spoken Language Processing. Sydney, Australia. pp. 1819-1822.

Schultz, T. and Waibel, A. 2001. *Language independent and language adaptive acoustic modelling for speech recognition*. Speech Communication. Vol. 35. pp. 31-51.

Seide, F. Li, G. Chen, X. and Yu, D. 2011. *Feature engineering in context-dependent deep neural networks for conversational speech transcription*. In: Proceedings. ASRU-2011 International Workshop. HI. USA. pp. 24-29.

Singh, R. Raj, B. and Stern, R. 2002. *Automatic Generation of Subword Units for Speech Recognition Systems*. IEEE Transactions on Speech and Audio Processing. Vol. 10. pp. 98-99.

Siniscalchi, S.M. Reed, J. Svendsen, T. and Lee, C.H. 2013. *Universal attribute characterization of spoken languages for automatic spoken language recognition*. Computer Speech & Language. Vol. 27. No. 1. pp. 209-227.

Sirum, P. and Sanches I. 2010. *Automatic Speech Recognition with Graphemes and Phonemes in Portuguese*. In Proceedings of IWSSIP 2010. Reo de Janeiro, Brazil. pp. 320-323.

Solera-Urena, R. Martın-Iglesias, D. Gallardo-Antolın, A. Pelaez-Moreno, C. and Dıaz-de-Marıa, F. 2007. *Robust ASR using support vector machines*. Speech Communication. Vol. 49. No. 4. pp. 253-267.

STATISTICS South Africa (Stats SA). 2012. *Census 2011 – Census in brief*. Online: *http://www.statssa.gov.za/Census2011/Products.asp*.

Stephenson, T.A. Escofet, J. Magimai-Doss, M. and Bourlard, H. 2002. *Dynamic Bayesian network based speech recognition with pitch and energy as auxiliary variables*. Technical Report Idiap-RR-24-2002. pp.10.

Stolcke, A. Grezl, F. Hwang, M. Lei, X. Morgan, N. and Vergyri, D. 2006. *Cross-domain and cross-lingual portability of acoustic features estimated by multilayer perceptrons*. In: Proceedings. IEEE ICASSP-2006. pp. 321-324.

Stuker, S. 2008a. *Integrating Thai Grapheme Based Acoustic Models into the ML-MIX Framework – For Language Independent and Cross-Language ASR*. In Proc. Of the 2008 Spoken Languages Technologies for Under-resourced Languages (SLTU). pp. 27-32.

Stuker, S. 2008b. *Modified Polyphone Decision Tree Specialization for Porting Multilingual Grapheme Based ASR Systems to New Languages*. In Proc. Of ICASSP. pp. 4249-4252.

Stuker, S. and Schultz, T. 2004. "*A Grapheme Based Speech Recognition System for Russian*. SPECOM'2004: 9th Conference Speech and Computer, St. Petersburg, Russia, pp. 297-303.

Stuker, S. Schultz, T. Metze, F. and Waibel, A. 2003. *Multilingual articulatory features*. In: Proceedings of ICASSP'03 IEEE International Conference on Acoustics, Speech, and, Signal Processing. Vol. 1 pp. I-144-I-147.

Tachbelie, M. Abate, S.T. Besacier, L. 2014. *Using different acoustic, lexical and language modelling units for ASR of an under-resourced language – Amharic*. Speech Communication. Vol. 56. pp. 181-194.

Tachbelie, M. Abate, S.T. Besacier, L. and Rossato, S. 2012. *Syllable-based and hybrid acoustic models for Amharic speech recognition*. In: SLTU – Workshop on Spoken

Language Technologies for Under-Resourced Languages. Cape Town. South Africa. pp. 5-10.

Thomas, S. Ganapathy, S. and Hermansky, H. 2012. *Multilingual MLP features for low-resource LVCSR systems*. In: Proceedings. ICASSP, Kyoto, Japan. pp. 4269-4272.

Toth, L. Frankel, J. Gosztolya, G. and King, S. 2008. *Cross-lingual portability of MLP-based tandem features – a case study for English and Hungarian*. In: Proceedings. INTERSPEECH. Australia. 2695-2698.

Trentin, E. and Gori, M. 2001. *A survey of hybrid ANN/HMM models for automatic speech recognition*. Neurocomputing. Vol 37. No. 1. pp. 91-126.

Ulla, U. 2001. *Multilingual speech recognition in seven languages*. Speech Communication. Vol. 35. pp. 53-69.

van Heerden, C. Badenhorst, J. de Wet, F. and Davel, M. 2013. *Lwazi asr evaluation*. CSIR, Tech. Rep.

van Heerden, C. Barnard, E. and Davel, M. 2009. *Basic speech recognition for spoken dialogues*. In Proc. INTERSPEECH. Brighton, UK. pp. 3003-3006.

van Heerden, C. Davel, M.H. and Barnard, E. 2013. *The semi-automated creation of stratified speech corpora*. In Proceedings. The 24th Annual Symposium of the Pattern Recognition Association of South Africa. PRASA 2013. Johannesburg, South Africa. pp. 115-119.

van Heerden, C. Kleynhans, N. Barnard, E. and Davel, M. 2010. *Pooling ASR data for closely related languages*. In Proceedings of the Workshop on Spoken Languages Technologies for Under-Resourced Languages (SLTU 2010). Penang, Malaysia. pp. 17-23.

van Heerden, C.J. Davel, M.H. and Barnard, E. 2012. *Medium-vocabulary speech recognition for under-resourced languages*. In Proceedings Workshop on Spoken Languages Technologies for Under-Resourced Languages (SLTU 2012), Cape Town, South Africa, pp. 146-151.

Vesely, K. Karafiat, M. Grezl, F. Janda, M. and Egorova, E. 2012. *The language-independent bottleneck features*. In: Proceedings. SLT. USA.

Viikki, O. and Laurila, K. 1998. *Cepstral domain segmental feature vector normalization for noise robust speech recognition*. Speech Communication. Vol. 25. No. 1-3. pp. 133-147.

Vimala, C. and Radha, V. 2012. *A Review on Speech Recognition Challenges and Approaches*. World of Computer Science and Information Technology Journal (WCSIT). Vol. 2. No. 1. ISSN: 2221-0741. pp. 1-7.

Vu, N.T., Kraus, F., Schultz, T., 2011. Rapid building of an ASR system for under-resourced languages based on multilingual unsupervised training. In: Proceedings. INTERSPEECH, Italy.

Whittaker, E.W.D. and Woodland, P.C. 2001. *Efficient class-based language modelling for very large vocabularies*. In: ICASSP-2001. Salt Lake City. USA. pp. 545–548.

Wilpon, J.G. Rabiner, L.R. Lee, C.H. and Goldman, E.R. 1990. *Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models*. IEEE Trans. On Acoustics, Speech and Signal Processing. Vol. 38. No. 11. pp. 1870-1878.

Wiqas, G. and Navdeep, S. March 2012. *Literature Review on Automatic Speech Recognition*. International Journal of Computer Applications (0975 – 8887), Vol 41. No.8. pp. 42-50.

Wissing, D. and Barnard, E. 2008. *Vowel variations in Southern Sotho: an acoustical investigation*. Southern African Linguistics and Applied Language Studies. Vol. 26. No. 2. pp. 255-265.

Young, S. 1996. *A Review of Large-Vocabulary Continuous-Speech*. Signal Processing Magazine. IEEE. Vol. 13. No. 5.

Young, S. Evermann, G. Gales, M. Hain, T. Kershaw, D. Liu, X. Moore, G. Odell, J. Ollason, D. Povey, D. Valtchev, V. and Woodland, P. 2006. *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department.

Young, S., 2008. *HMMs and Related Speech Recognition Technologies*. In: Springer Handbook of Speech Processing. Springer-Verlag. Berlin Heidelberg. pp. 539-557.

Yu, D. Siniscalchi, S.M. Deng, L. and Lee, C.H. 2012. *Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition*. In: Proceedings. ICASSP-2012. pp. 4169-4172.

APPENDICES

A: ASR Experiments – system.sh

```
#!/bin/bash
# This script calls other scripts to run an ASR experiment for a given language
ROOT_DIR=~/asr
MAIN_DIR1=$ROOT_DIR/msc/nchlt/nchlt_nso
ASR_SCRIPTS=$ROOT_DIR/asr_template
source Vars.sh
FEAT=1
PREPROC=1
DO_LISTS=1
DO_CHECK=1
DO_TRAIN=1
DO_EVAL=1
#-------------------------------------------------------------
# Create necessary directories!!

for dir in  $DIR_EXP/data $DIR_EXP/data/mfccs $DIR_EXP/data/mfccs/trn $DIR_EXP/data/mfccs/tst
$DIR_EXP/log $DIR_EXP/data/proc_trans $DIR_EXP/data/proc_trans/trn $DIR_EXP/data/proc_trans/tst
$DIR_EXP/lists/; do
  if [ ! -d $dir ]; then
     mkdir -p $dir
  fi
done

#-------------------------------------------------------------------
if [ $FEAT == 1 ]; then
    echo ""
    echo "FEATURE EXTRACTION using CMVN"
    echo "creating hcopylist.lst"
    date >> $DIR_EXP/log/time.feat
    perl $ASR_SCRIPTS/utility_scripts/create_hcopy_lists.pl $MAIN_DIR1/data/audio
$DIR_EXP/data/mfccs $DIR_EXP/lists/hcopylist.lst
    cd $DIR_EXP/src
    echo "running: CMVN.sh cmvn"
    bash CMVN.sh cmvn $DIR_EXP/lists/hcopylist.lst >& $DIR_EXP/log/feature.log
    date >> $DIR_EXP/log/time.feat
fi
#-------------------------------------------------------------

if [ $PREPROC == 1 ]; then
    echo ""
    echo "PREPROCESSING of Transcriptions"
    echo "creating preproclist.lst"
    date >> $DIR_EXP/log/time.pre
    perl $ASR_SCRIPTS/utility_scripts/create_preproc_lists.pl $MAIN_DIR1/data/trans/trn
$DIR_EXP/data/proc_trans/trn $DIR_EXP/lists/preproclist.trn.lst
    perl $ASR_SCRIPTS/utility_scripts/create_preproc_lists.pl $MAIN_DIR1/data/trans/tst
$DIR_EXP/data/proc_trans/tst $DIR_EXP/lists/preproclist.tst.lst
    cd $DIR_EXP/src
```

```
    echo "running: PREPROC.sh"
    bash PREPROC.sh $DIR_EXP/lists/preproclist.trn.lst all_phases >& $DIR_EXP/log/preproc.trn.log
    bash PREPROC.sh $DIR_EXP/lists/preproclist.tst.lst all_phases >& $DIR_EXP/log/preproc.tst.log
    date >> $DIR_EXP/log/time.pre
fi
#-------------------------------------------------------------------
if [ $DO_LISTS == 1 ]; then
    echo "Generating train and test lists"
    date >> $DIR_EXP/log/time.lists
    bash gen_nchlt_lists.sh
    date >> $DIR_EXP/log/time.lists
fi
#-------------------------------------------------------------------------------
if [ $DO_CHECK == 1 ]; then
    echo ""
    echo "CHECKING for Errors prior to Training"
    cd $DIR_EXP/src
    echo "running: CHECK.sh all_phases"
    date >> $DIR_EXP/log/time.check
    bash CHECK.sh all_phases >& $DIR_EXP/log/check.log
    date >> $DIR_EXP/log/time.check
fi
#---------------------------------------------------------------------
if [ $DO_TRAIN == 1 ]; then
    echo ""
    echo "TRAINING"
    cd $DIR_EXP/src
    date >> $DIR_EXP/log/time.train
    echo "running: TRAIN.sh all_phases"
    bash TRAIN.sh all_phases >& $DIR_EXP/log/train.log
    echo "running: TRAIN.sh semitied"
    bash TRAIN.sh semitied >& $DIR_EXP/log/semitied.log
    date >> $DIR_EXP/log/time.train
fi
#--------------------------------------------------------------------
if [ $DO_EVAL == 1 ]; then
    echo ""
    echo "EVALUATION"
    cd $DIR_EXP/src
    echo "running: HDecode"
    bash runHDecode.sh
    echo "running: TEST.sh words_results"
    bash TEST.sh word_results >& $DIR_EXP/log/results.words.log
    date >> $DIR_EXP/log/time.test
fi
echo "DONE."
```

B: Data Preparation – create_trans.py

```
#!/usr/bin/python
# This script creates training and testing transcriptions of the NCHLT Data

from xml.dom import minidom
```

```
import xml.etree.ElementTree as ET
import sys, codecs

def getTrans(a, txt):
	fname = a[20:-3] + 'txt'
	print 'Writting to: ' + fname
	print ""
	fhandle = codecs.open(fname, 'w', 'utf-8')
	fhandle.write(txt)
	fhandle.close()

def makeTrans( fl ):
	print 'Reading XML file...'
	xmldoc = minidom.parse(fl)
	recs = xmldoc.getElementsByTagName('recording')
	print 'Reading DONE.'
	for rec in recs:
		aud = rec.attributes['audio'].value
		trn = rec.getElementsByTagName('orth')[0].childNodes[0].data
		sent = trn + "\n"
		getTrans(aud, sent)
if __name__=='__main__':
	fl = sys.argv[1]

	makeTrans( fl )
```

C: Generating Question Files: create_quest.pl

```
# This script creates question file for triphone tying from a list of graphemes or monophones
#!/usr/bin/perl
use warnings;
use strict;
use open IO => ':encoding(utf8)';
my $monophone_list;
my $quest_file;
($quest_file, $monophone_list) = @ARGV;
if (@ARGV + 0 < 2) {
  print "./create_quests_file.pl <quests_file_out> <monophn_list>\n";
  exit 1;
}
my $ph;
my @elements;
my %monophones;
open IN, "$monophone_list";
while(<IN>) {
  chomp($ph = $_);
  if (($ph ne "sil")&&($ph ne "sp")) {
    $monophones{$ph} = 1;
  }
}
close(IN);
open OUT, ">$quest_file";
foreach $ph (sort keys %monophones) {
```

```
  print OUT "QS  \"R_$ph\"\t\t{ *+$ph }\n";
}
foreach $ph (sort keys %monophones) {
  print OUT "QS  \"L_$ph\"\t\t{ $ph-* }\n";
}
close(OUT);
```

D: Generating wordlists: gen_word_list.py

```
# This script generates a wordlist from a given pronunciation dictionary
#!/usr/bin/python
import sys
def genWordList1(wordList):
        for word in wordList:
                inc = 0
                for c in word:
                        if c != '\t':
                                inc = inc + 1
                        else:
                                break
                wrd = word[:inc]
                lst.append(wrd)

        return lst

def genWordList2(lst):

        flst = []

        for word in lst:
                wrd = word[:100]
                flst.append(wrd + '\n')

        return flst

if __name__ == '__main__':
    if len(sys.argv) < 2:
        print "Usage: " + sys.argv[0] +" inputfile "
        sys.exit()

    inFile = open(sys.argv[1],'r')

    wordList = []
    lst = []
    flst = []
    wrd = ""

    for ln in inFile:
        line = ln.strip()
        wordList.append(line)

    lst = genWordList1(wordList)
```

```
    flst = genWordList2(lst)

    toF = open('WordsList.txt','w')
    toF.writelines(flst)
    toF.close()
```

E: Creating Grapheme-based Pronunciation Dictionaries: create_dict.py

```
# This script creates a grapheme-based pronunciation dictionary from a wordlist
#!/usr/bin/python

import sys, codecs
def createDict(inFile):
        wordList = []
        flst = []
        wrd = ""
        for ln in inFile:
        line = ln.strip()
        wordList.append(line)

        for wrd in wordList:
                chars = ''
                word = wrd

                for char in wrd:
                        chars = chars + char + ' '
                        #print chars,
                word = word +'\t\t\t' + chars
                #print word
                flst.append(word + '\n')
        toF = codecs.open("Dict.txt","w","utf-8")
        toF.writelines(flst)
        toF.close()

if __name__ == '__main__':
    if len(sys.argv) < 2:
        print "Usage: " + str(sys.argv[0]) + " inputfile (.txt)"
        sys.exit()
    inFile = codecs.open(sys.argv[1],"r","utf-8")
    createDict(inFile)
```

F: Increasing Tri Mixtures: tri_inc_mixes.sh

```
# This script increments the number of mixtures
#!/bin/bash
LOCAL_NUM_ITERATIONS=$NUM_MIXES

while [ $LOCAL_NUM_ITERATIONS -gt 1 ]; do
  LOCAL_NUM_ITERATIONS=$(($LOCAL_NUM_ITERATIONS-1))
#=============================================================================
# Increment the mixtures
#=============================================================================
```

```
# Make sure the hmm dirs are up to date
  source $DIR_SRC/inc_hmm_cnt.sh auto_update
  HHEd -A -D -T 1 -V -H $DIR_HMM_CURR/macros -H $DIR_HMM_CURR/hmmDefs.mmf -M
$DIR_HMM_NEXT $HED_MIX_INC $LIST_TIED
  bash $DIR_SRC/check_exit_status.sh $0 $?

#============================================================================
# Re-estimate twice
#============================================================================
# ./herest.sh <model list> <trn mlf> <num re-estimations>
  bash $DIR_SRC/herest.sh $LIST_TIED $MLF_TRIPHNS_TRN 2
  bash $DIR_SRC/check_exit_status.sh $0 $?
echo done
```

G: Conference Publications

Full Paper

Manaileng, M.J. Manamela, M.J. 2014. *Graphemes and Phonemes as Acoustic Sub-word Units for Continuous Speech Recognition of Under-resourced Languages*. Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2014, ISBN: 978-0-620-61965-3, Nelson Mandela Bay, Eastern Cape, South Africa, pp. 93-98.

Short Paper

Manaileng, M.J. Manamela, M.J. 2013. *Grapheme-based Continuous Speech Recognition for Some of the Under-resourced Languages of Limpopo Province*. Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2013, ISBN: 978-0-620-57882-0, Spier Wine Estate, Stellenbosch, Western Cape, South Africa, pp. 387-388.