DEVELOPMENT OF ROBUST LANGUAGE MODELS FOR SPEECH RECOGNITION OF UNDER-RESOURCED LANGUAGES

by

DANIEL SINDANA

RESEARCH DISSERTATION

Submitted in 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. MJD MANAMELA

CO-SUPERVISOR: Dr TI MODIPA

2020

Dedication

This work is dedicated to my late mother, ***Thembi Joyce Sindana****, may her soul continue to flourish in the other worlds of God.*

Declaration

I, Daniel Sindana, hereby declare that the work in this research dissertation entitled, “DEVELOPMENT OF ROBUST LANGUAGE MODELS FOR SPEECH RECOGNITION OF UNDER-RESOURCED LANGUAGES”, is my original work and has never been submitted to any institution of higher learning for degree or examination purposes. All the references cited and referenced from materials and works of other researchers have been duly acknowledged.

D. Sindana

Acknowledgements

The completion of this work, to the extent in which it was submitted, would not have been possible were it not for the immense contribution, guidance and support of the following esteemed people and institutions:

My dearest supervisors: Mr. MJD Manamela and Dr TI Modipa. You served a capacity beyond mentorship for me. Your supervision, guidance, mentorship and support ever since Honours level is in many ways acknowledged. I aspire to continue studying beyond Masters, and thus wish to continue learning through your assistance.

Mabu Manailaneng, thank you for the script that helped create text files from the xml transcription files.

These Institutions: University of Limpopo (UL), Telkom SA (Ltd), Telkom Centre of Excellence for Speech Technology at UL, and the National Research Foundation (NRF); for the study platform, financial backing and research support. May these Institutions grow to attain greater heights and continue providing academic light to humanity.

My family, own and extended, you are my bedrock. Everything I do is ultimately for you and through you.

Colleagues and friends: the formal and informal spaces we always share impart lots of learnings, let us soldier on and strengthen the bonds of friendship furthermore.

Personal mentors (former Teachers and Lecturers, and Friends in the Baha'i community): thank you for the wisdom you graciously provide whenever I reach out to you.

Most importantly, to the Author of this life, the Creator, the All-Knowing, the All-Wise: *Allah'u'Abhá*.
Ngiyathokoza.

Abstract

Language modelling (LM) work for under-resourced languages that does not consider most linguistic information inherent in a language produces language models that inadequately represent the language, thereby leading to under-development of natural language processing tools and systems such as speech recognition systems. This study investigated the influence that the orthography (i.e., writing system) of a language has on the quality and/or robustness of the language models created for the text of that language. The unique conjunctive and disjunctive writing systems of isiNdebele (Ndebele) and Sepedi (Pedi) were studied.

The text data from the LWAZI and NCHLT speech corpora were used to develop language models. The LM techniques that were implemented included: word-based n-gram LM, LM smoothing, LM linear interpolation, and higher-order n-gram LM. The toolkits used for development were: HTK LM, SRILM, and CMU-Cam SLM toolkits.

From the findings of the study – found on text preparation, data pooling and sizing, higher n-gram models, and interpolation of models – it is concluded that the orthography of the selected languages does have effect on the quality of the language models created for their text. The following recommendations are made as part of LM development for the concerned languages. 1) Special preparation and normalisation of the text data before LM development – paying attention to within sentence text markers and annotation tags that may incorrectly form part of sentences, word sequences, and n-gram contexts. 2) Enable interpolation during training. 3) Develop pentagram and hexagram language models for Pedi texts, and trigrams and quadrigrams for Ndebele texts. 4) Investigate efficient smoothing method for the different languages, especially for different text sizes and different text domains.

Keywords: *Language modelling, automatic speech recognition, natural language processing, under-resourced languages*

Table of Contents

Abbreviations

Abbreviation	Expansion
AD	Absolute Discounting
ALU	Automatic Language Understanding
AS	Additive Smoothing
ASR	Automatic Speech Recognition
AST	African Speech Technology
CCER	Character-Cluster Error Rate
CMU-Cam SLM	Carnegie Mellon University – Cambridge University Statistical Language Modelling
CSIR	Centre for Scientific and Industrial Research
DAC	Department of Arts and Culture
DFT	Discrete Fourier Transform
DNN LM	Deep Neural Network Language Modelling
DST	Department of Science and Technology
FLM	Factored Language Modelling
FSM	Finite State Machine
GPU	Graphics Processing Unit
GT	Good-Turing
HLT	Human Language Technologies
HMI	Human Machine Interaction
HMMs	Hidden Markov Models
HTK	Hidden Markov Model Toolkit
HTK LM	Hidden Markov Model Toolkit Language Modelling
ICA	Independent Component Analysis
ICS	Intelligent Call Steering
IDFT	Inverse Discrete Fourier Transform
IoT	Internet of Things
IVR	Interactive Voice Response
KN	modified Kneser-Ney
LD	Linear Discounting
LDA	Linear Discriminate Analysis

Abbreviation	Expansion
LM	Language Modelling
LMs	Language Models
LPC	Linear Predictive Coding
Ltd	Limited
LVCSR	Large Vocabulary Continuous Speech Recognition
ME	Maximum Entropy
MFCCs	Mel Frequency Cepstrum Coefficients
MLE	Maximum Likelihood Estimation
NCHLT	National Centre for Human Language Technologies
ND	Natural Discounting
NLP	Natural Language Processing
NNLM	Neural Network Language Modelling
NRF	National Research Foundation
NWU	North West University
OOV	Out-of-Vocabulary
PCA	Principal Component Analysis
POS	Part-of-Speech
PPL	Perplexity
R&D	Research and Development
RNNLM	Recurrent Neural Network Language Modelling
RMA	Resource Management Agency
SA	South Africa(n)
SDSs	Spoken Dialogue Systems
SER	Syllable Error Rate
SLM	Statistical Language Model
SLU	Spoken Language Understanding
SM	Smoothing Method
SRILM	Stanford Research Institute Language Modelling
STT	Speech-to-Text
UKN	unmodified Kneser-Ney
UL	University of Limpopo

Abbreviation	Expansion
WB	Witten-Bell
WER	Word Error Rate
WSJ	Wall Street Journal

Chapter 1: Introduction

Language modelling (LM) is a process of developing models of word sequences that capture the regularities of a language such as syntactic, semantic and pragmatic characteristics to determine the likelihood of unknown word sequences as being legal or valid sequences of the language [1] [2]. Together with feature vectors, acoustic and lexical models, language models are used by an automatic speech recognition (ASR) system when attempting to transcribe utterances into their corresponding textual representation.

This chapter summarises the research work that is reported in this dissertation document. The problem area of investigation is briefly described in the Section 1.1. Section 1.2 defines the characterisation of a disjunctive and conjunctive language. Section 1.3 elaborates on some of the reasons why an investigation such as the one undertaken by this study may be deemed significant. The section continues to outline the aim and objectives behind the study. The structure of the rest of the dissertation is laid out in the last section of the Chapter, Section 1.4.

1.1. Problem Statement

The ideal situation in the development of natural language processing (NLP) applications and tools such as ASR systems is to incorporate as much linguistic information in their processing as possible. This step produces systems that accurately and sufficiently model a natural language. Language models better represent a language when they consider most linguistic information of that language [2].

However, a major challenge is modelling the unique linguistic attributes (such as special phonological, morphological and orthographic systems) of most under-resourced languages. This is one of the problems faced with when porting NLP systems and techniques to processing new languages [3]. The Southern Bantu[1] languages fall at the centre of the category of under-resourced languages, with the South African (SA) Nguni and Sotho language classes included [3].

[1] *https://global.britannica.com/topic/Bantu-languages*

Under-resourced languages are languages with lack of a unique or stable orthography, have limited presence on the web, lack linguistic expertise, lack electronic resources for speech and language processing such as monolingual corpora, bilingual electronic dictionaries, transcribed speech data, pronunciation dictionaries, vocabulary lists, language models and so on [3]. The failure to enhance the LM work for under-resourced languages does not help in their development, their language processing tools use obsolete and inefficient technology and techniques. Insufficient modelling of the unique attributes of the languages further aggravates the LM underperformances. Thus, the spoken language processing systems such as ASR systems cannot accu- rately recognise input utterances in the absence of robust language models.

To develop additional spoken technology resources, this study developed different language models for the unique linguistic attributes of the South African isiNdebele (autonym for Ndebele) and Sepedi (autonym for Pedi) languages. The development was conducted using proven methods and techniques for improved LM performance. The linguistic attribute of these languages that was taken into consideration is their conjunctive and disjunctive writing forms. The Ndebele language is typically conjunctively-written, meaning that morphemes such as nominal concords, prefixes and stems are clustered or combined to form words - whereas morphemes are written disjunctively to make words for the disjunctive Pedi [4]. For example, the Ndebele word *“ngiyakuthanda” (I love you)* is made up by the morphemes *ngi-, -ya-, -ku-, -thand-, and –a;* whilst the corresponding Pedi sentence “*ke a go rata” (I love you)* is constituted by the morphemes *ke, a, go, rat-, and -a.* The language models were developed for later engagement into speech recognition experiments.

1.2. Disjunctive and Conjunctive Languages

The Bantu languages are generally agglutinative, i.e., they use prefixes and suffixes in forming novel words from base words [5]. The words are formed by adding these affixes to the root morpheme, making the formed words inflections and derivations. South Africa’s four Nguni and three Sotho languages are spoken throughout the country. The official Nguni languages are Ndebele (Ndebele) or Southern Ndebele,

isiXhosa (Xhosa), isiZulu (Zulu), and siSwati (Swati); whereas the official Sotho languages include: Sepedi (Pedi) or Northern Sotho, Sesotho or Southern Sotho, and Setswana (Tswana).

The Nguni languages are generally conjunctively-written [6]. This means, their writing system concatenates several morphemes to make up word tokens [5]. Furthermore, the morphemes making up the linguistic words do not occur individually as words themselves. The Sotho languages on the other hand employ the disjunctive writing system. Here, a linguistic word may not be constituted by many morphemes, and the morphemes can be whole words on their own.

This study focuses on Ndebele and Pedi. Examples of sentences contrasting the conjunctive and disjunctive writing systems are shown in Table 1.1 and Table 1.2 shows the morphological analysis of one of the sentences.

Table 1.1: Conjunctive versus Disjunctive writing forms [5]

No.	Ndebele	Sepedi	English
1	*Uyakhamba.*	*O a sepela.*	*He/She is going.*
2	*Bayakhamba.*	*Ba a sepela.*	*They are going.*
3	*Uyathanda.*	*O a rata.*	*He/She loves.*
4	*Bayathanda.*	*Ba a rata.*	*They love.*
5	*Siyathandana.*	*Re a ratana.*	*We love each other.*
6	*Ube nekhambo eliphephileko.*	*O be le leeto leo le bolokegilego.*	*Have a safe journey.*

Table 1.2: Morphological analysis of Ndebele and Pedi text [7]

Language	**Sentence**	**Morphological analysis**				
English	*I like/love them*					
Ndebele	*Ngiyabathanda.*	*ngi-*	*-ya-*	*-ba-*	*-thand-*	*-a*
Sepedi	*Ke a ba rata.*	*ke*	*a*	*ba*	*rat-*	*-a*
Morphological class		s.c. 1p.sg	PRES	o.c. cl 2	verb, root	inflectional ending

The morphological classes from Table 1.2 have the following meanings: s.c 1p.sg (subject concord, first person singular), PRES (present tense) o.c cl 2 (object concord, class 2), verb (a word that indicates an action, event, or state)[2], root (basic form of a

[2] *http://www.yourdictionary.com/verb*

word, to which affixes are added)[3], inflectional ending (letter or group of letters added to the end of a word to change its meaning)[4].

1.3. Motivation

The ASR systems seek to accurately predict text corresponding to vocal input. More accurate ASR systems enable computers and other machines to freely interact with humans using natural language. To successfully develop an ASR system that determines the most likely sequence of the words spoken, it is necessary to build a language model.

Existing speech recognition literature carefully acknowledges that better and improved language models do not necessarily lead to better and optimal speech recognition systems [8]. However, there is also work that has seen good language models impacting positively on the speech recognition accuracy rates [9], [10], [11], [12], [13]; and consequently better and improved recognition systems. Generally, the performance of different language technology applications and tools gets improved with improved language models [9].

1.3.1. Aim

The aim of this study was to develop language models for speech recognition based on the disjunctive Pedi and conjunctive Ndebele languages.

1.3.2. Objectives

The objectives of this study were to:

i. determine the impact of conjunctive and disjunctive language orthography on ASR language modelling.
ii. evaluate standardized LM methods.
iii. compare language models' performance.

[3] *http://www.readingrockets.org/article/root-words-roots-and-affixes*
[4] *https://en.oxforddictionaries.com/grammar/grammar-a-z*

iv. contribute an in-depth analysis and study of LM work for the under-re-sourced South African Ndebele and Pedi languages.

1.4. Structure of the Dissertation

The rest of the the dissertation document is structured as follows: Chapter 2 explores the background and framework for ASR. Chapter 3 outlines the background and framework for LM. In the same chapter, previous studies that bear relation to our study are briefly explored.. The research design adopted for the experimental work is described in Chapter 4. Chapter 5 presents and analyses the results from the different experiments that were conducted. The results are further discussed in the same chapter. Chapter 6 summarises and gives conclusions to the conducted research work, and also hints at possible future work beyond this study.

Chapter 2: Automatic Speech Recognition Background

Automatic speech recognition as a human language technology is defined and its development process explored in the first two Sections 2.1 and 2.2. Section 2.3 defines the evaluation metric that could be used to assess the efficacy of a developed speech recogniser. The components of an ASR system as briefly introduced in Section 2.2 are further detailed in Section 2.4. The tools and toolkits that could be used for speech recognition development are mentioned in Section 2.5. Section 2.6 discusses the applications of the speech recognition technology as an enabling technology in various sectors of human technology.

2.1. Definition

ASR is the process of converting a speech signal to a text sequence of words by means of algorithms [14]. It takes a raw acoustic signal as input, and produces the corresponding hypothesised string of words as output [1]. It is also sometimes referred to as the speech-to-text (STT) process.

The goal of ASR and automatic language understanding (ALU) is for machines to be able to ‘hear’, ‘understand’, and ‘act upon’ spoken information [14]. An ASR partially addresses this goal computationally by building systems that map from an acoustic signal to a string of words, thus enabling the machine to ‘hear’ what was spoken. An ALU system then formulates an understanding associated with the produced string of words [1]. The completed intelligent system will thereafter act upon the formulated understanding.

By achieving adequate processing of spoken language, which is generalized as the spoken language understanding (SLU) problem, the historic ideal of enhancing human-machine interaction (HMI) by using natural language may be realized. Typically, any human being would be able to interact with and use a trained machine system (such as a computer, telephone or car) using spoken words and sentences through a ‘speak to’ and ‘listen to’ interface. This enables even the not-so-literate end-users to

enjoy the benefits and "wonders" of modern speech-enabled technological systems. Furthermore, many Pedi and Ndebele speakers will be able to use voice-enabled systems, thereby removing the necessity of having to know a foreign (and often previously colonial) language, such as English, before using modern technological systems. Many people will be empowered by a technology enabling human-machine interaction. This is likely to dominate almost every sphere of our modern lives as we aspire forthe reality of the Internet of Things (IoT)[5].

2.2. Automatic Speech Recognition Approach

Speech recognition takes as input an acoustic waveform and produces a probable string of words as output. The statistical-based approach to speech recognition which makes use of mathematical and statistical tools is explored. This approach starts with the collection of a corpus of transcribed speech recordings, then the computer system is trained to learn the correspondences between the recordings and their transcriptions and finally, at run time – statistical processes are applied to search through the space of possible sentences to select the most probable sentence matching the speech input.

Problem statement: "What is the most likely sentence *W* out of all sentences in the language *L*, given the observation sequence (i.e., acoustic input) *O*?" That is, find $\hat{W}$ such that:

$$\hat{W} = \text{argmax}_{w \in L} P(W|O) \quad (2.1),$$

with: $O = o_1, o_2, \ldots, o_k$ representing the observed sequence (i.e., acoustic observations such as feature vectors) and $W = w_1, w_2, \ldots, w_k$ representing the candidate word sequence. To make equation (8) operational, we employ the Bayes' rule, which states:

$$P(X|Y) = (P(Y|X) * P(X)) / P(Y) \quad (2.2).$$

Using equation (9), it then follows that equation (8) can be transformed to become:

$$\hat{W} = \text{argmax}_{w \in L} (P(O|W) * P(W)) / P(O) \quad (2.3).$$

The language model or prior probability, *P(W)*, models how likely a string of words *W* is to be a source sentence of the language *L*. The acoustic model (AM) or observation

[5] *https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#45da40091d09*

likelihood, *P(O|W),* expresses how likely the word sequence *W* can match the observation sequence *O*. The probability of the acoustic observation sequence O, *P(O),* expresses how likely the observation sequence could be a source sentence of the language *L*. From equation (10), the *P(O)* remains constant for each candidate sequence W and thus its omission will not influence the maximum likelihood estimation task of *W*. The simplified equation (11) then follows:

$$\hat{W} = \text{argmax}_{w \in L} P(W|O) = \text{argmax}_{w \in L} P(O|W) * P(W) \quad (2.4).$$

Given the AM and LM probabilities, i.e., *P(O|W)*s and *P(W)*s respectively, the probabilistic speech recognition model can be operationalized in a search algorithm, as shown on Figure 2.1, to compute the maximum probability sentence for the given acoustic waveform.

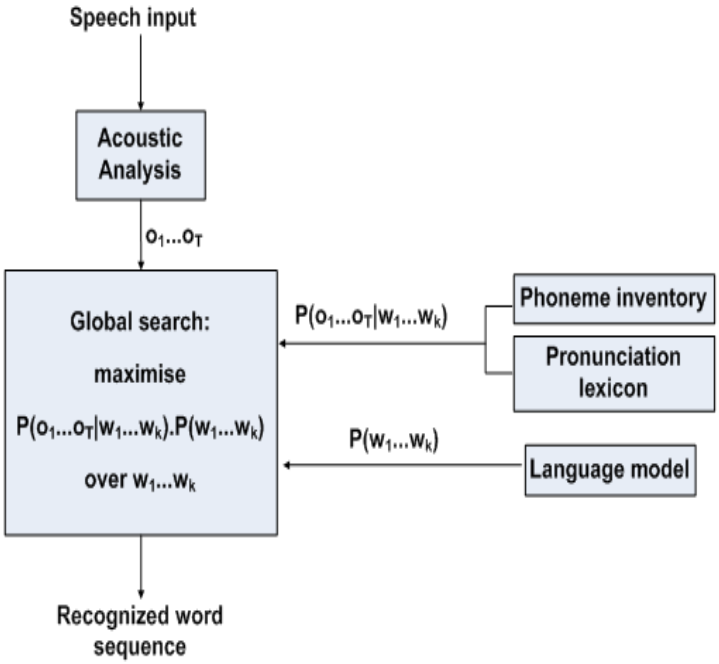

Figure 2.1: Observed sequence O processed by an HMM recogniser [15]

Figure 2.2 summarizes the main elements of a statistical approach to ASR through an HMM-based recogniser decoding a speech signal. The recognition stages are briefly described below [1]:

- In the *feature extraction* (or *signal processing*) phase, the acoustic waveform is sampled into a sequence of spectral features such as the Mel Frequency Cepstrum Coefficients (MFCCs).

- In the *acoustic modelling* (or *phone recognition*) phase, likelihoods of observed words, phones, or subparts of phones are computed.
- In the *LM* phase, prior probabilities of word sequences are computed to determine whether the sequences make valid sentences of the language. These word sequences are called n-grams in N-gram LM.
- In the *decoding or search* phase, the acoustic model (consisting of acoustic likelihoods), the lexicon or HMM dictionary (consisting of word or phone pronunciations), and the language model (consisting of n-gram prior probabilities) are all used to search and output the most likely sequence of words making up the recogniser's most likely hypothesised sentence.

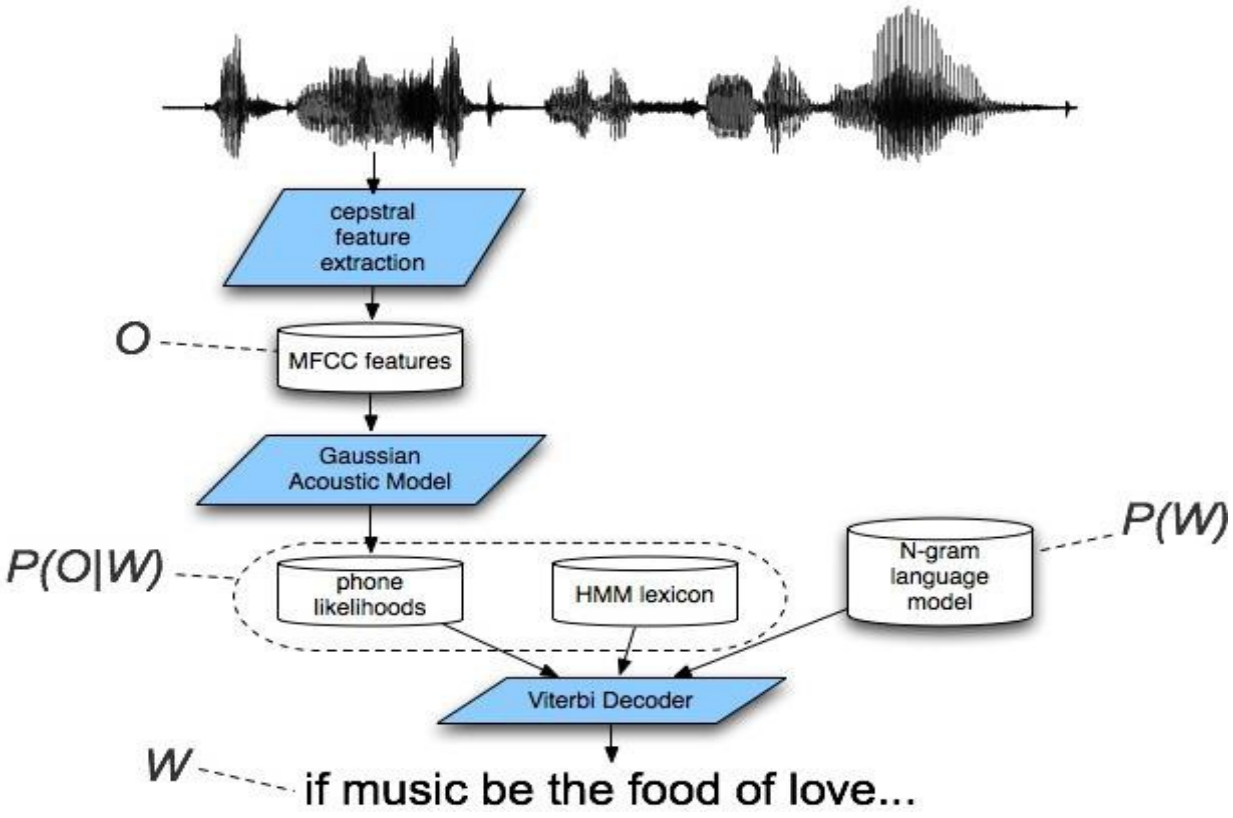

Figure 2.2: Statistical ASR [1]

2.3. Evaluation Metric

The standard evaluation metric for speech recognition systems is called the word error rate (WER): which is the rate or percentage of misrecognized words, and is based on how much the word string returned by the recogniser (called the *hypothesised word*

string) differs from its correct transcription (called the *reference transcription)*. Formally, the WER is defined by the following equation:

$$WER = ((D + I + S) / N) * 100 \quad (2.5),$$

where N is the number of words in the reference transcription; and D, I and S are respectively the deletion, insertion and substitution errors discernible when comparing or aligning the hypothesised transcription and the reference transcription. The lower the WER the more accurate the ASR system is in recognition.

2.4. ASR Components and Development Techniques

This section explores the different ASR components as introduced in Section 2.2.

2.4.1. Feature Extraction

At this stage of the ASR framework, the acoustic waveform (speech signal) is transformed into a sequence of discrete (or continuous) observations called the observation sequence, or input sequence. The symbols used to represent these observations are called feature vectors and contain acoustic information from the original signal. In a general setting, the feature vectors have a probability distribution associated with them [16].

An ASR system thus converts a speech signal into a symbolic description of the message produced in that signal during speech articulation [17]. Table 2.1 lists some of the various methods that can be used for feature extraction. One of the commonly used method, MFCCs approach, is detailed in Section 2.4.1.1.

Table 2.1: Feature Extraction techniques with their properties [15]

No.	Method	Property	Comments
1	Principal Component Analysis (PCA)	Nonlinear feature method; Linear map; Fast; Eigenvector-based.	Traditional eigenvector-based method; also known as Karhunen-Loeve expansion; Good for Gaussian data.
2	Linear Discriminate Analysis (LDA)	Nonlinear feature extraction method; Supervised linear map; Fast; Eigenvector-based.	Better than PCA for classification.
3	Independent Component Analysis (ICA)	Nonlinear feature extraction method; Linear map; Iterative non-Gaussian.	Used for de-mixing non-Gaussian distributed sources (features).
4	Linear Predictive Coding (LPC)	Static feature extraction method, 10 to 16 low order coefficients.	Used for feature extraction at lower order.
5	Cepstral Analysis	Static feature extraction method; Power spectrum.	Used to represent spectral envelope.
6	Mel-frequency Scale Analysis	Static feature extraction method; Spectral analysis.	Spectral analysis is done with a fixed resolution along a subjective frequency scale, i.e., Mel-frequency scale.
7	Filter bank analysis	Filters tuned required frequencies	
8	Mel-Frequency Cepstrum Coefficients (MFCCs)	Power is computed by performing Fourier Analysis.	Commonly used feature extraction method.

No.	Method	Property	Comments
9	Kernel based feature extraction method	Nonlinear transformation.	Dimensionality reduction leads to better classification and it is used to redundant features, and is improved in classification error.
10	Wavelet	Better resolution than Fourier Transform.	Replaces the fixed bandwidth of Fourier Transform with one proportional to frequency which allows better time resolution at high frequencies than Fourier Transform.
11	Dynamic feature extractions: (i) LPC (ii) MFCC	Acceleration and delta coefficients, i.e., II and III order derivatives of normal LPC and MFCC coefficients.	
12	Spectral subtraction	Robust Feature extraction method.	It is based on Spectrogram.
13	Cepstral mean subtraction	Robust Feature extraction method.	Same as MFCC but working on Mean statistical parameter.
14	RASTA filtering	Used for noisy speech.	Used to extract features in Noisy data.
15	Integrated Phoneme subspace	A transformation based on PCA + LDA + ICA.	Higher accuracy than existing methods.

2.4.1.1. The Mel Frequency Cepstrum Coefficients Feature Extraction Approach

One representation of the feature vectors extracted through a feature extraction process is the MFCC coefficients. The coefficients are extracted from a digitized and quantized speech signal. The feature extraction process follows seven steps [1], as shown by Figure 2.3.

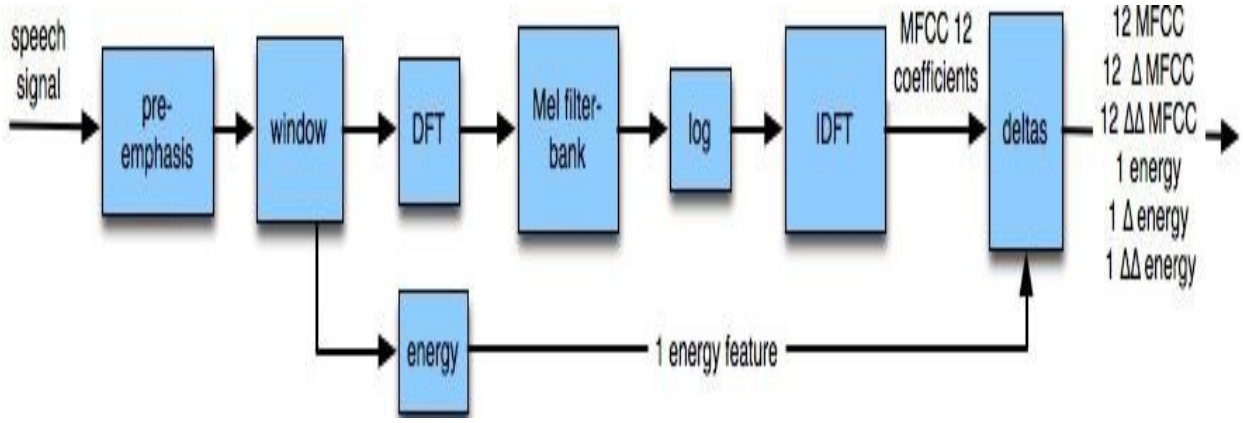

Figure 2.3: MFCCs feature extraction *[1]*

A. Pre-emphasis

Pre-emphasis is the first stage of MFCC feature extraction that boosts the amount of energy in the high frequencies of the signal. The spectrum of the speech signal has more energy in the lower frequencies and less energy in the higher frequencies. Boosting energy at these higher frequency levels makes information contained in them to be more available for acoustic modelling. This also improves phone detection accu- racy.

B. Windowing

A spectral feature is not extracted from the entire utterance signal, but from a small window of the signal that characterises a sub-phone or a phone – and for which we can assume that the signal is stationary (i.e., its statistical properties are constant). The windowed signal is called a frame.

C. Discrete Fourier Transform (DFT)

DFT is the process used to extract spectral information from the windowed speech signal. Information such as the amount of energy the windowed signal contains at different frequency bands. DFT gets its name from the fact that it extracts spectral information for discrete frequency bands for a discrete time.

D. Mel Filter Bank and Log

The DFT process results in information about the amount of energy at each frequency band. The Mel Filter Bank (a collection of filters) then collects energy from each frequency band of the signal. At the Log stage, a log (i.e., logarithmic computation) of each Mel spectrum value is computed. The log values make feature estimates to be less sensitive to variations in input – such as the speaker's mouth moving closer or further from the microphone during recording.

E. The Cepstrum: Inverse Discrete Fourier Transform (IDFT)

A *cepstrum* is a spectrum of the log of the Mel spectrum. After having the Mel spectrum, the log stage produced a log spectrum; IDFT then represents or visualizes the log spectrum as a waveform, wherein it is called a *cepstrum*. One visible distinction between a *spec*trum and a *ceps*trum (*ceps* reverse of *spec*) is that the former uses the frequency domain and the latter uses the time domain.

The MFCC extraction process then considers the first 12 cepstral values (the lower 12 cepstral values when detecting phones, and the first 12 higher values when detecting the pitch of the phones). A phone is the smallest unit of sound, while pitch refers to how high or low a sound is. These 12 cepstral values are called the 12 MFCC coefficients.

F. Deltas and Energy

The extraction of the cepstrum using IDFT results in 12 MFCC coefficients for each frame of the signal. A 13th feature, the energy feature from the frame - that relates with a phone - is added. For example, vowels and sibilants have more energy than stops in a language.

A speech signal is not constant from frame to frame. As such, features that relate to change are added to the cepstral features. To each of the 13 features, a delta (or

velocity) feature and a double delta (or acceleration) feature is added. Each of the added 13 delta features models the change between the frames in the corresponding energy feature, and each of the added 13 acceleration features models the change between the frames in the corresponding delta feature.

2.4.2. Acoustic Modelling

Acoustic modelling, also referred to as the *phone detection stage*, is the process of mapping acoustic features, derived from the feature extraction process, into distinct sub-word units such as phonemes, syllables, and words [18]. Acoustic models are developed for detecting the spoken phonemes in an utterance. Their creation involves the use of audio recordings of speech and corresponding text transcriptions, and then compiling them into a statistical representation of sounds which make up words [19]. Satisfactory performance with respect to acoustic modelling is achieved when the acoustic model is matched to a domain-specific task, and this is obtained through adequate domain-specific training data (corpora).

The hidden Markov models (HMMs), an important method for modelling time series data or sequences of observations [20], are the most popular acoustic models in use. Their popularity is attributed to two strong reasons [21]: (a) the models are very rich in mathematical structure and hence can form theoretical basis for use in a wide range of applications, and (b) when applied properly, the models work very well in practice for several important applications as seen successfully used in areas such as speech recognition, computational molecular biology, data compression, computer vision applications, and other areas of artificial intelligence and pattern recognition [20].

Some of the acoustic modelling techniques are explored next.

2.4.2.1. Adaptation Techniques

These are the techniques used to compensate for variation at the feature extraction level. One approach is estimating a linear (or nonlinear) transformation of the model parameters using a maximum likelihood (MLE) criterion, or a maximum posterior function [22]. Another approach is the Eigenvoice approach – which builds a low dimension eigenspace in which any speaker is located and modelled as a linear combination of ‘eigenvoices’.

With adaptation techniques, knowledge about the effect of the interspeaker variabilities is gathered in the model to re-estimate optimal model parameters for given circumstances – hence adaptation, while such speech variation information is discarded in other traditional approaches.

2.4.2.2. Multiple Modelling Techniques

Rather than adapting the models to certain conditions, the multiple modelling technique trains a collection of models specialized to specific conditions or variability. In such an environment where the speech recognition system should handle various conditions, several speech corpora can be used together for estimating the acoustic models, leading to mixed models or hybrid systems.

2.4.2.3. Auxiliary Acoustic Features Techniques

Speech recognition systems rely on the acoustic parameters that represent the speech signal, such as the (cepstral) coefficients. These features, however, are sensitive to auxiliary information inherent in the speech signal such as pitch, energy, rate-of-speech, formants, and so on. The auxiliary acoustic feature approach considers this auxiliary information in its modelling process. Here, the auxiliary features are directly introduced in the feature vector, along with the (cepstral) coefficients.

For example; formants were used together with MFCC coefficients in [23], the pitch parameter was included in the feature vector in [24], and both the pitch and energy features were used in [25].

2.4.3. Language Modelling

Statistical and grammar-based language models can be developed. On one hand, a statistical language model is basically a file containing probabilities of sequences of words [26]. It expresses how likely a sequence of words is to be an acceptable/legal sentence of a language. On the other hand, two constructs are important for the grammar-based language model: the *grammar* and the *parsing algorithm* [27]. The grammar is the formal specification of the permissible structures for the language, usually based on expert knowledge; and the parsing algorithm is the method of analysing a

sentence to see if its structure is compliant with the grammar. In ASR, statistical models are conventional due to their simplicity, modelling performance, and applicability to even large domain tasks [28].

The LM process is specially detailed in Chapter 3.

2.4.4. Decoding

The decoding process searches for the most likely word sequence corresponding to the observed acoustic data [29] using inputs from the various acoustic, lexical, and language models. This process finds the best word sequence that might match the input speech signal. The techniques used by speech recognition engines to match a detected word sequence to a known word sequence include the following:

2.4.4.1. Whole-word Matching

Here, the recognition engine compares an incoming digital-audio signal with a pre-recorded template of a word. This technique requires that there be a pre-recorded template for every word that is to be recognised. Each word template occupies storage amounts between 50 and 512 bytes. Whole-word matching is practical only if a small recognition vocabulary is known in advance.

2.4.4.2. Sub-word Matching

In this decoding technique, the engine looks for sub-words, usually phonemes, and then performs further pattern recognition on them. Each sub-word word requires 5 to 20 bytes in storage. This technique takes more processing time than whole-word matching.

2.5. ASR Tools

A wide range of tools are readily and freely available from designated internet portals for use to develop a working speech recognizer, and some of these tools and toolkits that could be used are [30] [31] [8] [32] [33]:

- HTK: a toolkit used to build and manipulate HMMs.
- SPHINX: an open-source large vocabulary speech recognition toolkit, based on C, C++, and Java programming languages.

- PRAAT: a free software tool for recording and analysing human speech, which can run on different platforms such as UNIX, Windows, and Macintosh.
- JULIUS: a two-pass, real-time, and open-source large vocabulary continuous speech recognition engine.
- KALDI: an open-source speech recognition toolkit that is based on finite-state transducers and the C++ programming language.
- Microphone: a tool used to record speech data.

2.6. Applications Areas of ASR

Since its introduction, the speech recognition technology has been applied in a wide variety of areas [1] [34], including but not limited to:

- Human-Computer Interaction – enabled ability for humans to interact with the machine systems (such as phone, car, escalator, and robotic systems) using voice recognition interfaces.
- Transcribing systems - that transcribe recorded speech to its textual form.
- Telephony – incorporating telephone systems with intelligent speech recognition technology, adding capabilities such as:
    - Entering/dialling digits using speech
    - Call routing/forwarding/steering – putting callers through to the right department or unit of an organisation.
    - Booking or enquiring about airplane or train information.
    - Online help services
    - Automated caller identification – where there is a need to authenticate someone's identity on the phone without using risky personal data.
    - Removing and/or enhancing the interactive voice response (IVR) menus – replacing the complicated and often frustrating 'push button' IVR using intelligent call steering (ICS) where the IVR system simply asks the customer to say what they want (to which they respond in their words) and then transfers them to the most suitable resource to handle their call; or add the speech recognition capability on the already available IVR keypad menu.
- Dictation – speaking to a writing system that writes down what is spoken.

- Eyes-busy and hands-busy applications (for use in automobiles, for medical doctors, etc.)
- Speaker Identification – identifying the person who is speaking by characteristics of their voice.
- Language identification – identifying the language of the speaker from their utterance.
- Voice search – searching files on the internet using voice or speech.
- Language advancement – technological systems such as ASR recognisers help advance the identity and usage of a language.
- Language learning – helping speakers to learn and understand transcriptions for utterances of a language.

The many application areas for speech recognition signify its importance in modern day technology since its advent in the early 1930s. As such, the speech recognition technology has become an established community with its own language and terminology, and this way it continues to grow day by day.

Chapter 3: Language Modelling Framework

This chapter gives the background to LM as an aspect of automatic speech recognition. Sections 3.1 and 3.2 define the concept of LM in relation to speech recognition; and the LM process, implemented by different techniques, is explored in Section 3.3. Section 3.4 describes the measures used to evaluate the accuracy of developed language models. LM challenges leading to LM research are briefly explored in Section 3.5. The classification of LM development is characterised in Section 3.6. Section 3.7 and 3.8 gives a mention of some the tools and toolkits used for LM development. Lastly, Section 3.9 surveys some of the existing research works bearing similarity to this study.

3.1. Language Modelling Definition

LM as the art or process of determining the probability of a sequence of words [35], is one of the core processes in NLP systems that deals with text such as: speech recognition, machine translation, optical character recognition, handwriting recognition, and spelling recognition systems. For most of these NLP technologies, statistical language models have become state-of-the-art models.

In ASR systems, statistical n-gram models are conventional due to their simplicity and modelling performance [28]. An n-gram is a sequence of n elements (e.g. words) and an n-gram language model is used to estimate the chances of any element occurring in the sequence given its n-1 predecessors.

3.2. Role of Language Modelling in Automatic Speech Recognition

Statistically, the ASR LM task is to determine the prior probability P(W) that the speaker would utter the word sequence W = w_1, w_2,…, w_n in the recognition problem

$$\hat{W} = argmax_W P(W|X) = argmax_W P(X|W)P(W) \quad (3.1),$$

where X is the observed speech signal and W any valid sequence of words from the prescribed vocabulary [36] [37]. The language model P(W) of the word sequence W, together with the acoustic model P(X|W) of the acoustic signal X, are used in speech recognition when searching for the best word sequence $\hat{W}$ that matches X. In this relation, the LM component complements the acoustic models with prior information

about word sequences occurring in a language or language task. The prior information helps the speech recogniser to be able to choose between word and sentence hypotheses for which there is evidence in the acoustic data. The correctness of a sentence hypothesis is backed by evidence of more likely words in the hypothesis.

The crucial role that is played by the language model can be summarized into: (a) constraining acoustic analysis, guiding the search through multiple text hypotheses, and contributing to the final transcription; and (b) encapsulating as much syntactic, semantic, and pragmatic characteristics of the language task as possible [2]. In terms of (b), the successful capture of most of this information is important to help the recogniser determine the most likely sequence of words spoken as the information quantifies which word sequences are valid and which are not in terms of the language task.

3.3. Language Modelling Approaches

The general LM process for speech recognition is presented by Figure 3.1. The major steps involved include data preparation, language model training, and then language model testing. The data preparation step prepares the text data into a format suitable for LM training, such as having the text separated into one sentence per line and each sentence marked with sentence boundary markers <s> (start-of-sentence symbol) and </s> (end-of-sentence symbol). The LM training step estimates the statistical LM distribution from the training text using various techniques such as n-gram modelling, smoothing, and back-off. The LM testing step then evaluates the trained language model on the testing and/or development data and outputs its *perplexity* (PPL) score.

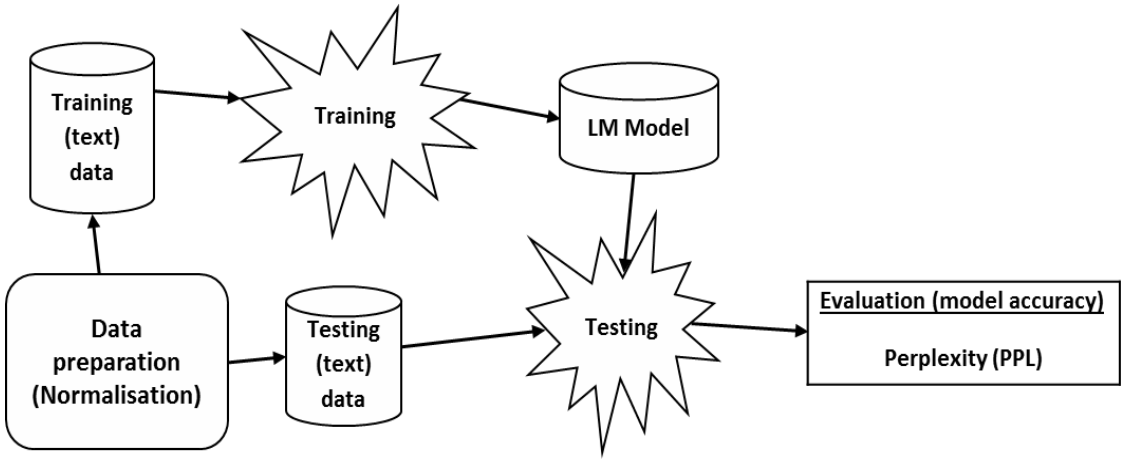

Figure 3.1: General LM framework

The general step-by-step procedure as described through Figure 3.1 gets adapted and further detailed by different LM frameworks, especially on the training phase. Sec- tions 3.3.1 through 3.3.8 discuss this point for various LM techniques.

3.3.1. Conventional Statistical N-gram Language Modelling

An n-gram language model is used to predict each symbol in the n-gram sequence, given its n-1 predecessors. This prediction assumes that the probability of a specific n-gram occurring in some unknown test text can be estimated from the frequency of its count/occurrence in the training text.

The n-gram LM construction process is three-phased, as shown by Figure 3.2. In the first phase, the training text is scanned and its n-grams are counted and stored in a database of gram files. In the second phase, some class mapping may be applied and some of the words be mapped to classes such as the out-of-vocabulary (OOV) class. Then in the final phase, the counts from the resulting n-gram files are used to compute n-gram probabilities which are stored in a language model file.

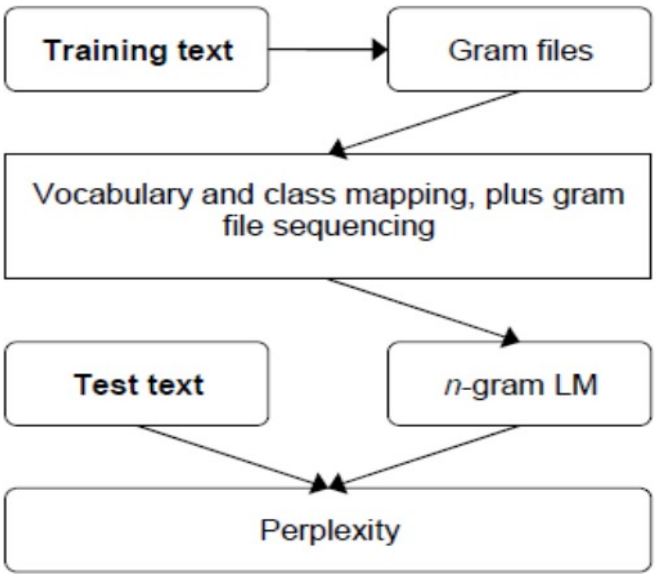

Figure 3.2: N-gram LM framework [8]

Ultimately, the effectiveness of the resulting language model is determined using the perplexity measure on some unseen test text data set. In general, a better language model has a lower test text perplexity.

3.3.2. Neural Network Language Modelling

Neural network language modelling (NNLM) embeds words in a continuous space where LM probability estimation is performed using single hidden layer trained neural networks. NNLM with multiple hidden layers is called Deep Neural Network LM (DNN LM). A schematic representation of a single hidden layer NNLM framework is shown in Figure 3.3.

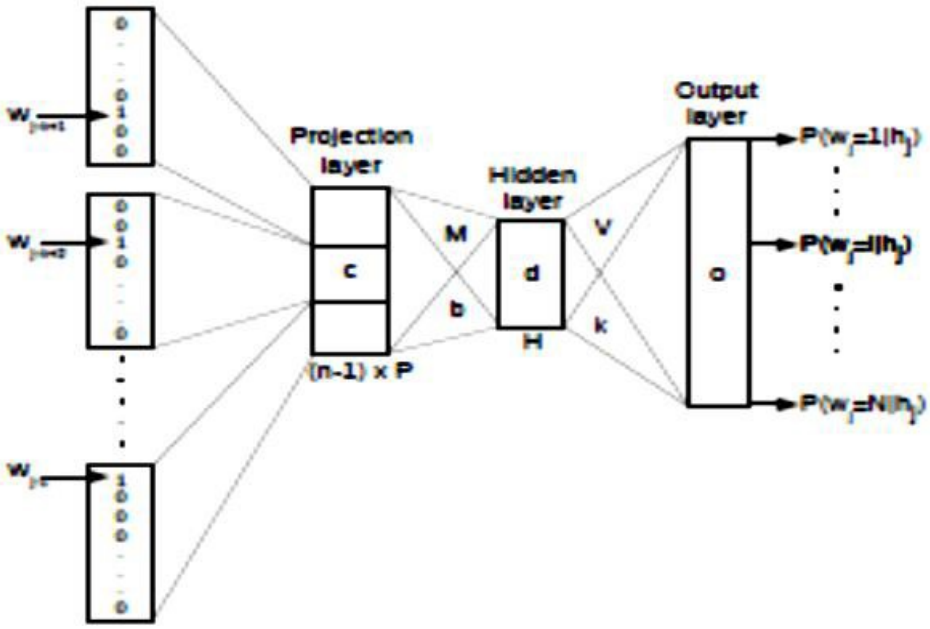

Figure 3.3: NNLM framework [28]

Each word in the vocabulary is represented by an N dimensional sparse vector, in which only the index of that word is 1 and the rest of the entries 0. Input to the network is the concatenated continuous feature representations of previous n-1 words, i.e. the history/context words. Each word is mapped to its continuous space representation using linear projections. Discrete to continuous space mapping is a look-up table with N x P feature dimension. The i^{th} row of the table corresponds to the continuous space representation of the i^{th} word in the vocabulary.

The continuous space feature representations of the history words are combined to form the projection layer. The hidden layer has H hidden units and it is followed by a

hyperbolic tangent non-linearity. The output layer has N targets followed by the soft-max function. Its posterior probabilities, $P(w_i=i|h_j)$, are the LM probabilities of each word in the vocabulary for a specific history h_j.

From the architecture in Figure 3.3, c represents linear activations in the projection layer, while M and V are the weight matrices between the projection and hidden layers and hidden and output layers, respectively. d, o, and p are the allowable operations in the framework and they are defined as follows:

$$\mathbf{d} = \tanh(\mathbf{M} * \mathbf{c} + \mathbf{b}) \quad (3.2),$$
$$\mathbf{o} = \mathbf{V} * \mathbf{d} + \mathbf{k}$$
$$\mathbf{p} = \exp(\mathbf{o}) / \sum_{l=1}^{N} e^{o_l}$$

where b_j and k_i are the hidden and output layer biases [37].

The algorithm used in training the models is the standard stochastic back-propagation algorithm. Basically, a neural network language model performs two tasks: first, it projects all words of the context ($h_j = w_{j-n+1}^{j-1}$) onto a continuous space; and second, it calculates the language model probability $P(w_j=i/h_j)$ for the given context. These two tasks are performed by the NNLM model using two layers, the projection layer and the hidden layer. That is, the projection layer is responsible for the continuous representation of all words in the context and the hidden layer does non-linear probability estimation. The network is trained by stochastic back-propagation to minimize perplexity of the training data.

Set N to be the size of the vocabulary and P the dimension of the continuous space. Input to the network are the N dimensional binary vectors of the previous (n-1) words in/from the vocabulary. The vectors are created using the 1-of-n coding mechanism, where the i^{th} word of the vocabulary is coded by setting the i^{th} element of the vector to 1 and all other elements to 0. These continuous projections/representations of all words in the context are concatenated to form the projection layer. The projection layer uses a linear activation function. The activities taking place at this layer could be denoted as c_i with i=1, 2, ..., (n-1) x P.

3.3.3. Neural Networks: an emerging modern standard for language modelling

Neural network language models (NNLMs) as introduced in the preceding section (Section 3.3.2) are becoming state of the art for LM as they do for other modelling tasks such as ASR and machine translation [38] [39]. NNLMs' prominence over n-gram models is due to, amongst other factors: ability to model long term history contexts, implicit parameter sharing in a continuous space of projected words, and efficient interpolation or merging with other effective language models such as n-grams. Another key factor with NNLMs, however, is that they are processing power demanding, thus successful development and application thereof relies on computation intensive hardware and techniques such as multi-core processors, graphical processing units (GPUs), multi-threading and parallel computation.

The widely developed NNLMs are feedforward and recurrent NNLMs [39]. A survey of the use of NNLMs in LM and ASR was conducted by Kipyatkova et al. [39] and includes the following highlighted studies. Initial work on NNLM in 2005 by Schwenk et al. [40] which compared an NNLM model with a Kneser-Ney smoothed trigram model. Recurrent NNLMs (RNNLMs) were first used by Mikolov et al. [41] for rescoring n-best lists of ASR hypotheses generated through a Kneser-Ney smoothed pentagram to obtain the best (1-best) hypothesis of the input speech. Sundermeyer et al. [42] compared feedforward and recurrent NNLMs that were separately interpolated with an n-gram model. Speech recognition results indicated RNNLMs outperforming feedforward NNLMs. In general, the following common traits are seen from the surveyed NNLM studies: use of NNLMs to rescore n-best output generated through use of n-gram LMs; interpolating NNLMs with n-gram LMs; improved language model and speech recognition results; and the prevalence of recurrent NNLMs over feedforward NNLMs.

3.3.4. Syntactico-Statistical N-gram Language Modelling

The syntactico-statistical LM approach exploits both the syntactic and statistical text analyses for LM. One such process is illustrated in Figure 3.4.

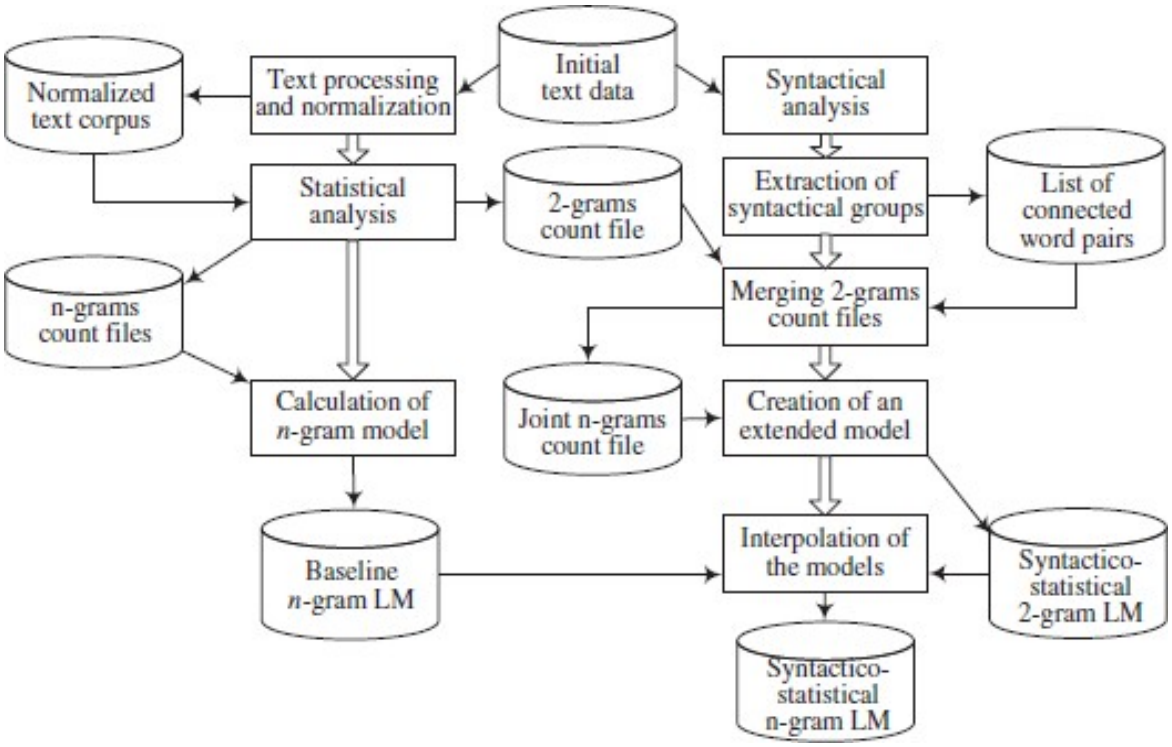

Figure 3.4: Syntactico-statistical n-gram LM framework/flowchart [11]

Initially, the training text corpus is processed in parallel to discover regular n-grams and syntactic word dependencies in sentences. The results of both analyses are then processed to obtain count files, after which statistical and syntactic n-gram models are computed. At the final stage, the two types of language models are interpolated to create a model of the required n-gram order. Thus, the eventual n-gram language model is the result of joint statistical and syntactic text analysis (where the latter learns long-distance/grammatical dependencies between non-adjacent words, and the former examines such relations between adjacent words) at the training stage.

3.3.5. Syntactic Language Modelling with Formal Grammars

The syntactic LM approach described by Kaufmann and Pfister [43] follows the two-stage decoding ASR paradigm, also known as lattice rescoring recognition, in which LM is carried out in two stages. In this paradigm, formal grammar models are integrated in the speech recogniser's decoding stage.

A formal grammar intends to discriminate between grammatical word sequences and ungrammatical word sequences. Grammatical word sequences are those sequences

that can be generated by a sequence of rule applications. The formal grammars that are used additionally make the following assumptions. (a) The grammar allows the determination of possible syntactic structures of a grammatical sentence. With these structures, statistical models can compute probabilities of derivations or word sequences. (b) The grammaticality and syntactic structures can be determined for linguistically motivated units other than sentences, e.g. noun phrases. This allows extraction of linguistic information even in the face of ungrammatical sentences.

The combination of the described formal grammars with a statistical model is different from a statistical parsing model. A typical statistical parser is completely guided by a statistical model and allows for any (word sequence) derivation that is structurally possible. A grammar-based parser, on the other hand, is restricted by the hard-linguistic constraints encoded in the grammar – and thus the statistical model of the parser complements the grammar with quantitative information.

Formal grammars have been used as language models since the beginning of time for speech recognition. However, they have almost exclusively been applied to re- stricted domains because, amongst other reasons, small domains (such as money value description, digits, and vowel recognition tasks) allow for very restrictive gram- mars that constrain both the syntax and semantics of the acceptable utterances. Large vocabulary speech recognition domains (such as media news, parliamentary debates, and university lectures transcription tasks) are difficult for formal grammars because for one the syntax for such domain is very productive, and thus many incorrect hypotheses should be considered grammatical.

The architecture for the described syntactic LM approach is shown in Figure 3.5.

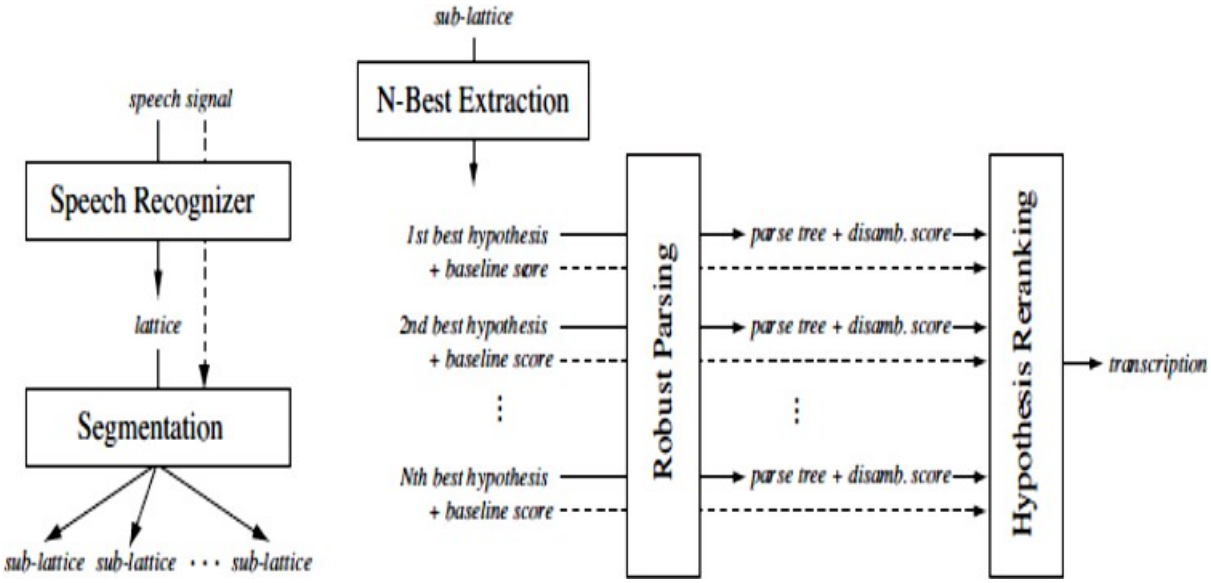

Figure 3.5: Syntactic LM (with formal grammars) architecture [43]

In the first stage, the speech signal is processed by a baseline speech recogniser (having a baseline n-gram model) and the resulting word lattice is automatically segmented into sub-lattices that represent sentence-like units. In the second stage, for each sub-lattice, N best hypotheses are extracted with respect to the baseline recogniser score. The baseline recogniser score is the weighted sum of a score and a word insertion penalty.

For each (sub-lattice) hypothesis, a parser determines a unique parse tree and its associated disambiguation score. The disambiguation score represents the plausibility (validness or likeliness) of the parse tree: highly plausible trees receive large negative scores.

Finally, a discriminative re-ranking component chooses the most likely hypothesis for each sub-lattice. Various features taken into consideration during the re-ranking process include the baseline recogniser score, the disambiguation score and different properties of the parse tree. Together, these features are used to compute the final score for each hypothesis, and then the hypothesis with the maximum score gets chosen as the recogniser's transcription result.

3.3.6. Approximate Language Modelling Inference

The approximate LM approach approximates long span and complex language models (such as neural or recurrent language models) using simple n-gram models [44]. This is done using variational inference, the widely used method for approximate inference. The approximated language model is then used for first pass decoding and the resulting lattices are then rescored with a bigger full-blown model, such as the one being approximated. The pictorial representation of this concept of distributions is shown in Figure 3.6.

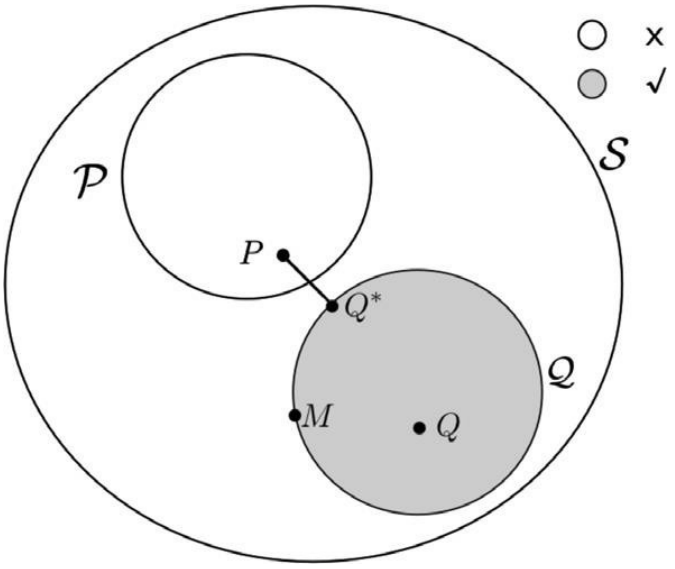

Figure 3.6: Pictorial representation of family of distributions in variational inference [44]

Given a complex long-span model P, we seek a computational tractable model Q* that will be a good surrogate for P. In particular, among all models Q of the family $\mathcal{Q}$ of tractable models, we seek one that minimizes the Kullback-Leibler divergence (KLD) from P. When found, the model is used for first pass decoding and richer lattices and/or faithful N-best lists are produced as a result. Then, full blown (i.e. non-approximated) bigger language models are used to rescore the lattices or they are deployed on the extracted N-best lists.

In variational (approximate) inference, a surrogate model (characterized by the distribution $Q \in \mathcal{Q}$) is selected to replace a complex model (characterized by the distribution P) such that inference under Q becomes more tractable. Q is found such that it is

closest to P in some way. Given that the models are probability distributions, a natural choice of distance metric is Kullback-Leibler divergence. Thus, the surrogate model Q is chosen such that among all the distributions in the family of the parameterization distribution Q, it has the minimum KLD with the complex distribution P.

3.3.7. Factored Language Modelling

Factored LM (FLM) approach considers various sources of information and combines the information in a manner that produces an efficient statistical language model [9]. The FLM approach makes it possible to build a statistical model over heterogeneous/numerous factors assigned to (or incorporated in) each input word.

Within the FLM framework, as described by Falavigna and Gretter [10] , and Karpov *et al.* [11], each word (w_i, $1 \leq i \leq N$) is regarded as a bunch of (k) factors, i.e., $w_i = f_i^1 f_i^2 \ldots f_i^k$. Factors are features of the word such as: the class of the word, the word itself, part-of-speech (POS) of the word, possible lemmas of the word, syntactic and semantic factors, and distinct factors corresponding to the different morphemes of the word. The choice of factors depends on the available information and the researcher's ideas to better language models. The chosen factors eventually become the features of the resulting model.

In summary, the FLM approach to LM makes it possible to build a statistical model over different/heterogeneous factors inherent in a word. Each word is thus regarded as a bunch of factors such as the word itself and its inherent morphemes that include the stem, POS, lemma, morphological tag and so on. The chosen factors, as per the will of the designer/developer/researcher, become features of the resulting language model.

3.3.8. Statistical Language Model Adaptation

When the discourse in training and recognition tasks differs in terms of lexical, syntactic, or semantic characteristics; language model adaptation becomes necessary to compensate for the mismatch as it severely affects the performance of statistical language models [2]. Generally, an adaptive language model seeks to maintain an adequate representation of the current task domain under changing (language) conditions such as variations in vocabulary, syntax, content, and style. This helps reduce the

degradation in speech recognition performance usually observed with a new set of operating conditions.

Figure 3.7 depicts the general statistical language model (SLM) adaptation framework. The framework considers two text corpora in training the adapted model: a usually small adaptation corpus A, that is associated to the recognition task; and a usually large background corpus B, associated with a somewhat different task.

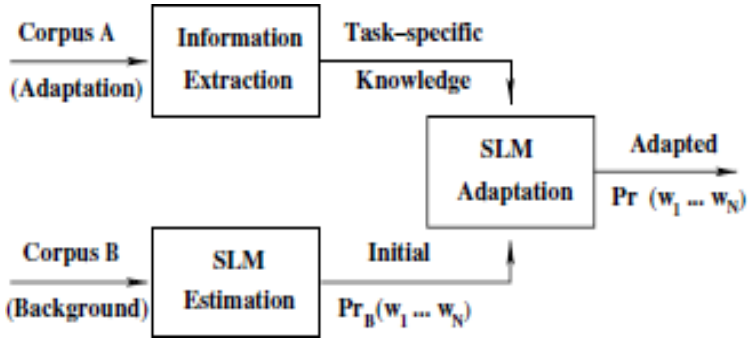

Figure 3.7: SLM adaptation framework [2]

For the adaptation problem, given a sequence of *n* words w_q ($1 \le q \le N$) consistent with the corpus A, the goal is to determine a robust estimate of the LM probability

$$P(w_1,\ldots,w_N) = \prod_{q=1}^{N} P(w_q|h_q) \quad (3.3),$$

where h_q is the history (composed of previous words) available at time q. The Markovian assumption implies that $h_q = w_{q-n+1},\ldots, w_{q-1}$.

Estimation of $P(w_1,\ldots,w_N)$ exploits two distinct sources of knowledge: (i) the well-trained, possibly mismatched, background statistical language model that yields $P_B(w_1,..,w_N)$ as shown on Figure 3.7, and (ii) the adaptation data that is used to extract some information relevant to the current task we are adapting to. The general idea is to dynamically modify the background SLM estimate based on the information extracted from the adaptation corpus A. All SLM probabilities are assumed to be appropriately smoothed.

The adaptation method depends much on the quality of the available adaptation data (i.e., corpus A). If not already available, ways of gathering corpus A include: (i) when

the recognition task is recovered by a grammar - using the grammar to generate expected user utterances to make up an artificial version of corpus A; and (ii) accumulating the data during the recognition process from N-best lists of (multiple) sentence hypotheses. Adaptation data may also be from repositories such as online databases or the Internet.

3.4. Evaluation Metric

The most commonly used metrics for evaluating language models are: the probability assigned to the test data by the language model, cross-entropy, and perplexity measures [13]. Given the language model that assigns probabilities $P(w_i | w_{i-n+1}^{i-1})$, the probability of a sentence P(s) is calculated using the equation:

$$P(s) = P(W) = \prod_{i=1}^{l+1} P(w_i | w_{i-n+1}^{i-1}) \quad (3.4),$$

where *s* is the sentence or word sequence $W=w_1w_2\ldots w_l$ of length l. Then, for a test data set T composed of sentences $(s_1, s_2, \ldots, s_T)$, the probability of the test set P(T) is derived from the product of the probabilities in the set, i.e.,

$$P(T) = \prod_{i=1}^{l_T} P(s_i) \quad (3.5).$$

The *cross-entropy* $H_p(T)$ for the language model $P(w_i | w_{i-n+1}^{i-1})$ on the test data T, is defined as:

$$H_p(T) = -\frac{1}{W_T} \log_2 P(T) \quad (3.6),$$

where W_T is the length of T in terms of words.

The *perplexity measure PPL* on the test data is defined by its relation with the cross entropy as:

$$PPL = 2^{H_p(T)} \quad (3.7).$$

The quality of the language model in terms of modelling the test data is better if it assigns the highest test data probability P(T), the lowest cross-entropy $H_p(T)$, and the lowest perplexity PPL. The PPL metric defines, within it, relations to both P(T) and $H_p(T)$ and is thus used to measure the performance of the language models in this study. Therefore, using equations (5), (6), and (7), one can convert from a test data

set’s PPL value to its cross-entropy and/or probability, and vice-versa.

As an approximate rule of thumb, it is suggested by Rosenfeld [45] that PPL reductions of 5% may be said to be practically insignificant to the application domain, noteworthy if it is a reduction of about 10-20%, and quite significant (and rare) when the reduction is of 30% or more. These significance or insignificance levels were mentioned in relation to the speech recognition application domain.

In multi-stage recognition systems, lattice rescoring approach is used to evaluate the output of language models. The lattice is made of the several transcription hypotheses output generated through the baseline language model (with relevant scores), and is rescored by another language model in the next stage of recognition.

Other measures can be used to report on other valuable information about the quality and performance of the language models, over and above those mentioned here. For example, n-gram hit rates are also reported on by Karpov *et al.* [11].

3.5. Language Modelling Challenges

LM challenges, some unique to a language, are often the central focus of LM research. This section attempts to survey some of these challenges as they have manifested themselves in studies of LM and ASR.

At the forefront of the many challenges, there is the need to advance both ASR and LM research, especially for under-developed languages.

The nature and regularities of a language - such as rich morphology, high inflexion, flexible word order, and compounding of words - present unique LM challenges that need to be addressed specifically for that language. The language structures of different languages belonging to different language families often warrant different LM approaches and features. Furthermore, these complex language structures lead to large vocabulary sizes and OOV rates. Large word vocabularies themselves demand added knowledge about words. As examples: Russian is a morphologically rich inflective language that creates words by using affixes on stems and by inflecting for various syntactic features such as case, number, gender, etc. [46]; Khmer is written without

spaces between words [47]; French and Italian are highly inflective; and German compounds its content words [36].

The under-development of languages, as defined in terms of NLP resources scarcity, is the major LM challenge. Resources necessary for developing human language technologies - such as text and speech corpora, pronunciation dictionaries, monolingual and multilingual electronic dictionaries, etc. - are lacking for the under-resourced languages. Most of these languages are particularly African, South Asian, and Eastern European in origin [11].

There is a need to develop improved ASR systems with improved and/or robust models for convenient and efficient HMI to be realisable. The improvement is towards continuous, speaker-independent and spontaneous ASR systems with increased speed, robustness, vocabulary, and usefulness for the end-user [11].

N-gram modelling on word dependencies alone fails to represent the sub-word unit dependencies that may not be present at word level such as the relatedness of words sharing the same stem [48].

The standard n-grams (e.g. trigrams) modelling only on adjacent words result in low n-gram coverage, since this is a minimal observation of the information contained by the training data, and thus lead to high misrecognized valid n-grams. The poor word n-gram coverage is also realized when some sub-word language models are developed [47].

The standard n-grams are also fallible to failing to capture complex syntactic, semantic and/or discourse information inherent in text sentences during LM. The relationships or dependencies between non-adjacent words in the sentence are not reliably estimated, if any, by the n-grams' use of the independence Markov chain assumption of order n-1 [48]. Thus, the n-grams estimate with information or knowledge that is less representative of the language or language task.

Generalisation to unseen n-grams is difficult for language models estimating from a discrete space, such as standard n-grams using the word vocabulary, because a change of a word causes arbitrary changes in the n-gram probability [28].

The complex language models (such as approximated language models) that advance on the standard n-grams - increase the size of the (sentence hypotheses) search space, are computationally intensive, require added memory, and take time to train [44]. Even more demanding are the combinations of such models. At times, good language models may work quite well when used separately but diminish in performance when combined [35].

The data sparsity prevents the use of the full word history to estimate the word, and thus leads to poor estimates of LM probabilities. Across NLP domains, data sparsity (also known as data scarcity or data paucity) refers to the phenomenon of not observing enough data to accurately model a language [49]. This makes it difficult to determine the (true) distribution and pattern of a language as aspired by language models. Generally, sufficiently large in-domain data is often lacking to accurately model most language tasks. One of the unpleasant consequences of data sparsity is when there is large irrelevant training corpora leading to extremely small or zero probabilities being assigned to many valid word sequences.

On the other hand, working or training with large data has implications of requiring increased amounts of training time and memory consumption [50]. Such models are sometimes too large for real applications to implement or use.

The demand by statistical models for large in-domain training data for any language task, even the simplest, calls for universal language models that are universal and robust enough to be adapted to changing purposes [36].

There is no finite vocabulary, however chosen, that can fully cover a speaker's need [36]. Speakers are fond to using routine words (such as names of friends, technical terms, etc.), code-switching speech, or a way of speaking depending on who they talk to.

The PPL measure for evaluating language models is not directly related to the recogniser's evaluation measure, namely, the WER. A better performing language model does not guarantee a better performing speech recogniser, since a lower PPL language model may not result in lower WER speech recognition performance. In part, the cause of such a relation may be because the PPL metric does not consider the acoustic similarity between words [36].

The dependence of statistical models on relative frequency estimates is not sufficient for the determination of the likeliness of an n-gram or word sequence. Is a trigram seen *k* times really *k* times more probable than one never seen? Or, is the appearance of a singleton (i.e., n-gram occurring just once in the training data) only a lucky coincidence signifying little likelihood [36]? Many perfect valid word sequences may not appear even in very large corpora [49].

Another LM challenge is the inability to use linguistic grammars - constructed based on expert knowledge - for large domain language tasks. Grammars are thus far used for small restricted domains [43], and thus the grammatical knowledge uniquely mod- elled by the expert-based grammars is lost for broad domains.

3.6. Characterisation and Classification of Language Modelling

The units of LM include whole words, sub-word units such as morphemes (e.g. stems, roots, affixes, etc.), and could extend to even whole sentences. The use of whole words has become standard and ideal for traditional state-of-the-art speech recogni- tion LM [47]. Unique structures of a language, however, may prompt and necessitate deviation from convention to achieve improved modelling for the language task at hand.

The use of a finite vocabulary by an ASR system means that all the words not listed in the vocabulary get classified as OOV and thus have high chances of being misrecognised [51]. The vocabulary mainly lists the units (words and/or sub-words) of LM. It thus follows that when the vocabulary entries are words – LM is word-based, and sub-word LM is implemented when sub-word units are listed by the vocabulary. On the hand, sentence-based LM will require the vocabulary to list individual sentences. As a

result, the complexity of the LM process grows as the LM unit gets bigger. It grows in terms of the required LM vocabulary size that would best represent the language task. An increasing vocabulary increases system complexity especially for morphologically rich languages. It also demands increase in the amount of training data required to train the language models. However, a bigger LM unit has benefits such as good word coverage though with less representative vocabulary [47].

Morpheme or sub-word based modelling has therefore become a promising direction that achieves better language models than those based on whole words and sentences for other languages, especially languages rich in morphemic structure such as inflective, derivative, and compounding languages [9]. This type of modelling yields morpheme or sub-word unit sequences as the recognised output and thus after recognition, the sequences are reconstructed to form whole words hypothesised sentences.

The use of LM units other than whole-words requires that the vocabulary and the corpora be reconstructed to reflect such units. Morphological decomposers (e.g. Morfessors [52]) and text segmenters (e.g. Finite state machine (FSM)-based segmenter [51]) may be used to transform the word corpora into one that is morphemic for exam- ple.

The modelling technique employed in the LM process further classifies the LM approach. The usage of neural networks leads to a neural network LM approach [28]. The factored LM is based on modelling over a set of morphemes grouped into factors [53]. A decision-tree-based approach uses decision trees such as random forests in modelling word sequences [9]. The syntactico-statistical LM uses a combination of syntactic and statistical analyses of the text data to model the word sequences [11].

Both word-based and morphology-based language models were studied in Kirchhoff *et al.* [54] for the dialectic Arabic language. The models trained were: word bigrams and trigrams, particle models, class-based models where classes were defined by morphological components, morphological stream models where sequences of morphemes (e.g. stems, morph tags, etc.) are considered individually, and factored language models. The language models trained on morphological text data produced by

the Morfessor and FSM-based morphology learning and segmenting tools, respectively, were investigated in Tachbelie *et al.* [51] for the Amharic language whose writing system is syllabary. In Seng *et al.* [47] - word, syllables and character-cluster sub-word models were investigated in modelling the Khmer language that is written without spaces between words. Combinations or hybrids of the sub-word models were also investigated. Word and morphological random forests, standard word and morphological decision trees, and morphological class models have been studied for the inflective languages Czech and Russian by Oparin [9].

3.7. Language Modelling Tools

There exists processing and development tools for the various stages of LM (the stages as shown in Figure 3.1). For each stage, there may be special tools for that kind of processing – such as text normalization or pre-processing tools, training and testing tools. There are also toolkits that often include processing services and applications for all stages of language model development.

The existing LM toolkits include: Carnegie Mellon University-Cambridge University Statistical LM (CMU-Cam SLM) toolkit [55], Hidden Markov Model toolkit (HTK) for LM (HTK LM) [8], and Stanford Research Institute LM (SRILM) toolkit [56]. Amongst these toolkits, SRILM is the most widely used for LM research [50].

SRILM was first implemented in 1995 and released for public usage in 1999 [56]. Over and above the functionality for text processing and language model training, it has added functionality for perplexity computation, N-best and lattice rescoring, text tagging and text segmentation. The toolkit's functionality is mainly distributed across three layers. The core functionality of the toolkit is contained in the first layer, where C++ libraries, language model classes, data structures, and smoothing methods are found. The second layer consist of functionality that is most relevant to the users of the toolkit: set of executable tools to carry out standard language model building and application tasks, and manipulation of LM lattices, N-best lists and confusion networks. The third layer comprises of scripts to conduct text manipulation tasks such as replacing words with classes and creating word lists from the training data.

The major LM techniques and/or algorithms supported by SRILM include: support for factor language models; a range of smoothing methods (such as Good-Turing, Witten-bell, modified and unmodified Kneser-Ney, additive smoothing, natural discounting, and absolute discounting); methods for reading n-gram counts in google format; vocabulary mapping mechanism to port count statistics; methods for n-gram adaptation; n-gram approximation mechanism of any implemented non-standard language models; a client-server implementation that enables connection of LM computations and applications over a TCP/IP network connection; and vocabulary selection mechanism that allows selection of ranked vocabulary words [57].

The work in Karpov *et al.* [11] used SRILM for statistical text analyses and VisualSynan for syntactic text analysis to develop syntactico-statistical language model. The two tools, SRILM and VisualSynan, were also used in Kipyatkova *et al.* [46] for respective statistical text analysis and for obtaining morphological word features. Factor language models were developed as a result. The features (or factors) obtained were: the word, its lemma, stem, POS (part-of-speech), and morphological tag.

The *ClipText toolkit* [58] can be used at the text processing stage to normalize/process the data into a format suitable for LM development. Morfessor, a freely available morphology learning tool that attempts to identify all morphemes found in a word, can be used for morphological decomposition [51].

3.8. Language Modelling Development

This section details the development framework of the three LM toolkits that were used to implement the LM work of this study: HTK LM, SRIM, and CMU-Cam SLM.

3.8.1. HTK LM development

The LM building process as implemented by the HTK LM toolkit is three phased [8]: in the first phase, the training text is scanned and its n-grams are counted and stored in a database of gram files. In the second phase, some class mapping may be applied and some of the words be mapped to classes such as the OOV class. Then in the final phase, the counts from the resulting n-gram files are used to compute n-gram proba-

bilities which are stored in a language model file. Ultimately, the goodness/effectiveness of the resulting language model can be determined using the perplexity measure on some unseen test text data set. This process is represented by Figure 3.2.

3.8.2. SRILM development

The SRILM toolkit is based on the LM development framework shown on Figure 3.8. It is mainly a three steps modelling process [56] [59]: an n-gram count statistics file is firstly generated from the training text; then a language model is estimated using the count file and a word or sub-word unit lexicon/vocabulary; and lastly, the generated language model is evaluated on the test data. The main SRILM tools that are invoked in these three steps are *ngram-count* and *ngram* as shown on Figure 3.9.

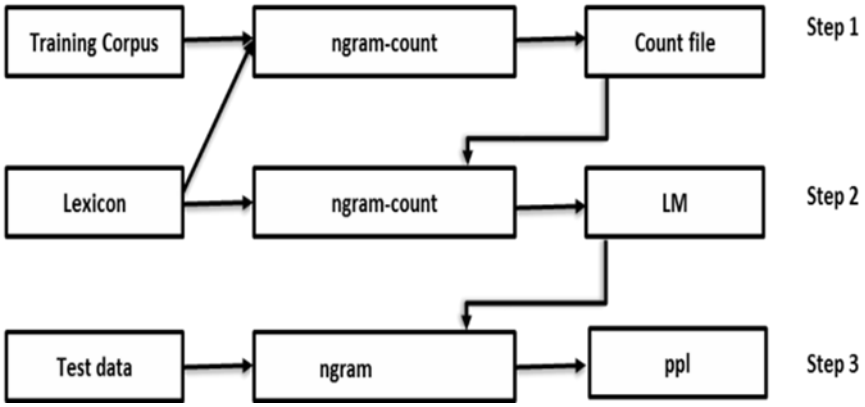

Figure 3.8: SRILM LM framework [51]

3.8.3. CMU-Cam SLM Development

The LM development process as implemented by the CMU-Cam SLM toolkit is captured on Figure 3.9. At the first stage, the language model's vocabulary is defined from the training text using the tools *text2wfreq* and *wfreq2vocab*. The first step of the second stage turns the training text into id n-grams (n-grams in which each word is mapped into an integer id). The second step uses the id n-grams and the vocabulary to estimate a language model. The last stage of the LM process evaluates the lan- guage model on the test text using performance measures such as perplexity.

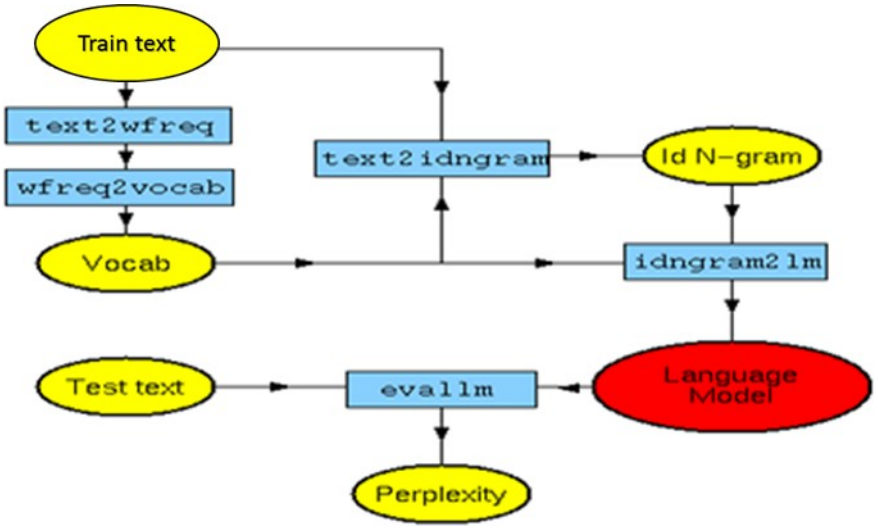

Figure 3.9: CMU-Cam SLM framework [55]

3.9. Related Work

The work of this study takes a focused attention on LM for under-resourced languages. The under-resourced languages considered are Ndebele and Pedi, SA Nguni and Sotho languages respectively. One peculiar characteristic of these languages is their writing system or orthography: they are conjunctively- and disjunctively-written respectively. This means that the morphemes are written separately/disjunctively in the Sotho languages while they are often written clustered to form single words in Nguni languages [4]. The interest is on determining the effect that this unique writing system may have on the quality of the language model produced to model word sequences for these languages. Studies done for other under-resourced languages in the context of LM are now explored.

Improvements in LM for the dialectic Arabic language were investigated by Kirchhoff *et al.* [54] by development of various morphology-based language models. Among the LM challenges associated with Arabic are its complex morphology, huge dialectical variability, and differences between the spoken and written forms of the language. The stated challenges lead to increased PPL and OOV rates of language models developed for the language. To address the associated challenges and thus improve the LM capability of the language models, four types of morphology-based language mod- els were explored: particle based models, morphological stream models, class-based

models, and factored language models. Improved LM results from some of the morphology-based language models are reported over standard word-based models. Significant WER reductions on an ASR application system are also reported.

Factored language models were also researched for rescoring N-best lists for the morphologically rich inflective Russian [46]. To adequately model the rich morphology of the language, very large vocabulary and thus more data are required. Lack of such vocabulary and data leads to increased OOV entries. The influence of factor models on LM PPL and recognition WER were investigated. Language models were developed using five factors: the word, its lemma, stem, part-of-speech, and morphological tag. The conducted experiment on large vocabulary continuous Russian speech recognition showed that FLM can reduce WER.

The Russian language is also under-resourced in terms of language technology resources and research. To further develop better and improved statistical models for Russian, and address LM challenges associated with the language, Karpov *et al.* [11] introduced a method that empowers statistical text analysis with syntactic analysis. Here, text data was both statistically and syntactically analysed before n-gram LM development. Mainly, the method increases n-gram (bigram) coverage with the consideration of non-adjacent bigrams (i.e., bigrams from grammatically connected words separated by other words). The increased n-gram coverage reduces WER for large vocabulary continuous speech recognition (LVCSR) of Russian. Software tools used in the study include SRILM for LM statistical analysis, VisualSynan software for LM syntactical analysis, HTK for training of acoustic models, and Julius version 4.2 for decoding. The created language models used Kneser-Ney discounting and did not apply n-gram cut-off.

The study by Oparin [9] implemented morphological random forest language models for automatic speech recognition of Russian and Czech languages. These languages are inflectional by nature characterised by a relatively free word order, reflected on the lexical level by rich morphological and derivation system. Morphological random forests are tree-based language models that incorporate various sources of morphological information into a language model. This work showed that exploiting random mor-

phological information using the random forest approach helps improve perplexity performance of language models, and ultimately recognition accuracies of Czech and Russian ASR systems. Word and morphological random forests were trained and then compared with a standard n-gram language model, word and morphological decision trees' models, and morphological class models. Morphological random forests reported superior LM performance.

In Seng *et al.* [47] different views of the text data (word and sub-word units) were exploited for the LM of the under-resourced Cambodian Khmer language. Language characteristics associated with Khmer that are challenging for ASR include: lack of text and speech language resources in digital form; a writing system with no explicit word boundary (i.e., like Chinese and Thai, Khmer is written without spaces between words) and thus requiring automatic segmentation into words or sub-words to make LM feasible; and inadequately studied acoustical and phonological characteristics. The LM work implemented uses a sub-word vocabulary and corresponding sub-word training data (syllable and character-cluster sub-words), a word vocabulary, and a hy- brid sub-word/word (character-cluster/word) vocabulary.

The hybrid character-cluster/word vocabulary is created by progressively adding N-most frequent words to the character-cluster vocabulary (N ϵ {1 000, 5 000, 10 000, 15 000, and 20 000}). Trigram language models were trained on the different vocabularies. Language models developed using different corpora were linearly interpolated, and interpolation parameters were tuned on a separate development text set. For evaluation, the language models were tested on a speech recognition system and three measures of performance were used: WER, syllable error rate (SER), and character-cluster error rate (CCER). CCER gave more accurate evaluation. For the recognition task, the word-based language model performed best. The progressively built hybrid character-cluster/word language model performed like (and elsewhere slightly better than) the word-based language model when the hybrid vocabulary contained at least 5 000 words.

Word and morpheme language models were investigated for the under-resourced morphologically rich Ethiopian Amharic language [51]. Coupled with data sparseness problem (having insufficient relevant training data) is the high OOV problem for ASR

of under-resourced languages. An OOV word is misrecognised by an ASR system, causing the neighbouring words to be also misrecognised. On average, an OOV word in the test data contributes to 1.6 errors in the speech recognition system. The approach of vocabulary optimization (where the vocabulary is selected such that the OOV rate is reduced by, for example, increasing the vocabulary size or including more frequent words in the vocabulary – which in turn require increased amounts of data) does not work well for under-resourced languages due to the lack of substantial amounts of relevant data. System complexity is also increased with large vocabularies for morphologically rich languages. Thus, different approaches of dealing with the OOV problem are sought, and here morpheme-base language models are investi- gated.

Naturally, sub-word vocabularies are smaller than word vocabularies. Text was segmented into morphemes using the unsupervised morphological segmentation method (employing the Morfessor tool) and a finite state machine-based supervised method. Word and morpheme-based trigram models have been developed using the SRILM toolkit. The language models were smoothed with modified Kneser-Ney smoothing, and all trigrams (regardless of their number of occurrence) found in the training data were included in the models. The study concluded that the use of morphemes for lexical and language modelling together with syllables and/or hybrid acoustic units for acoustic modelling is best for Amharic speech recognition.

Sub-word units were investigated and found to reduce PPL significantly for the morphologically rich agglutinative Turkish language [48]. The LM challenges posed to the standard LM techniques by the language structure of such a language include the large OOV rates and modelling of regularities (such as morpheme syntactic depend- encies) possessed at sub-word level. Split words (words split into their stem and suffix components) n-grams and flexible n-grams (n-grams that condition the probability of a token on the previous n-1 tokens anywhere in the token sequence, not only on the preceding adjacent n-1 tokens as used by standard n-grams), derived using a morphological analyser and a disambiguator, are investigated over the standard word n-grams. The toolkit used for LM development was SRILM, and interpolated Kneser-Ney smoothed language models were developed. Split-words and flexible hexagrams

achieved a PPL reduction of 25% and 27% respectively, over the standard word hex-agram.

On the local front, the SA context, a few NLP resource-creation projects for the SA languages are noted: the African Speech Technology (AST) project [60], the LWAZI project [61], and the National Centre for Human Language Technology (NCHLT) project [6] [62]. These three projects (and in their order) were publicly-funded by the SA Government.

The AST project led to the creation of annotated speech corpora for five SA languages (Xhosa, Sesotho, Zulu, English, and Afrikaans), and ASR and TTS systems for deployment in a multilingual hotel booking system (a prototype) [60]. This was a four years project (2000 to 2003) and was funded by the Department of Science and Tech- nology (DST). The development of indigenous languages at a technological level for modern ICT and to help facilitate information access for all citizens, these were the motivating reasons at the back of the project.

Considering the different speech varieties used by mother-tongue and non-mother-tongue speakers, eleven (11) transcribed speech databases were developed: 5 for English, 3 for Afrikaans, 1 for Xhosa, 1 for Zulu, and 1 for Sesotho. The speech varieties for English were mother-tongue speakers' variety and four non-mother-tongue speakers' varieties (Black, Coloured, Asian, and Afrikaans speakers). The speech varieties for Afrikaans included the mother-tongue speakers' variety and two non-mother-tongue speakers' (Black and Coloured) varieties. For each speech database, between 300 and 400 speakers were recruited, and phone call recordings of each speaker had about 40 utterances of a mixture of spontaneous and read speech.

The eventual SLU or spoken language dialogue prototype system that was developed for a hotel reservation task had its ASR sub-system developed using HTK and the TTS sub-system using Festival toolbox[6]. Usability tests that used mother-tongue speakers from three languages (Afrikaans, English, and Xhosa) indicated high percentage rates of successful bookings. Out of between 78 and 88 calls: 83%, 77%, and 60% were the

[6] *www.festfox.org*

successful bookings through the Afrikaans, English, and Xhosa prototype systems, respectively.

The LWAZI project [61] [63] [64] carried on the aims of the AST project in demonstrating the usage of speech technology for information service delivery in SA. The three years long (2006 to 2009) LWAZI project produced core tools, technologies, and linguistic resources for the development of multilingual spoken dialogue systems (SDSs) in all the eleven official SA languages. The project was funded by the Department of Arts and Culture (DAC).

The speech technology resources developed include: DictionaryMaker[7] (atoolkit used for bootstrapping of pronunciation dictionaries), ASR-Builder[8] (tools for training and experimenting with acoustic models for speech recognition), Speect[9] (toolkit for TTS system development), LWAZI platform (Asterik platform for the development and deployment of SDSs), corpora (speech and text corpora for the development of ASR and TTS systems), phoneme sets, and electronic pronunciation dictionaries (with approximately 5000 words) [65]. The ASR speech data (approximately 5-10 hours long, per language) is made up of read and elicited speech, recorded over a telephone channel, from about 200 speakers (each producing 30 utterances) per language.

Phone and word recognition ASR systems were developed. The systems were developed using the HTK toolkit, with a flat phone-based language model employed for phone recognition. The recognition results from the developed systems included: phone-recognition correctness (i.e., percentage of correctly recognised phone labels relative to total number of expected phone labels), phone-recognition accuracy (phone-recognition correctness taking into consideration phone label insertions and deletions as well), phone PPL (bigram PPL on phoneme sequences that occur in the training data), and word-recognition accuracies from a selected small ten-words vocabulary recognition task. English recognisers (LWAZI, Ntimit, and WSJ) were used to perform the cross-language transfer procedure where a well-trained recogniser for

[7] *www.dictionarymaker.sourceforge.net*
[8] *www.asr-builder.sourceforge.net*
[9] *www.speect.sourceforge.net*

a well-resourced language like English is used to recognise utterances of an under-resourced language.

The NCHLT text and speech corpora projects like LWAZI, were done with collaboration by the DAC, Council for Scientific and Industrial Research (CSIR), and North-West University (NWU). Like its predecessors, i.e., the AST and LWAZI projects, the publicly funded NCHLT projects continued the development of linguistic and speech technology resources for the advancement of research and development in relation to NLP for all official SA languages. Unlike its predecessors, however, NCHLT was earmarked for the development of large vocabulary and broadband corpora.

From the text corpora project, data resources and associated core technologies were developed for ten languages (excluding English) [6]. The data resources were: domain specific monolingual unannotated corpora and parallel annotated corpora (annotated at the token, orthographic, morphological, and morpho-syntactic layers). The associated core technologies developed were: tokenisers, sentecisers, lemmatisers, POS taggers, and morphological decomposers.

The NCHLT speech corpora project produced as results speech technology resources that include: orthographically transcribed speech corpora (approximately 50 hours long for each language, 800 hours in total, and from around 200 different speakers per language), pronunciation dictionaries (with approximately 15 000 words per language), benchmark ASR and TTS systems, and a data collection smartphone tool (Woefzela) [62].

The ASR systems were built using Kaldi and HTK to measure both word and phone accuracies during speech recognition. Phone-based systems were developed using HTK and word-based systems using Kaldi. The word-based systems used two types of language models: modified Kneser-Ney 3-gram (trigram) and 4-gram (quadrigram), and an ergodic word loop.

An audit conducted in 2009 by Grover *et al.* [66] established the landscape of the South African languages with regards to human language technologies. The audit

came about after the realisation that the HLT research and development (R&D) community – comprising of universities, science councils, and private companies – was not thriving as expected given the many opportunities of the field. The R&D activities were found to be fragmented and lacking central coordination. Hence the establishment of the audit as a step to improving on the situation by identifying HLT resources that were available and the amount of work being done for each SA language, and what needs to be done further towards creating a thriving HLT R&D industry.

The audit was conducted through various steps: from developing the HLT audit terminology (to create a common frame of reference), defining an HLT inventory criteria framework (which specified the criteria on which HLT resources would be audited and documented), creating the audit questionnaire (to aid with the recording of information for various categories of resources – data, modules, applications, tools and platforms), toperforming an inventory gap analysis (which identified gaps between current status of HLT components in South Africa and HLT components prioritised by the SA HLT community) upon feedback from the collected information.

As part of the output from the audit, the following frameworks resulted: the HLT language index (list that ranks SA languages based on total HLT activity within the language, as well as the stage of maturity and accessibility of the language's resources and applications), HLT component index (which provides an alternative perspective of the quantity of activity taking place within the various HLT component groupings – data, modules, and application categories), maturity index (level of maturity for the HLT component), accessibility index (accessibility of the language resources and applications), HLT inventory analysis, and an inventory gap analysis representations.

The mentioned representations are available as an online resource[10], and *application based language models* for complete speech recognition are prioritised as needed speech modules. Although there has been much significant HLT work since the completion of the audit, the findings remain relevant even today and help guide on-going efforts for HLT R&D research, such as the efforts of this study.

[10] *https://static-content.springer.com/esm/art%3A10.1007%2Fs10579-011-9151-2/MediaObjects/10579_2011_9151_MOESM1_ESM.pdf*

With the explored literature, one notes similar objectives that were studied in comparison with our study:

i. studying LM for under-resourced languages;
ii. developing languages models with consideration to the unique nature of a language;
iii. employing standard LM methods such as: word-based LM, n-gram LM, interpolation, LM smoothing/discounting techniques such Kneser-Kney smoothing;
iv. using standard development toolkits such as SRILM;
v. varying the LM development approach to improve the quality of the models and thus PPL rates;
vi. developing NLP resources for SA ‘s indigenous languages;

Simultaneously, we observe the following main differences (in objectives):

i. Using other standard LM techniques such as morphology-based random forests and FLM to incorporate various sources of morphological information into the developed model over and above the word level information;
ii. examining the developed language models in a speech recognition system to determine influence on the systems’ performance;

Chapter 4: Methodology for Experiments

This chapter describes the methodological development framework on which this study is based. The question of the need for LM research is explored in Section 4.1. Section 4.2 describes the secondary data used to conduct the work. Sections 4.3, 4.4, and 4.5 briefly describe the smoothing, interpolation and back-off, as well as higher- order n-gram LM techniques, respectively, that were implemented in the experiments. To end the chapter, Section 4.6 details the conducted experiments.

4.1. Significance of the Study

Several reasons, theoretical and practical, have and continue to influence and motivate research studies of language models. We survey some of these reasons and thereby attempt signifying the necessity of continuing the investigation of better language models as ASR continues to evolve. At the onset, we are faced with the major LM problem: which LM unit, approach or technique, tools and toolkits best model a specific language task?

The nature of, and thus the need to specially optimize LM development for, the language or language task; the interest to advance further standard LM approaches and techniques; addressing specific LM challenges such as data scarcity and efficient smoothing (and those mentioned in Section 3.5); and incorporating additional language information such as syntactic, semantic, and pragmatic information for improved modelling - these are amongst some of the reasons signifying LM research.

Varying language model smoothing techniques such as absolute discounting, Good-Turing, un-modified and modified Kneser-Ney, Witten-Bell, and so on, have provedto lead to that smoothing technique that best models a language or language task. For example, modified Kneser-Ney smoothing, a variation of Kneser-Ney, was found to best model English [13]; and modified Kneser-Ney smoothing and absolute discounting for Portuguese [12].

Under-resourced languages, also known as low density or low developed languages, mainly lack sufficient resources and tools required for implementation of human language technologies such as speech recognisers [3]. Their lack of unique writing system or stable orthography; limited presence on the web; lack of linguistic expertise; and lack of NLP resources such as text and speech corpora, dictionaries, robust acoustic and language models makes it difficult to port successfully and with ease HLT systems that are available for the resourced languages of the world such as English and Spanish. Furthermore, the complex morphology of many of these languages aggravates the data sparsity problem associated with languages that lack sufficient development data. It thus seems necessary to identify and recommend methods that will model the unique nature of a language and make best with the limited resources available for that language.

Exploiting different views of the text data, or sub-word units other than whole words, has yielded noteworthy results for various (under-resourced) languages. The influence of a word, its lemma, stem, POS, and morphological tag on the quality of the language model for Russian was investigated by Kipyatkova *et al.* [46]. Additional information was incorporated in the models for English by Deoras *et al.* [44]. Here, long context information, i.e., relations between non-adjacent words in a sentence, were incorporated in LM for a multi-stage recognition process. Additional syntactic information was incorporated to statistical models via formal grammars by Kaufmann and Pfister [43].

The investigation of better models that advance the standard n-gram models has seen several efficient types of models being contributed as outcomes. Some of these advances were elaborated on in Section 3.3. Such models include factored language models [53], neural network language models [28], approximated models [44] and maximum entropy language models [50]. Especially over the standard trigram model, other improvements found are: higher-order n-grams (beyond trigrams), caching, skipping, interpolated, modified Kneser-Ney smoothed, clustered, and sentence mixture models [35].

LM challenges, some unique to a language, have always motivated LM research. It has been found that data sparsity and word error segmentation, for example, could be addressed by exploiting different views of the text data other than the standard whole

word [47]. More training data for data sparsity, and varied smoothing and pruning techniques have shown to lead to better LM parameter estimates [12]. Given that the internal structure of language models is language dependent, unlike acoustic models, many theoretical and practical LM advances cannot be applied to different languages and similar gain be realized [9]. Languages belonging to different language families usually differ greatly in structure and thus unique features, if not different approaches, should be developed for LM.

4.2. Data

The training data used in this study was obtained or acquired from the LWAZI [63] and the NCHLT [62] transcribed speech corpora projects. These were projects done with collaboration by the CSIR, DAC, and the NWU as part of the ongoing efforts for developing large reusable resources for spoken language processing of under-resourced languages of South Africa. The data is managed and distributed by the Resource Management Agency (RMA) at NWU [67] and is freely downloadable for research purposes. As exemplified in these projects, we develop language models using the orthographic transcriptions of the produced speech.

The LWAZI data was developed from 200 speakers with about 30 transcribed utterances each, producing about 5 to 8 hours of speech for each of the eleven SA languages. About 148 to 210 speakers produced the NCHLT data, with around 56 hours of speech transcribed for each language. Both forms of data were compiled for developing sufficient speech recognition vocabularies using read and elicited recorded prompts. The LWAZI data was telephone-based recordings and NCHLT recordings were smartphone-based. Tables 4.1 and 4.2 summarise the two corpora.

Table 4.1: LWAZI data summary [63]

Language	Code	# total minutes	#speech minutes	# distinct phones
Afrikaans	Afr	213	182	37
SA English	Eng	304	255	44
isiNdebele	Nbl	564	465	46
Sepedi	Nso	394	301	45
Sesotho	Sot	387	313	44
siSwati	Ssw	603	479	39
Setswana	Tsn	379	295	34
Xitsonga	Tso	378	316	54
Tshivenda	Ven	354	286	38
isiXhosa	Xho	470	370	52
isiZulu	Zul	525	407	46

Table 4.2: NCHLT data summary [62]

Language	Code	Speakers	Words		Duration
			Types	Tokens	
Afrikaans	Afr	210	8640	191023	56:22
SA English	Eng	210	8351	222884	56:25
isiNdebele	Nbl	148	15283	151276	56:14
isiXhosa	Xho	209	29130	136904	56:15
isiZulu	Zul	210	25650	130866	56:14
Sepedi	Nso	210	11196	294081	56:19
Sesotho	Sot	210	10600	273834	56:19
Setswana	Tsn	210	5610	280853	56:19
siSwati	Ssw	197	12246	132225	56:14
Tshivenda	Ven	208	7728	245510	56:16
Xitsonga	Tso	198	6118	236062	56:16

Guided by the planned experiments, data was downloaded for all the Nguni (Ndebele, Zulu, Swati, and Xhosa) and Sotho (Pedi – also known as Northern Sotho, Southern Sotho, and Tswana) languages. The data was specially prepared before LM development was carried out as detailed by the next sub-section.

4.2.1. Data Preparation and Analysis

The text is prepared or pre-processed to a format suitable for LM development. The preparation procedure data gets tuned for optimal development on a specific toolkit.

Using the *tocorpus.pl*[11] text data preparation script, the original acquired text was processed. Amongst other text elements, sentence constructs such as the following were removed in order to not form part of sentence/n-gram context during n-gram development: ellipses (…), unicodes (character codes), dashes surrounded by spaces and those at word ends (as in "phrase **–** phrase" and "three**-** to five-years"), some punctuation markers (such as , ; : % ! i ()), quotes, trailing spaces, double spaces, line/sentence leading spaces, starting and trailing tags (such as <s> and </s>, <orth> and </orth>, and), and annotation tags. The prepared text was added with sentence boundary markers <s> and </s> for HTK LM development. All text was case folded to lower case. Table 4.3 shows instances of prepared Pedi and Ndebele sentences.

Table 4.3: Pre-processed text examples

Pre-processed Text Examples			
Text Version	**Pedi**		**Ndebele**
Original	le ge [s] Lepelle la ka go, le ka goa [um] bjang [n], le bjang , [s] ke tla yo bona moro- , moratiwa, [s] wa ka gosasa.		[n] [um] lilanga, elimatjhumi amabili [n], nabunane, kuNtaka.
Without sentence boundary markers	le ge lepelle la ka go le ka goa bjang le bjang ke tla yo bona moro- moratiwa wa ka gosasa		lilanga elimatjhumi amabili nabunane kuntaka
With sentence boundary markers	<s> le ge lepelle la ka go le ka goa bjang le bjang ke tla yo bona moro- moratiwa wa ka gosasa </s>		<s> lilanga elimatjhumi amabili nabunane kuntaka </s>

The data was partitioned into train and test sets as detailed in Table 4.4. The LWAZI data was divided using the ratio of 80% train and 20% test sets; the NCHLT data came partitioned already into train and test sets (with test sets comprising data from 8 speakers and the rest for training sets). The training sets were augmented further when data pooling was done from related languages.

[11] *http://source.cet.uct.ac.za/svn/people/smarquard/sphinx/experiments/scripts/tocorpus.pl*

Table 4.4: Train and Test data statistics

Corpus	Train					Test					Total		
	# words	# sentences	Average #words per sentence	Min. Sentence length	Max. Sentence length	# words	# sentences	Average #words per sentence	Min. Sentence length	Max. Sentence length	# words	# sentences	Average #words per sentence
IsiNdebele_LWAZI	33098	4810	**7**	**1**	**26**	8173	1203	**7**	**1**	**31**	**41271**	**6013**	**7**
IsiNdebele_NCHLT	140871	39415	**4**	**1**	**16**	10405	3108	**3**	**1**	**11**	**151276**	**42523**	**4**
Sepedi_LWAZI	45206	4512	**10**	**1**	**38**	11317	1128	**10**	**1**	**32**	**56523**	**5640**	**10**
Sepedi_NCHLT	279995	56284	**5**	**1**	**6**	14086	2829	**5**	**2**	**5**	**294081**	**59113**	**5**
IsiNdebele_LWAZI+NCHLT	173969	44225	**4**	**1**	**26**	18578	4311	**4**	**1**	**31**	**192547**	**48536**	**4**
	173969	44225	**4**			8173	1203	**7**			**182142**	**45428**	**4**
	173969	44225	**4**			10405	3108	**3**			**184374**	**47333**	**4**
Sepedi_LWAZI+NCHLT	325201	60796	**5**	**1**	**38**	25403	3957	**6**	**1**	**32**	**350604**	**64753**	**5**
	325201	60796	**5**			11317	1128	**10**			**336518**	**61924**	**5**
	325201	60796	**5**			14086	2829	**5**			**339287**	**63625**	**5**
Nguni_LWAZI	146509	22675	**6**			8173	1203	**7**			**154682**	**23878**	**6**
Nguni_NCHLT	540866	174241	**3**			10405	3108	**3**			**551271**	**177349**	**3**
Nguni_LWAZI+NCHLT	687375	196916	**3**			18578	4311	**4**			**705953**	**201227**	**4**
	687375	196916	**3**			8173	1203	**7**			**695548**	**198119**	**4**
	687375	196916	**3**			10405	3108	**3**			**697780**	**200024**	**3**
Sotho_LWAZI	144228	16509	**9**			11317	1128	**10**			**155545**	**17637**	**9**
Sotho_NCHLT	834682	172227	**5**			14086	2829	**5**			**848768**	**175056**	**5**
Sotho_LWAZI+NCHLT	978910	188736	**5**			25403	3958	**6**			**1004313**	**192694**	**5**

The average sentence (or word sequence) lengths in Table 4.4 show that the LWAZI sentences were designed (as prompts) longer than the NCHLT sentences. Because of the bigger size of the NCHLT corpus, the combination with the LWAZI corpus resembles more of the properties of the NCHLT text such as average sentence size. The average sentence length also suggests that n-gram development will go up to around this size for the different corpora, after-which LM evaluation of test sentences will rely on measures such as backing-off to lower n-grams to arrive at PPL estimates.

4.3. Language Model Smoothing Techniques

Language modelling mainly uses n-gram (i.e., word sequence) frequency counts from the training data to arrive at probability rates for the likeliness of a word sequence. Without smoothing, the MLE estimation method is used on its own for the estimation of these probabilities. In MLE estimation, using n-gram counts from the training data: more frequent n-grams will have high probabilities, less frequent n-grams low probabilities, and n-grams not present in the training data zero probabilities [1]. Even to those acceptable n-grams (according to the rules of the language, or generally known from human intuition), incorrect zero or low probabilities are assigned when they have no or less frequency counts according to the frequency distribution of the training data.

Furthermore, given that a language is ever evolving and that part(s) of the word sequence may be correct when the entire sequence is not (e.g., individual words correct on their own) – it is thus difficult to arrive with certainty that a word sequence is entirely not probable, that it has a zero probability. Smoothing methods (SMs) are there to aid LM estimation in conditions when there are these zero and low-frequency counts leading to zero and poor low probabilities associated with n-grams [1] [13].

There exist several smoothing techniques including the following that were implemented in this study: Good-Turing (GT), Absolute discounting (AD), Witten-Bell (WB), Linear discounting (LD), Additive smoothing (AS), Natural discounting (ND), modified Kneser-Ney (KN), and unmodified Kneser-Ney (UKN).

4.4. Language Model Interpolation and Back-off

Language model back-off and interpolation are other ways that help deal with the problem of zero counts leading to zero probable n-grams, by relying on the n-gram hierarchy [1]. In back-off, LM estimation "backs off" to a lower n-gram whenever there are higher-order n-grams not seen in the training data. In LM interpolation, probability estimates are mixed from all n-grams in the estimation task.

For example, a trigram model may be interpolated with unigram and bigram models as follows [35]:

$$\begin{aligned} P_{interpolate}(w|w_{i-2}w_{i-1}) &= \lambda P_{trigram}(w|w_{i-2}w_{i-1}) \\ &+ (1-\lambda)[\mu P_{bigram}(w|w_{i-1}) \\ &+ (1-\mu)P_{unigram}(w)] \end{aligned} \quad (4.1),$$

where λ and μ are constants such that $0 \leq \lambda \leq 1$ and $0 \leq \mu \leq 1$. In back-off, when the trigram is observed in the training data – the count is used in estimating the trigram's probability; otherwise – estimates from the lower bigram and unigram are recursively used to estimate the trigram's probability. That is:

$$P(w|w_{i-2}w_{i-1}) = \begin{cases} P^*(w|w_{i-2}w_{i-1}) \text{ if } c(w_{i-2}w_{i-1}w_i) > 0, \\ \alpha(w_{i-1}w_i)\, P^*(w_i|w_{i-1}) \text{ elseif } c(w_{i-1}w_i) > 0, \\ \alpha(w_i)\, P^*(w_i) \text{ otherwise.} \end{cases} \quad (4.2),$$

where α is a constant such that $0 \leq \alpha \leq 1$.

Language model interpolation also acts as one of the ways of combining LM techniques and/or language models together [35] [13]. Generally, higher-order n-grams are combined with lower-order n-grams, whilst elsewhere differently developed n-grams may be combined for efficient estimation. In the combination/interpolation, the different techniques or n-grams empower each other during estimation to avoid estimating probabilities of zero for some word sequences [13].

4.5. Higher-order N-grams

Trigrams (i.e., n-grams of order 3, relying on two words contexts/histories) have proven to be the best performing standard n-grams [35]. However, in other LM tasks – longer or higher-order n-grams may be helpful and give better word sequence estimates than the conventional trigram model. The word histories or contexts relied on by the higher-order n-grams to predict the probability of a word may not be found in the training data, in such a case, techniques such as smoothing, back-off and interpolation are useful to help use estimates from lower n-grams.

4.6. Experimentation

The nature of the experimentation work determines best LM approaches to develop language models for the orthographic Pedi and Ndebele text for speech recognition. The designs of the experiments and how they were implemented are discussed next.

4.6.1. Experiment Design and Implementation

This section details the experiment work carried out in this study. The experiments were conducted using three LM toolkits: SRILM, HTK LM, and CMU-Cam SLM.

When it was possible according to the limitations of - amongst others - the data, toolkit or LM approach, the experiments had n-gram development up to n-gram order 6 as a

default setup, and up to order 20 for experiment 3. The LM smoothing methods supported in the toolkits were all exploited in the experimentation. The CMU-Cam SLM toolkit supports GT, AD, WB, and LD. The SRILM supports GT, AD, WB, AS, ND, KN, and UKN. GT is default for both toolkits. The HTK LM was used with GT discounting.

Effective n-grams and smoothing methods for LM development of the two languages were thus investigated. In all the experiments, either the Ndebele or Pedi test data set was used when testing the developed models.

4.6.1.1. Experiment 1: Baseline N-gram Models

Ndebele and Pedi baseline n-gram language models were developed on the prepared data using the three toolkits: SRILM, HTK LM, and CMU-Cam SLM. The development framework for these toolkits is as described in Section 3.8.

4.6.1.2. Experiment 2: Pooling data

Text data was pooled from the two different text corpora (LWAZI and NCHLT), and from languages belonging to the same group (Nguni or Sotho). The different pooled data combinations were thus: Ndebele LWAZI+NCHLT, Nguni LWAZI, Nguni NCHLT, Nguni LWAZI+NCHLT, Pedi LWAZI+NCHLT, Sotho LWAZI, Sotho NCHLT, Sotho LWAZI+NCHLT. Experimentation was then carried out on the different data combinations.

Table 4.5 presents the unique number of words per data set, with the unique words in non-pooled sets subsets of the pooled sets.

Table 4.5: Unique words count per data set

Text	Total # Words	Unique # words
Ndebele_LWAZI	33098	4503
Ndebele_NCHLT	140871	14930
Ndebele_LWAZI+NCHLT	173969	18293
Pedi_LWAZI	45206	3171
Pedi_NCHLT	279995	11083
Pedi_LWAZI+NCHLT	325201	12862
Nguni_LWAZI	146509	17975
Nguni_NCHLT	540866	72026
Nguni_LWAZI+NCHLT	687375	84238
Sotho_LWAZI	144228	7664
Sotho_NCHLT	834682	22911
Sotho_LWAZI+NCHLT	978910	27097

4.6.1.3. Experiment 3: Higher-order N-grams

Language models were developed up to n-gram order 20 when it was possible with the toolkit, to determine the performance trend of the models as they increase in n-gram order. Other experiments developed up to n-gram order 6. N-gram order 6 was chosen because language model training for hexagrams is the maximum allowable for toolkits such as HTK LM and SRILM; and 20 was randomly chosen considering that the length of Pedi sentences could easily be 20 and more words. For each language, the experiment considered the different sized text in LWAZI, NCHLT, LWAZI+NCHLT, and cluster grouped text.

4.6.1.4. Observations from the first three experiments

At this stage of development, observations were made based on the initial three experiments to arrive at insights as to which n-grams and which smoothing methods modelled better the different texts and thus the different languages. The drawn insights were meant to inspire the manner of development to be carried out in the other experiments that were to follow.

4.6.1.5. Experiment 4: Interpolation

Experiment 4 combined different language models. Models were combined using SRILM's tools, during the training stage through the *"-interpolate"* parameter and during testing through the *"-mix-lm"* parameter. An experiment based on the LWAZI and NCHLT data was re-conducted with interpolation enabled during training. Best performing models from previous experiments were then combined with the hope of yielding an improved model out of them with the *"mix-lm"* parameter enabled at the testing stage.

Chapter 5: Results and Discussion

Chapter 4 outlined the methodological approach of the study and described the nature and details of experiments that were designed. This chapter presents and discusses the results from the implementation of the experiments. The results are presented and analysed for each experiment at a time after analysing the vocabulary statistics of the LM training text. The chapter concludes by discussing the observed results.

5.1. Vocabulary Statistics

The vocabulary statistics presented in Table 5.1 map the unique most frequent words found in the training text, the lists of words are required for language model development. The words found in the testing data but unknown to the language model (as captured by the vocabulary lists) are modelled as out-of-vocabulary (OOV) words. The vocabularies consist of the top 20 000 words that are most frequent. The vocabularies were open such that OOV words were mapped to a special token such as "*!!UNK"* during LM estimation. The vocabulary tools "*text2wfreq*" and "*wfreq2vocab*" from the CMU-Cam SLM toolkit were used to define the vocabularies.

In general, the statistics show that the bigger the size of the training text the bigger the vocabulary mapping the unique topmost frequent words. The vocabulary size, as was the text size in Section 4.2.1, increases as you analyse Ndebele/Pedi LWAZI, Ndebele/Pedi NCHLT, Nguni/Sotho LWAZI, Ndebele/Pedi LWAZI+NCHLT, Nguni/Sotho NCHLT and Nguni/Sotho LWAZI+NCHLT texts, in that order.

We note in the LWAZI+NCHLT and Nguni/Sotho LWAZI texts that the effect of pooling the data in this manner reduces the SRILM and CMU-Cam SLM's OOV rates for the corresponding testing data. This observation suggests that pooling the data this way increases the coverage of the vocabulary with the addition of extra commonly used unique words. A similar observation can be observed from the HTK LM rates for the Ndebele test data, however, the Pedi test data rates are on the contrary increasing. Whilst we suspect that this contrary increase, in comparison with results from the other two toolkits, may be influenced by the nature of the toolkit more than that of the language, we could not verify this assertion.

The reduction in OOV rates is also not seen for the Nguni/Sotho NCHLT and LWAZI+NCHLT texts, here the rates significantly increase. Given that the test data is the same in both the grouped and non-grouped cases, the worsening of the rates in the grouped cases may be due to the limitation of the vocabulary containing the most frequent 20 000 words from the training corpora. This limited vocabulary is not wide enough to match the augmented training data (because of pooling and grouping) and thus does not include all the most frequent words as contained by the non-grouped text vocabulary (which alone make more than half of the maximum vocabulary size).

Table 5.1: Text Vocabulary Size and OOV rates

Train	Test	Vocabulary Size (most frequent 20000 words)	Out-Of-Vocabulary	
			OOV	OOV%
Ndebele_LWAZI	Ndebele_LWAZI	4503	426	5.21
Ndebele_NCHLT	Ndebele_NCHLT	14930	367	3.53
Pedi_LWAZI	Pedi_LWAZI	3171	319	2.82
Pedi_NCHLT	Pedi_NCHLT	11083	158	1.12
Ndebele_LWAZI+NCHLT	Ndebele_LWAZI+NCHLT	18293	717	3.86
	Ndebele_LWAZI	18293	372	4.55
	Ndebele_NCHLT	18293	345	3.32
Pedi_LWAZI+NCHLT	Pedi_LWAZI+NCHLT	12862	393	1.55
	Pedi_LWAZI	12862	244	2.16
	Pedi_NCHLT	12862	149	1.06
Nguni_LWAZI	Ndebele_LWAZI	17975	353	4.32
Nguni_NCHLT	Ndebele_NCHLT	20000	2032	19.53
Nguni_LWAZI+NCHLT	Ndebele_LWAZI+NCHLT	20000	3711	19.98
Sotho_LWAZI	Pedi_LWAZI	7664	264	2.33
Sotho_NCHLT	Pedi_NCHLT	20000	322	2.29
Sotho_LWAZI+NCHLT	Pedi_LWAZI+NCHLT	20000	719	2.83

5.2. Baseline N-gram Models

This section reports on the languages models developed as per Experiment 1 setup designed in the previous methodology chapter. Results from the three toolkits are analysed. The language model performance from this experiment will serve as a baseline on which models from the other experiments will be compared.

5.2.1. HTK LM Results

LM experimentation using the HTK LM toolkit estimated statistical language models using the default GT smoothing method, for the allowable n-gram orders 1-6. Testing, however, is allowed up to n-gram order 10 by the toolkit. The data used in developing the toolkit's models had sentence boundary markers. Table 5.2 shows the PPL results of the baseline language models with the associated OOV rates. The results are for the GT smoothed unigram to hexagram models for both LWAZI and NCHLT corpora.

Table 5.2: Baseline PPL values using HTK LM toolkit

Language	Corpora	SM	N-gram PPL						OOV%
			1g	2g	3g	4g	5g	6g	
Ndebele	LWAZI	GT	657.60	11.47	8.83	8.83	**8.75**	8.77	4.39
	NCHLT		675.78	46.34	**30.46**	46.15	42.37	37.38	2.26
Pedi	LWAZI	GT	146.57	23.54	9.63	**9.13**	9.17	9.44	13.95
	NCHLT		45.14	22.63	11.39	**10.29**	10.36	10.85	11.56

The lowest PPL value obtained for the LWAZI text was 8.75 by a 5-gram (pentagram) model, and 30.46 for the NCHLT text by a 3-gram (trigram) model. It is worth noting that the LWAZI pentagram differed with other models with an insignificant PPL amount of not more than 0.10, whilst the NCHLT trigram model differed with the higher models by a PPL difference from 6.00 to 16.00. We recall the sentence length averages from Table 4.4 that indicated that Ndebele LWAZI text sentences are on average 7 words long (train and test), whilst sentences in the NCHLT corpus are on average 4 and 3 words long for the train and test sets, respectively. A correlation between the highest performing n-gram and average sentence length is thus observed.

For the Pedi language, quadrigrams estimated the Pedi text with a relatively low PPL value of 9.13 for the LWAZI text, and 10.29 for the NCHLT text. The differences in PPL performance of the quadrigrams' performance with other higher n-gram models are very low if not insignificant**,** they were found to be less than absolute 1.00. Recalling the sentence lengths averages Table 4.4 (10 words for LWAZI test and train sentences, and 5 words for the NCHLT sets), the quadrigram performance and the relative higher n-grams' absolute PPL difference of 1.00 substantiate why n-grams higher than trigrams perform relatively better.

5.2.2. SRILM Results

The second toolkit used for developing statistical language models was the SRILM toolkit. Tables 5.3 and 5.4 report results for the baseline SRILM language models. The results are for unigrams to hexagram models smoothed differently and not smoothed. KN language model development was not supported for the prepared LWAZI texts, hence the exclusion of the KN smoothing results on both tables for the LWAZI texts.

Observantly, non-smoothed models are not necessarily under-performing when compared to all smoothed models. For example, unsmoothed versus AS smoothed (LWAZI and NCHLT) models for the Pedi language, and unsmoothed versus AD, KN, and AS (NCHLT) smoothed models for the Ndebele language. However, in most cases, and on average, smoothed models consistently outperform the unsmoothed models. For the higher n-gram models (i.e., trigrams to hexagrams), UKN (trigrams) and WB (quadrigrams to hexagrams) models consistently model better the LWAZI Pedi text; GT models for the NCHLT Pedi text; ND models for the LWAZI Ndebele text; and GT models for the NCHLT Ndebele text.

Table 5.3: Baseline PPL values for Pedi using SRILM toolkit

Language	Corpora	SM	N-gram PPL						OOV%
			1g	2g	3g	4g	5g	6g	
Pedi	LWAZI	NoSM	**511.98**	58.32	19.36	17.21	17.00	17.02	2.82
		GT	**511.98**	41.31	13.82	12.34	12.26	12.33	
		AD	**511.98**	38.95	13.34	11.83	11.69	11.72	
		WB	522.15	38.76	12.76	**11.15**	**10.99**	**10.99**	
		UKN	**511.98**	38.50	**12.24**	11.20	11.50	11.82	
		ND	**511.98**	**37.44**	12.74	11.22	11.06	11.06	
		AS	**511.98**	38.95	38.95	11.83	11.69	11.72	
	NCHLT	NoSM	383.98	67.17	26.88	20.91	19.51	18.75	1.12
		GT	383.98	68.48	**21.35**	**14.96**	**13.46**	**12.65**	
		AD	383.98	68.48	33.24	27.29	25.30	24.32	
		WB	382.64	69.85	26.58	19.94	18.30	17.36	
		KN	**380.95**	77.66	31.76	24.47	22.42	19.83	
		UKN	383.98	67.59	26.44	19.44	17.92	16.86	
		ND	383.98	**66.69**	26.20	19.95	18.38	17.47	
		AS	383.98	68.48	68.48	27.29	25.30	24.32	

Table 5.4: Baseline PPL values for Ndebele using SRILM toolkit

Language	Corpora	SM	N-gram PPL						OOV%
			1g	2g	3g	4g	5g	6g	
Ndebele	LWAZI	No Sm.	**2596.81**	25.38	19.48	19.57	19.62	19.66	5.21
		GT	**2596.81**	19.41	15.76	16.14	16.26	16.35	
		AD	**2596.81**	18.83	14.53	14.61	14.56	14.70	
		WB	2686.11	18.99	14.32	14.34	14.38	14.39	
		UKN	**2596.81**	18.16	14.11	14.69	15.08	15.31	
		ND	**2596.81**	**17.85**	**13.69**	**13.70**	**13.74**	**13.75**	
		AS	**2596.81**	18.83	18.83	14.61	14.56	14.70	
	NCHLT	No Sm.	11414.60	103.06	70.80	66.59	66.57	66.57	3.53
		GT	11414.60	94.40	**55.49**	**50.28**	**50.27**	**50.27**	
		AD	11414.60	100.15	93.69	81.54	83.56	79.50	
		WB	11290.80	96.82	61.87	56.63	56.60	56.60	
		KN	**11245.50**	139.36	104.10	87.18	76.95	70.58	
		UKN	11414.60	**88.82**	61.83	54.24	55.87	57.21	
		ND	11414.60	89.08	58.59	53.79	53.77	53.77	
		AS	11414.60	100.15	100.15	81.54	83.56	79.50	

5.2.3. CMU-Cam Results

The language models developed with the third LM toolkit, CMU-Cam SLM, gave results that bear similarity with some of the results derived from statistical language models of the other two toolkits already presented. Using trigrams to hexagrams for reporting, Tables 5.5 and 5.6 show CMU-Cam baseline n-grams' performance. Unigram and bigram models are excluded because the CMU-Cam development setup either did not support their development with our data or resulted in unusually high PPL values such as *559230269.12*.

Of the four smoothing methods, WB, appeared to produce bettermodels for both languages. The lowest PPL values for trigram to hexagram models used WB smoothing for most corpora.

Table 5.5: Baseline PPL values for Pedi using CMU-Cam SLM toolkit

Language	Corpora	SM	N-gram PPL				OOV%
			3g	4g	5g	6g	
Pedi	LWAZI	AD	12.18	11.25	11.75	12.32	2.82
		GT	12.82	12.22	13.11	13.96	
		LD	13.26	12.62	13.33	14.16	
		WB	**11.50**	**10.06**	**9.98**	**10.04**	
	NCHLT	AD	18.85	13.34	12.18	12.17	1.12
		GT	378.02	268.94	246.81	251.75	
		LD	19.03	13.58	12.62	13.09	
		WB	**18.28**	**12.32**	**10.76**	**10.37**	

Table 5.6: Baseline PPL values for Ndebele using CMU-Cam SLM toolkit

Language	Corpora	SM	N-gram PPL				OOV%
			3g	4g	5g	6g	
Ndebele	LWAZI	AD	14.81	15.65	16.25	16.73	5.21
		GT	14.92	16.02	16.78	17.40	
		LD	18.50	20.63	22.37	23.80	
		WB	**14.78**	**15.14**	**15.32**	**15.42**	
	NCHLT	AD	12.03	10.65	11.00	11.47	3.53
		GT	14.08	12.26	12.69	13.17	
		LD	**11.52**	10.17	10.52	10.90	
		WB	11.79	**10.11**	**10.12**	**10.21**	

In analyzing N-gram models' performance, quadrigrams and pentagrams do relatively well to model the Pedi LWAZI text, whilst pentagrams and hexagrams best model the Pedi NCHLT text. The Ndebele text appears to be best modeled by trigrams and quadrigrams for LWAZI and NCHLT respectively.

5.3. Pooling Data

The pooling data experiment augmented the corpora by combining data from those languages that are most likely to share words and/or text with the Pedi and Ndebele languages, i.e., the Sotho and Nguni languages respectively. The text data size increased to help develop better estimates, the language domain of the text was widened and models were exposed to new text to learn. Data was pooled from the two

LWAZI and NCHLT corpora, and from similar corpora belonging to other languages in the Nguni and Sotho language groups.

5.3.1. HTK LM Results

For the Pedi LWAZI+NCHLT data combination, the lowest PPL result of 13.50 was achieved with the trigram model, whereas the pentagram model was the other higher n-gram model closest in terms of performance by a PPL result of 14.19 on the same test data. LM PPL results from this experiment are tabulated in Table 5.7.

Table 5.7: Pedi LWAZI+NCHLT data language models PPL performance

Experiment 2, LWAZI+NCHLT								Baseline		
Language	Corpora		N-gram PPL				OOV%	N-gram	PPL	OOV%
	Training	Testing	3g	4g	5g	6g				
Pedi	LWAZI+NCHLT	LWAZI	**13.95**	20.52	28.74	15.29	17.20	**4g**	**9.13**	**13.95**
		NCHLT	12.58	18.17	24.26	**12.35**	19.13	**4g**	**10.29**	**11.56**
		LWAZI+NCHLT	**13.50**	19.73	27.14	14.19	11.91	N/A		

On the Pedi LWAZI test text, the trigram model gave the lowest PPL of 13.95. On Pedi NCHLT test data, the hexagram model gave the lowest 12.35 PPL. The models trained on the combined Pedi language data have not outdone those trained on the individual LWAZI or NCHLT data as reported with the baseline n-grams.

While the Pedi LWAZI+NCHLT trigram model modelled better the Pedi LWAZI+NCHLT and LWAZI test data, the hexagram did for the Pedi NCHLT test text. The trigram also modelled the NCHLT test data well in that its PPL result differed with the hexagram's by 0.23 PPL.

Table 5.8 shows that for the Ndebele language, the higher n-gram models did not differ much in PPL performance amongst themselves as they did for the Pedi language. The model combinations, on each of the test data, do not give a PPL difference of more than 2; whilst the highest difference of 14.79 is realised for Pedi models when comparing the trigram and pentagram models for example on the LWAZI test data.

Table 5.8: Ndebele LWAZI+NCHLT data language models PPL performance

Experiment 2, LWAZI+NCHLT								Baseline		
Language	Corpora		N-gram PPL				OOV%	N-gram	PPL	OOV%
	Training	Testing	3g	4g	5g	6g				
Ndebele	LWAZI+NCHLT	LWAZI	**14.91**	15.04	15.24	15.39	2.67	**5g**	**8.75**	**4.39**
		NCHLT	32.08	30.20	30.38	**30.15**	3.52	**3g**	**30.46**	**2.26**
		LWAZI+NCHLT	23.85	**23.13**	23.40	23.43	2.13	N/A		

Although without significant differences among the LWAZI+NCHLT higher n-gram models: trigrams modelled Ndebele LWAZI test data better than other models (with PPL of 14.91), hexagrams outperformed on the NCHLT data (with PPL of 12.35), and quadrigrams gave better estimates on the combined LWAZI+NCHLT test data (with PPL of 23.13). We note that the LWAZI+NCHLT quadrigrams to hexagrams modelled the NCHLT data better than the baseline models.

The models trained with the Sotho class LWAZI text and tested on the Pedi LWAZI text had the trigram model giving the lowest PPL of 14.61 as reflected in Table 5.9. The lowest PPL from a model trained on the combined Sotho languages' NCHLT text is 13.80 from the quadrigram model tested on the Pedi NCHLT text. The Sotho class LWAZI+NCHLT quadrigram model gave the lowest 17.32 PPL on the Pedi LWAZI+NCHLT test data.

Table 5.9: Pedi and Sotho n-gram language models PPL performance

Experiment 2, Sotho								Baseline		
Corpora		N-gram PPL				OOV%	N-gram	PPL	OOV%
Training	Testing	3g	4g	5g	6g				
Sotho_LWAZI	Pedi_LWAZI	**14.61**	14.91	16.23	17.53	22.40	**4g**	**9.13**	**13.95**
Sotho_NCHLT	Pedi_NCHLT	13.97	**13.80**	14.54	15.81	19.05	**4g**	**10.29**	**11.56**
Sotho_LWAZI+NCHLT	Pedi_LWAZI+NCHLT	22.61	**17.32**	19.12	20.42	26.80	**3g**	**13.50**	**11.91**

A gain from the Sotho class data is observed from the LWAZI+NCHLT models. The Sotho LWAZI+NCHLT quadrigram and pentagram performed better (with PPLs of 17.32 and 19.12 respectively) than their corresponding baseline Pedi LWAZI+NCHLT quadrigram and pentagram (which had 19.73 and 27.14 PPL respectively as shown on Table 5.7).

Therefore, a single instance of benefit was observed from pooling data for the Pedi language in terms of PPL reduction. Except for LWAZI+NCHLT quadrigram and pentagram, both forms of augmentation (i.e., combining different corpora and combining corpora from related languages) resulted in models that gave higher/worse PPL values than the baseline models.

Nguni cluster language models performed as shown on Table 5.10 on the Ndebele test data. The only lowered PPL performance was seen from the Nguni class NCHLT quadrigram model which gave better estimates of the NCHLT test data with a PPL result of 40.09 compared to Ndebele NCHLT quadrigram's of 46.15 (see Table 5.2).

Table 5.10: Ndebele and Nguni n-gram language models PPL performance

Experiment 2, Nguni							**Baseline**		
Corpora		**N-gram PPL**				**OOV%**	**N-gram**	**PPL**	**OOV%**
Training	**Testing**	**3g**	**4g**	**5g**	**6g**				
Nguni_LWAZI	Ndebele_LWAZI	12.96	13.00	13.05	**12.79**	3.37	**5g**	**8.75**	**4.39**
Nguni_NCHLT	Ndebele_NCHLT	**37.21**	40.09	42.47	43.12	12.22	**3g**	**30.46**	**2.26**
Nguni_LWAZI+NCHLT	Ndebele_LWAZI+NC	33.37	**31.30**	33.37	34.11	13.66	**4g**	**23.13**	**2.13**

5.3.2. SRILM Results

No gain in terms of lowered PPL was deduced from combining the LWAZI and NCHLT corpora for training language models using the SRILM toolkit. The combination of the LWAZI and NCHLT data did not result in better models for modelling either corpus. Tables 5.11 and 5.12 show that no improvement in PPL was gained when testing the models trained on the augmented data on the same test text as the baseline models. In comparison, the baseline models gave better PPL results for both languages.

Table 5.11: Pedi LWAZI+NCHLT PPL results on different test data

Experiment 2, LWAZI+NCHLT									**Baseline**		
Language	**Corpora**		**SM**	**N-gram PPL**				**OOV%**	**N-gram**	**PPL**	**OOV%**
	Training	**Testing**		**3g**	**4g**	**5g**	**6g**				
Pedi	LWAZI+NCHLT	LWAZI	WB	31.29	23.83	22.90	22.85	2.16	**5g / 6g**	**10.99**	**2.82**
			UKN	24.16	17.94	**17.60**	17.97		**4g**	**11.20**	
		NCHLT	GT	22.31	15.39	13.76	**12.93**	1.06	**6g**	**12.65**	**1.12**
		LWAZI+NCHLT	WB	27.23	20.31	18.88	18.29	1.55	N/A		
			GT	25.91	18.68	17.24	**16.64**				
			UKN	26.19	19.31	18.29	17.88				

Table 5.12: Ndebele LWAZI+NCHLT PPL results on different test data

Experiment 2, LWAZI+NCHLT									Baseline		
Language	Corpora		SM	N-gram PPL				OOV%	N-gram	PPL	OOV%
	Training	Testing		3g	4g	5g	6g				
Ndebele	LWAZI+NCHLT	LWAZI	ND	24.79	**24.65**	24.73	24.74	4.55	**3g**	**13.69**	**5.21**
		NCHLT	GT	59.95	54.21	**54.20**	**54.20**	3.32	**5g / 6g**	**50.27**	**3.53**
		LWAZI+NCHLT	**ND**	42.08	**39.97**	40.01	40.02	3.86	**N/A**		
			GT	43.51	41.41	41.54	41.64				

We note a lowered OOV rates when testing the LWAZI+NCHLT models on the Pedi NCHLT data, Ndebele LWAZI data, and Ndebele NCHLT data

Although there are the benefits of the broadened domain and increased size of text, together with decreased OOV rates, the LWAZI+NCHLT results show that the merging of these two different corpora does not generally improve the performance of the language models in terms of lowering PPL. One factor to this may be the different approaches used in designing and developing the two corpora.

We also note from a smoothing methods perspective that the UKN models have shown improved performance when the two corpora were merged for the Pedi language.

In borrowing data from other related languages to create Sotho or Nguni class data, no improvements in PPL were observed for both Pedi and Ndebele texts. Tables 5.13 and 5.14 show PPL results from this experiment.

Table 5.13: Sotho class language models' PPL results

Experiment 2, Sotho								Baseline		
Corpora		SM	N-gram PPL				OOV%	N-gram	PPL	OOV%
Training	Testing		3g	4g	5g	6g				
Sotho_LWAZI	Pedi_LWAZI	WB	17.02	14.21	**13.93**	**13.93**	2.33	**5g / 6g**	**10.99**	**2.82**
Sotho_NCHLT	Pedi_NCHLT	GT	29.23	19.82	17.72	**16.66**	2.29	**6g**	**12.65**	**1.12**
Sotho_LWAZI+NCHLT	Pedi_LWAZI+NCHLT	WB	34.30	24.39	22.49	21.76	2.83	**6g**	**18.29**	**1.55**
		GT	34.52	23.97	21.98	**21.22**		**6g**	**16.64**	
		UKN	32.16	22.55	21.19	20.71		**6g**	**17.88**	

Table 5.14: Nguni class language models' PPL results

Experiment 2, Nguni								Baseline		
Corpora		SM	N-gram PPL				OOV%	N-gram	PPL	OOV%
Training	Testing		3g	4g	5g	6g				
Nguni_LWAZI	Ndebele_LWAZI	ND	21.10	21.10	21.17	21.18	4.32	3g	13.69	5.21
Nguni_NCHLT	Ndebele_NCHLT	GT	175.67	160.79	160.75	160.75	19.53	5g / 6g	50.27	3.53
Nguni_LWAZI+NCHLT	Ndebele_LWAZI+ NCHLT	ND	114.98	110.15	110.24	110.28	19.98	4g	39.97	3.86
		GT	117.59	113.30	113.55	113.76		4g	41.41	

The Sotho LWAZI models worsened the Pedi LWAZI performance by PPL results of up to 6 for the WB higher n-gram models. Increases between 4 and up to 16 in PPL were observed for the Sotho NCHLT GT models when compared to their baseline. The Sotho UKN LWAZI+NCHLT models outperformed the Sotho WB LWAZI+NCHLT models. This emergence of the UKN models in giving better performance for the merged corpora, suggests for Pedi and Sotho languages, an effort of investigating a fitting smoothing method whenever there is a change of corpus for LM purposes is necessary. The Pedi LWAZI OOV rate was lowered by the Sotho LWAZI models.

The Nguni class language models did not show any PPL improvements over the Ndebele baseline language models. Increases in PPL of more than 200% are realized when the performance of Nguni GT NCHLT higher n-gram models are compared to their baseline, and more than 100% increases are seen for the Nguni GT LWAZI+NCHLT models. It appears that the size of the text data, amongst other fac- tors, was a huge factor for the Nguni language models as the difference in average PPL is very large when compared to the NCHLT and LWAZI+NCHLT Ndebele models (approximately 113 and 71 respectively) and small when compared to the LWAZI Nde- bele models (≈7).

5.3.3. CMU-Cam Results

No performance gains from pooling the data under the CMU-Cam SLM toolkit were observed as was generally the case for the other two toolkits. From all the combinations, the baseline models had better performance. Cross examining the PPL rates however, the rates of the Sotho models are generally not that far from those of the Pedi language, with PPL differences of at least 1.97 and at most 5.44. This may suggest that the added data from the other languages does not cause much confusion to

the models, thus not degrading their quality. This also remarks on the closeness/relatedness of the text of the concerned Sotho class of languages. Wide PPL differences (at least 4.79 and at most 16.64) were observed when comparing the Nguni class and Ndebele language models. Tables 5.15 and 5.16 reflect these observations.

Table 5.15: Pedi data pooling experiment, CMU-Cam SLM

Experiment 2, LWAZI+NCHLT and Sotho								Baseline		
Corpora		SM	N-gram PPL				OOV%	N-gram	PPL	OOV%
Training	Testing		3g	4g	5g	6g				
Sepedi_LWAZI+NCHL	Sepedi _LWAZI+NCHLT	WB	19.71	13.71	12.40	**12.13**	1.55	N/A		
Sotho_LWAZI	Sepedi_LWAZI		14.74	12.26	**12.08**	12.14	2.33	**5g**	**9.98**	**2.82**
Sotho_NCHLT	Sepedi_NCHLT		23.72	15.11	12.87	**12.34**	2.29	**6g**	**10.37**	**1.12**
Sotho_LWAZI+NCHLT	Sepedi _LWAZI+NCHLT		25.03	16.44	14.48	**14.08**	2.83	**6g**	**12.13**	**1.55**

Table 5.16: Ndebele data pooling experiment, CMU-Cam SLM

Experiment 2, LWAZI+NCHLT and Nguni								Baseline		
Corpora		SM	N-gram PPL				OOV%	N-gram	PPL	OOV%
Training	Testing		3g	4g	5g	6g				
Ndebele_LWAZI+NCHLT	Ndebele_LWAZI+NCHLT	WB	16.26	**14.99**	15.07	15.19	3.86	N/A		
Nguni_LWAZI	Ndebele_LWAZI		**21.50**	21.96	22.25	22.38	4.32	**3g**	**14.78**	5.21
Nguni_NCHLT	Ndebele_NCHLT		26.19	16.38	15.18	**15.04**	19.53	**4g**	**10.11**	3.53
Nguni_LWAZI+NCHLT	Ndebele_LWAZI+NCHLT		32.90	23.58	22.11	**21.89**	19.98	**4g**	**14.99**	3.86

5.4. Higher-order N-grams

The higher-order n-grams experiment sought to estimate language models up to the *(random selected)* order 20 to determine the performance trend of the models as they increase in n-gram order.

5.4.1. HTK LM Results

There was not much that could be deduced from this experiment (i.e., developing and estimating with n-grams beyond hexagrams) when using the HTK LM toolkit. This is because the toolkit supports estimation of n-grams up to order 6 (i.e., n = 6). Since testing is allowed beyond n-gram order 6 and up to order 10, the back-off method of estimation - that uses estimates from lower n-grams when not present in higher n-grams - was employed to test and give estimates at the n-gram order levels of 7-10 (i.e., heptagrams to decagrams). The results indicate rising or stable levels of PPL performance beyond hexagrams, this is expected since the testing model is not of the

required order, i.e., it is not of the same n-gram order as the word sequences that are tested.

Figure 5.1 shows the PPL performances of the Pedi and Sotho models forthe n-gram orders 1 to 10. The performances for Ndebele and Nguni n-gram models are projected on Figure 5.2.

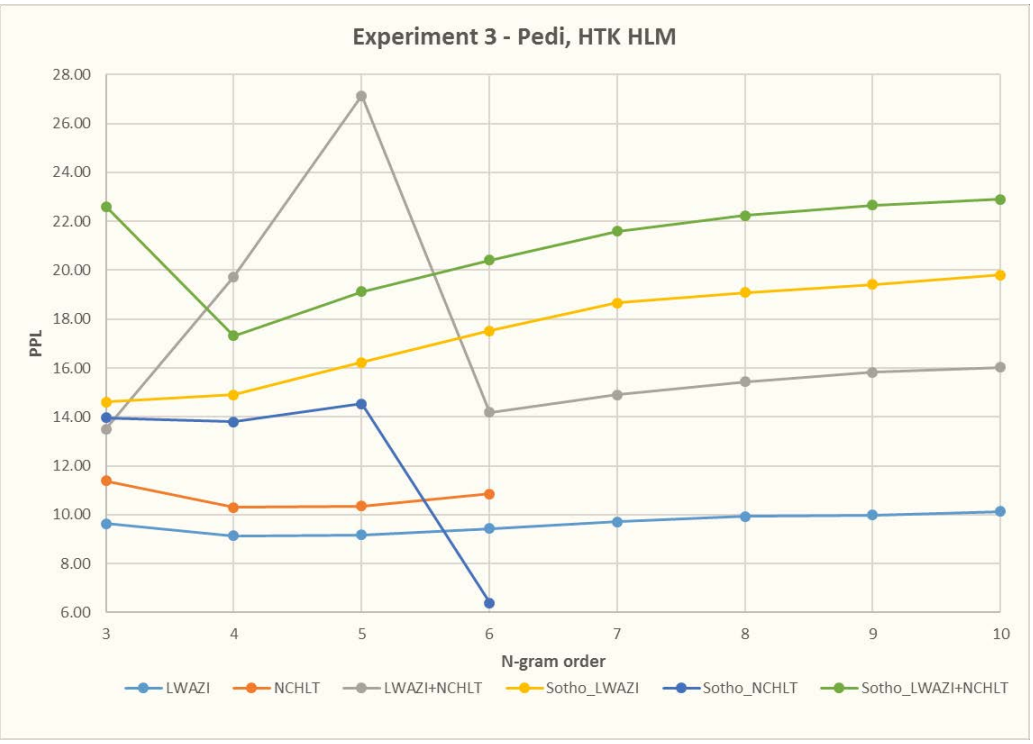

Figure 5.1: Pedi and Sotho higher-order language models' PPL results, HTK LM

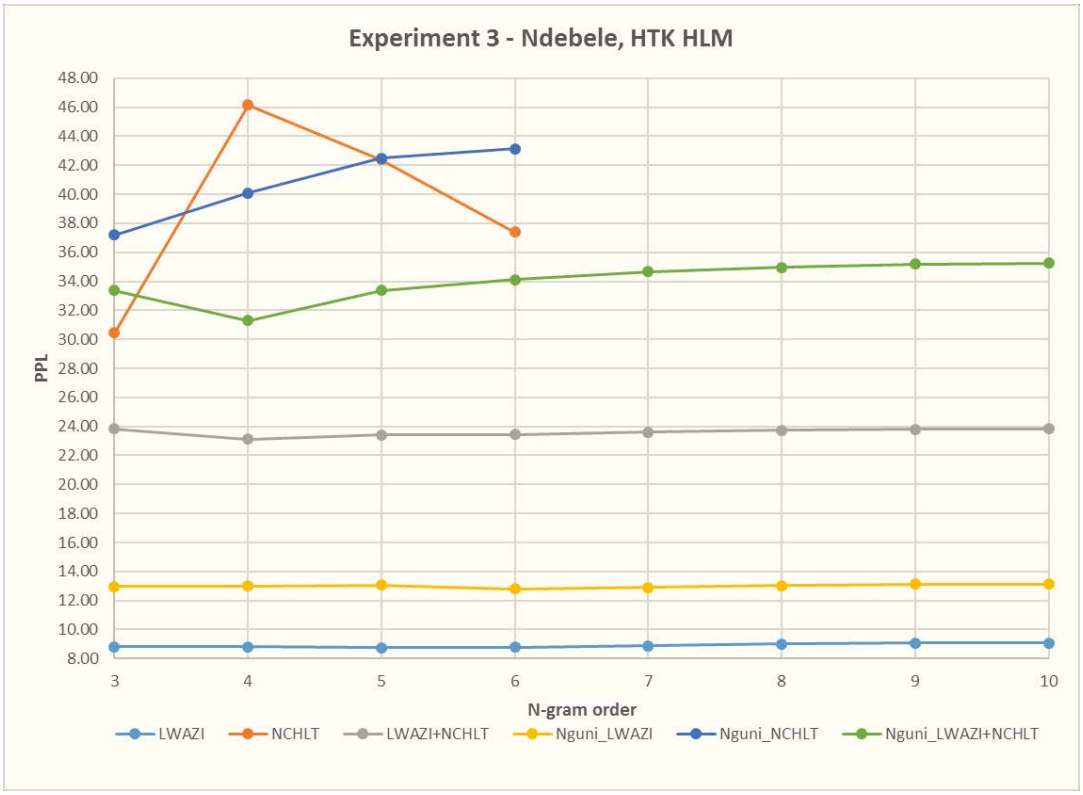

Figure 5.2: Ndebele and Nguni higher-order language models' PPL results

5.4.2. SRILM Results

SRILM allows, according to the manual description of the *ngram-count* [12] tool, n-gram estimation up to order 9 although not restricting estimation when commands specify orders beyond 9.

The results shown on Figures 5.3 and 5.4 indicate that high-order n-grams above the standard trigrams appear to be suitable for modelling both the disjunctive and conjunctive Pedi and Ndebele languages. The high order appears to be bounded to n = 6, after which the performance remains stable throughout or stable up to n-gram order 9 and then deteriorates a bit thereafter.

For the Pedi language, low PPL n-gram language models were pentagrams and hexagrams. Trigrams and quadrigrams were observed to better model the Ndebele text.

[12] *http://www.speech.sri.com/projects/srilm/manpages/ngram-count.1.html*

An exception, however, and the only PPL gain observable beyond hexagrams out of this experiment was when the Sotho LWAZI 10-gram (decagram) best modeled the Pedi LWAZI data.

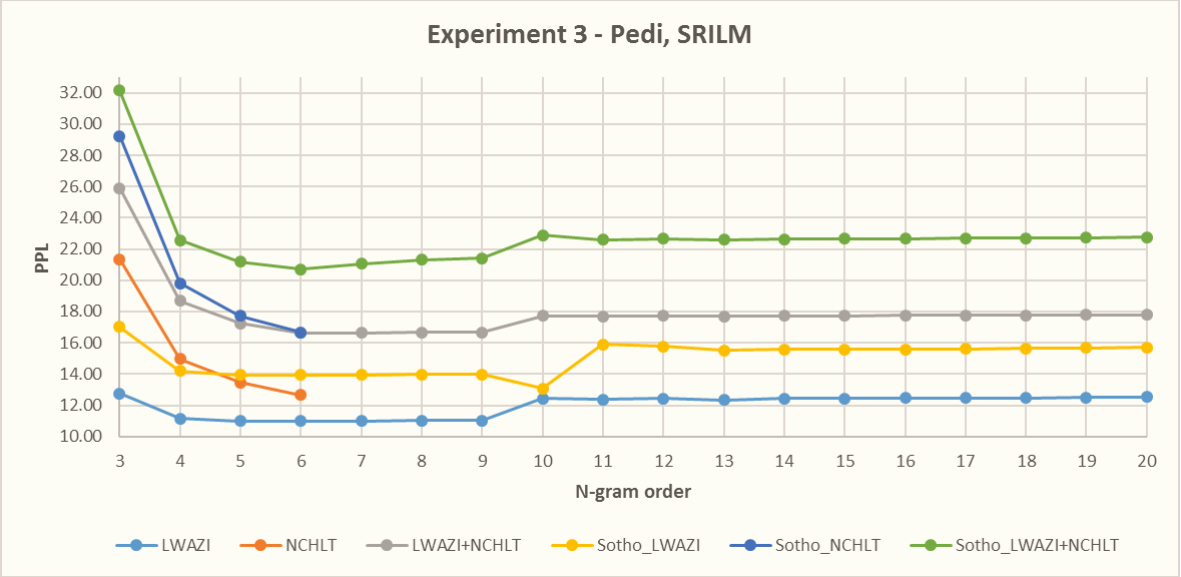

Figure 5.3: Pedi and Sotho higher-order language models' PPL results, SRILM

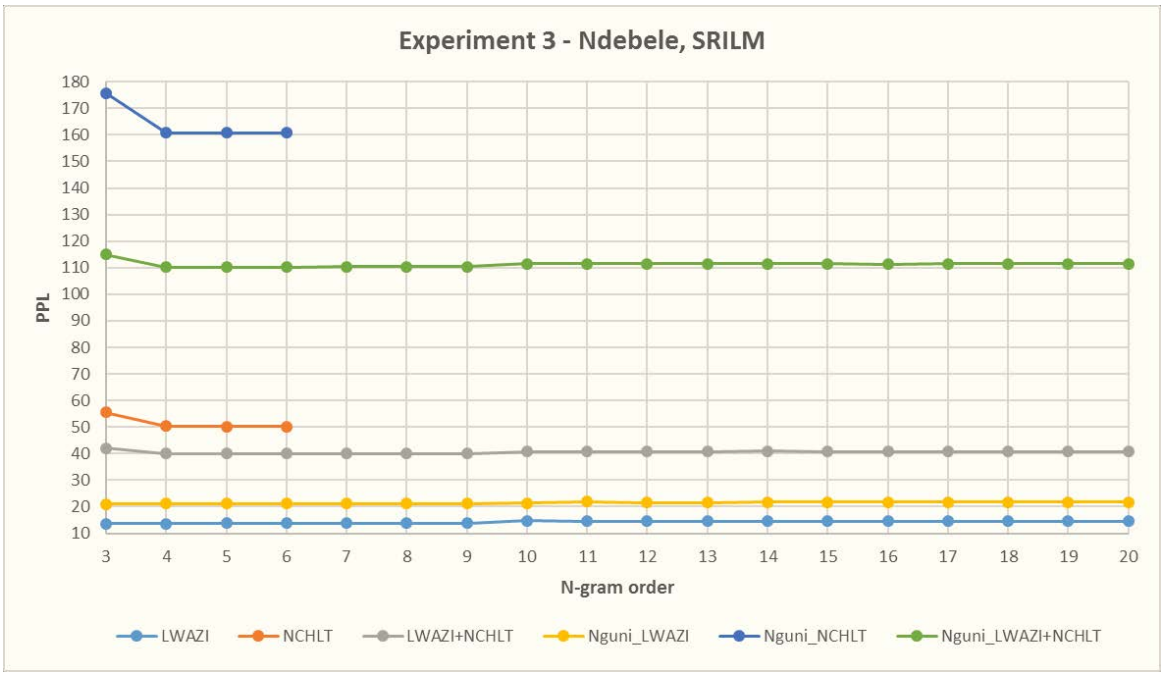

Figure 5.4: Ndebele and Nguni higher-order language models' PPL results, SRILM

5.4.3. CMU-Cam Results

The third experiment revealed that CMU-Cam SLM supports LM training up to n-gram order 6 like HTK LM, and LM testing up to n-gram order 9. The results, as shown on Figure 5.5 and Figure 5.6 indicate that when using the CMU-Cam SLM toolkit for LM development: pentagrams and hexagrams are sufficient to model the disjunctive Pedi text, and trigrams and quadrigrams for modelling the conjunctive Ndebele text.

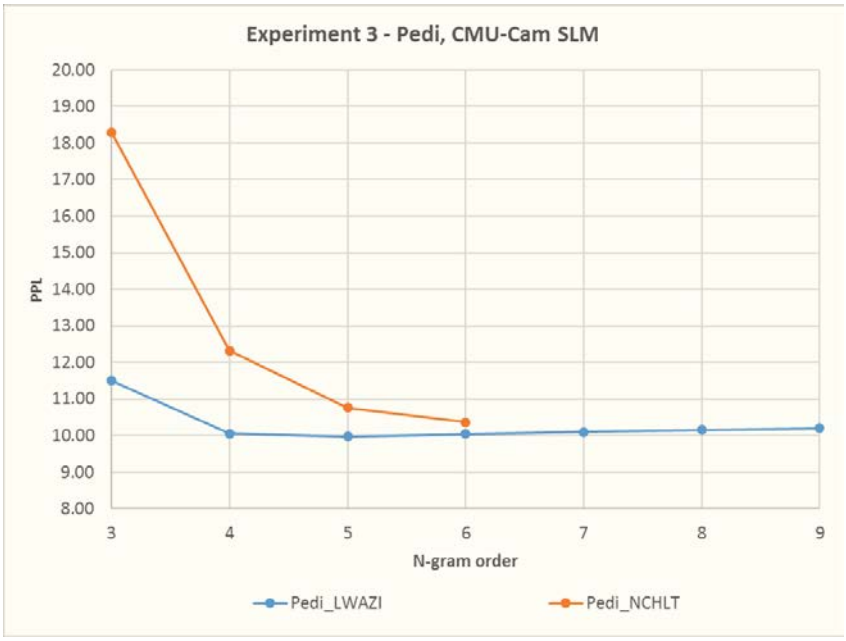

Figure 5.5: Higher-order n-gram performance for Pedi, CMU-Cam SLM.

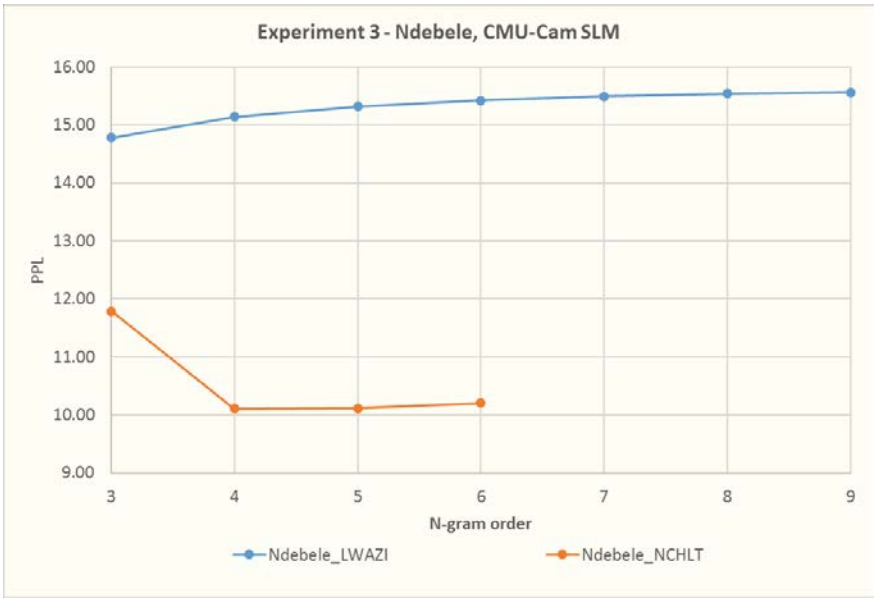

Figure 5.6: Higher-order n-gram performance for Ndebele, CMU-Cam SLM.

5.5. Interpolation

In further attempts to improve on the developed language models, this experiment augmented the developed models with interpolation: during training and during testing. By default, n-gram language models are not trained to include interpolation for SRILM LM estimation. The interpolation technique was enabled and language models were retrained using the development procedure of the baseline models. Best performing models from previous experiments, trained with interpolation and not, were then mixed for the during-testing interpolation.

5.5.1. Interpolation at Training

When interpolation was enabled during training, LM performance remained the same for some smoothing methods and noticeably improved for others. Thus, interpolation during training had effect in the quality of some of the language models. The OOV rates remained unchanged when compared to the baselines.

In the case of Ndebele LWAZI text, training with interpolation enabled did not change the performance of baseline GT, AD, KN, ND, and AS smoothed models. Table 5.17 shows the (positive) changes in PPL performance for the WB and UKN smoothed language models. Interestingly, interpolated WB models emerged as better modelling the LWAZI text – better than baseline ND models (interpolated or not). The difference in performance between the two sets of models is not negligible though, interpolated WB higher n-grams gave an average PPL performance of 13.42 and 13.72 was given by the ND higher n-grams (thus an average difference of 0.30). The interpolated WB quadrigram model performed with the lowest 13.30 PPL.

Table 5.17: Ndebele n-grams trained with interpolation

Experiment 4, Interpolation enabled at Training							Baseline		
Toolkit & Language	Corpus	Smoothing Method	PPL				N-gram	PPL	OOV%
			3g	4g	5g	6g			
SRILM, Ndebele	LWAZI	ND	13.69	13.70	13.74	13.75	**3g**	**13.69**	**5.21**
		WB	13.41	**13.30**	13.44	13.56	**3g**	**14.32**	
		UKN	13.58	13.89	14.05	14.18	**3g**	**14.11**	
	NCHLT	GT	55.49	50.28	50.27	50.27	**5g / 6g**	**50.27**	**3.53**
		WB	56.96	50.80	50.15	50.15	**5g / 6g**	**56.60**	
		KN	77.02	66.84	61.06	58.74	**6g**	**70.58**	
		UKN	58.37	**50.14**	51.22	51.83	**4g**	**54.24**	

Interpolated WB, KN, and UKN smoothed models were observed with changes in PPL performance when compared to their corresponding baseline models for the Ndebele NCHLT data. No changes in performance were shown by interpolated GT, AD, ND, and AS smoothed models. In terms of higher n-grams average performance, interpolated/baseline GT smoothed models remained better in estimation by 51.58 PPL, followed by interpolated WB smoothed models with 52.01 PPL. However, interpolated UKN quadrigram model the lowest PPL of 50.14, followed by interpolated WB pentagram and hexagram models with PPLs of 50.15.

Table 5.18 helps us analyse the effect of interpolation at training for Pedi language models. Like the Ndebele language, interpolation improved the performance of the WB and UKN smoothed models for the LWAZI data; and WB, KN, and UKN smoothed models for the NCHLT data. The performance for the interpolated GT, AD, ND, and AS smoothed models was the same as the baseline models'.

Table 5.18: Pedi n-grams trained with interpolation

Experiment 4, Interpolation enabled at Training							Baseline		
Toolkit & Language	Corpus	Smoothing Method	PPL				N-gram	PPL	OOV%
			3g	4g	5g	6g			
SRILM, Pedi	LWAZI	**WB**	12.38	10.64	**10.54**	10.68	**5g / 6g**	**10.99**	**2.82**
		UKN	11.80	10.61	10.72	10.89	**4g**	**11.20**	
	NCHLT	GT	21.35	14.96	13.46	**12.65**	**6g**	**12.65**	**1.12**
		WB	25.63	18.97	17.28	16.38	**6g**	**17.36**	
		KN	28.59	21.40	19.42	17.57	**6g**	**19.83**	
		UKN	25.69	18.71	17.04	15.88	**6g**	**16.86**	

The WB and UKN higher n-grams continue complementing each other in better modelling the Pedi LWAZI data.

For the Pedi NCHLT data, interpolated GT higher n-grams maintained the relatively best performance although with same PPL results as in the non-interpolated models case. The WB, KN, and UKN models did improve in performance but not to the level of toppling the GT models.

5.5.2. Interpolation at Testing

Interpolation was further explored, this time at the testing phase and for selected models. Using SRILM's *"–mix-lm"* tool, best performing language models from the previous experiments were mixed/interpolated with the hope that a better performing model would result out of their combination. Models that did not implement interpolation during training from experiment setups 1 through 3, and the models that implemented interpolation during training in experiment setup 4 were used for the mixing experiment. Furthermore, having learnt that certain n-grams do well in modelling either the Pedi or Ndebele language (i.e., mostly pentagrams and hexagrams for the former and trigrams and quadrigrams for the latter), non-interpolated and interpolated pentagrams and hexagrams were mixed for the Pedi language, and trigrams and quadrigrams for the Ndebele language. For each corpus, the test data was used to compute the required mixture weights for the mixing exercise.

A survey of best performing models from the first three experiments revealed the models in Table 5.19 for the different language corpora. These models were mixed and their PPL performance recorded in the "Mixed_LM" column. In comparison, the mixing led to relatively better models (although the PPL differences are negligible) compared to at least one of the models in the mixture. The performance of the mixed model was better than that of the two individual models that were mixed for the Ndebele language and Nguni class LWAZI texts only.

Table 5.19: Mixed best language models

Experiment 4, Interpolation at Testing (models trained without interpolation)							
Toolkit	**Corpus**	**LM1**	**PPL**	**LM2**	**PPL**	**Mixture Weight**	**Mixed_LM**
SRILM	Ndebele_LWAZI	**3g_ND**	13.69	**3g_UKN**	14.11	0.679969	**13.26**
	Ndebele_NCHLT	**5g_GT**	50.27	**4g_ND**	53.79	0.672540	52.77
	Ndebele_LWAZI+ NCHLT	**4g_ND**	39.97	**4g_UKN**	40.35	0.933360	41.83
	Nguni_LWAZI	**3g_ND**	21.10	**3g_KN**	21.28	0.587611	**20.09**
	Nguni_NCHLT	**5g_GT**	160.75	**4g_UKN**	167.98	0.926465	169.31
	Nguni_LWAZI+N CHLT	**4g_ND**	110.15	**4g_WB**	111.69	0.712472	113.51
	Pedi_LWAZI	**5g_WB**	10.99	**5g_ND**	11.06	0.383562	12.50
	Pedi_NCHLT	**6g_GT**	12.65	**6g_UKN**	16.86	0.972926	20.76
	Pedi_LWAZI+NC HLT	**6g_GT**	16.64	**6g_UKN**	17.88	0.898434	23.07
	Sotho_LWAZI	**5g_WB**	13.93	**4g_UKN**	14.10	0.971018	17.07
	Sotho_NCHLT	**6g_GT**	16.66	**6g_UKN**	20.02	0.967848	28.27
	Sotho_LWAZI+NC HLT	**6g_UKN**	20.71	**6g_GT**	21.22	0.114647	30.35

The second part of the mixing experiment combined pentagrams and hexagrams for the Pedi language and trigrams and quadrigrams for the Ndebele language. No benefit in terms of lowered PPL is noted for the Pedi pentagram and hexagram model combinations as shown by Table 5.20.

Table 5.20: Mixed Pedi pentagrams and hexagrams

Experiment 4, Interpolation at Testing (penta-hexagrams trained without interpolation)						
Toolkit	**Language & Corpus**	**LM1**	**PPL**	**LM2**	**PPL**	**Mixed_LM PPL**
SRILM	Pedi_LWAZI	**5g_WB**	10.99	**6g_WB**	10.99	12.76
		5g_ND	11.06	**6g_ND**	11.06	12.74
	Pedi_NCHLT	**5g_GT**	13.46	**6g_GT**	12.65	21.35
		5g_UKN	17.92	**6g_UKN**	16.86	62.26
	Pedi_LWAZI+NCHLT	**5g_GT**	17.24	**6g_GT**	16.64	25.91
		5g_UKN	18.29	**6g_UKN**	17.88	65.34
	Sotho_LWAZI	**5g_WB**	13.93	**6g_WB**	13.93	17.02
		5g_UKN	14.44	**6g_UKN**	14.87	51.25
	Sotho_NCHLT	**5g_GT**	17.72	**6g_GT**	16.66	29.23
		5g_UKN	21.28	**6g_UKN**	20.02	77.59
	Sotho_LWAZI+NCHLT	**5g_UKN**	21.19	**6g_UKN**	20.71	79.65
		5g_GT	21.98	**6g_GT**	21.22	34.52

Although there is also no gain in terms of lowered PPL values for the Ndebele trigram and quadrigram models as reflected by Table 5.21, some of the mixed models gave performances similar to individual models of the combination.

Table 5.21: Mixed Ndebele trigrams and quadrigrams

Experiment 4, Interpolation at Testing (tri-quadrigrams trained without interpolation)						
Toolkit	**Language & Corpus**	**LM1**	**PPL**	**LM2**	**PPL**	**Mixed_LM PPL**
SRILM	Ndebele_LWAZI	**3g_ND**	13.69	**4g_ND**	13.70	**13.69**
		3g_UKN	14.11	**4g_UKN**	14.69	14.12
	Ndebele_NCHLT	**3g_GT**	55.49	**4g_GT**	50.28	55.49
		3g_ND	58.59	**4g_ND**	53.79	58.59
	Ndebele_LWAZI+NCHLT	**3g_ND**	42.08	**4g_ND**	39.97	42.08
		3g_UKN	42.37	**4g_UKN**	40.35	42.41
	Nguni_LWAZI	**3g_ND**	21.10	**4g_ND**	21.10	**21.10**
		3g_WB	21.94	**4g_WB**	21.94	**21.94**
	Nguni_NCHLT	**3g_GT**	175.67	**4g_GT**	160.79	175.65
		3g_UKN	184.24	**4g_UKN**	167.98	184.43
	Nguni_LWAZI+NCHLT	**3g_ND**	114.98	**4g_ND**	110.15	116.77
		3g_WB	116.77	**4g_WB**	111.69	114.98

The language models trained with interpolation were also mixed. Mainly, interpolated WB and UKN smoothed models were found to best model the Ndebele language and were thus mixed; and interpolated WB, GT, and UKN models were mixed for the Pedi

language. As shown by Table 5.22, the only comparatively lowered PPL value was observed for the Ndebele LWAZI mixed smoothed model.

Table 5.22: Mixed models trained with interpolation

Experiment 4, Interpolation at Testing (tri-quadrigrams trained with interpolation)						
Toolkit	**Language & Corpus**	**LM1**	**PPL**	**LM2**	**PPL**	**Mixed_LM PPL**
SRILM	**Best Ndebele LMs**					
	Ndebele_LWAZI	**4g_WB**	13.30	**3g_UKN**	13.58	**13.16**
	Ndebele_NCHLT	**4g_UKN**	50.14	**6g_WB**	50.15	57.04
	Best Pedi LMs					
	Pedi_LWAZI	**5g_WB**	10.54	**4g_UKN**	10.61	12.39
	Pedi_NCHLT	**6g_GT**	12.65	**6g_UKN**	15.88	20.83
	Tri-quadrigrams trained with interpolation					
	Ndebele_LWAZI	**3g_WB**	13.41	**4g_WB**	13.30	13.41
		3g_UKN	13.58	**4g_UKN**	13.89	13.60
	Ndebele_NCHLT	**3g_UKN**	58.37	**4g_UKN**	50.14	58.42
		3g_WB	56.96	**4g_WB**	50.80	56.96
	Penta-hexagrams trained with interpolation					
	Pedi_LWAZI	**5g_WB**	10.54	**6g_WB**	10.68	12.38
		5g_UKN	10.72	**6g_UKN**	10.89	33.04
	Pedi_NCHLT	**5g_GT**	13.46	**6g_GT**	12.65	21.35
		5g_UKN	17.04	**6g_UKN**	15.88	54.42

5.6. Discussion of Results

This section discusses the analysed results and findings.

Selected findings from this study confirmed some standard LM practices to be applicable to the development of language models even for the Pedi and Ndebele languages. A few findings are noted.

Data preparation is a crucial LM step that adequately prepares the text data for efficient LM development. *Smoothing* of language models yielded models that improved

modelling quality. *Higher-order n-gram language models* (i.e., trigrams to hexagrams) were more efficient than lower-order n-grams (unigrams and bigrams) in LM estimation. *Interpolation* can improve the LM estimation process to arrive at a reasonable estimation for a likely word sequence due to the combination of models and therefore combined modelling efficiency and knowledge.

That *trigrams* and *quadrigrams* model Ndebele text better and *quadrigrams* to *hexagrams* model better the Pedi text could be stemming from the average length of word sequences and sentences of the two languages as a consequence of their *conjunctive and disjunctive* writing systems respectively. The Ndebele language word sequences are usually short, whilst the Pedi language word sequences are usually long. This observation is confirmed when one analyses the text of the concerned languages as could be deduced, for example, in Tables 1.1, 4.3, 4.4, and 4.5.

Especially as realised using the widely adopted *SRILM toolkit*, *different smoothing methods* cope differently with the two writing systems. ND and GT seem to naturally cope well with the conjunctive writing of the Ndebele language, whilst UKN and WB appeared to cope better with the disjunctive Pedi language writing

Some of the findings did not confirm existing literature assertions. For example, *trigram n-grams* and the *KN smoothing method* are recommended for standard LM development. In this study, higher-order n-grams beyond trigrams (i.e., quadrigrams to hexagrams) were most performing and thus recommended; and other smoothing methods such as GT, WB, ND, and UKN smoothing led to better language models. As per the observations, KN smoothed models either performed with higher PPL values or their estimation was not supported on the development setup. Trigram n-grams did give best performances in some of the LM estimation cases, especially for the Ndebele language, but such performances could not be observed generally as expected.

The *pooling data experiment* did not yield improved language models. There were positive expectations behind the design of the experiment given that such a design would significantly increase the size of the text data, widen the text domain, and comparatively reduce OOV rates. Also, the two classes of languages (Nguni and Sotho classes) are in speech or when spoken closely related to an extent that you need only

be fluent or master one of them too ably converse with speakers of the other languages belonging to the same group. Whilst this holds true from the speech point of view, a contrast is observed on the orthography of the language. The pooling data exercise appears to have brought confusion (in terms of unfamiliar words and sentences) to the language models and their performances were thus poor. This observation was more elaborated for the Nguni class of languages than for the Sotho class of languages. Therefore, although the clustered languages are similarly spoken and written, the individual languages – the pooling experiments reveal - use significantly different word forms that cannot be simply combined in attempts of increasing training data.

In order to not carelessly discount or disregard any hidden or potential benefits resulting from pooling data from the clustered languages, an interesting endeavour would be to interpolate pooled-data models with non-pooled-data models to realise the inherent efficacy the former may add to the latter's, and vice versa.

Chapter 6: Conclusions, Summary and Future Work

In this chapter, conclusions from the results and findings are drawn in Section 6.1. The study is summarised in Section 6.2. Section 6.3 sets potential targets of future work beyond this study.

6.1. Conclusions

The following conclusions are drawn from the results and findings thus far reported.

- Specially prepared text produces better performing language models. Using the LWAZI and NCHLT text corpora, the text preparation process should include removal of all sentence text markers and annotation tags, punctuations and all character symbols that are not part of the words.
    - For development with HTK LM, the normalised text should be appended with start- and end-of-sentence markers (e.g., <s> and </s>).
    - Changing the casing of the text streams does not affect the performance of the language models. The performance is only degraded when the casing is mixed.
- Trigrams to hexagrams better estimate both the Ndebele and Pedi texts across all three toolkits.
    - Using the HTK LM toolkit, quadrigrams estimate better the Pedi text, and trigrams the Ndebele text.
    - Developing with the SRILM toolkit: better models for the Pedi text were WB pentagrams and hexagrams for the LWAZI text; and GT hexagrams for the NCHLT text. For the Ndebele text, ND trigrams model the LWAZI text well, and GT pentagrams the NCHLT text.
    - On the CMU-Cam SLM toolkit, best n-gram performance is derived by developing WB quadrigrams for the Pedi LWAZI text, WB pentagrams for the Pedi NCHLT text, WB trigrams for the Ndebele LWAZI text, and WB quadrigrams for the Ndebele NCHLT text.

- Pooled LWAZI and NCHLT data does not necessarily yield better performing language models. However, there were a few exceptions to this observation from models developed using the HTK LM toolkit:
    - Ndebele LWAZI+NCHLT quadrigrams to hexagrams gave best PPL performance on the Ndebele NCHLT test data when compared to their Ndebele NCHLT counterparts;
    - Sotho quadrigrams and pentagrams outperformed Pedi quadrigrams and pentagrams in modelling the Pedi LWAZI+NCHLT text;
    - Nguni quadrigrams gave better estimation of the Ndebele NCHLT text than Ndebele quadrigrams.
- Varied smoothing yields better models.
    - Using the SRILM toolkit, best models were smoothed with UKN and WB for the Pedi LWAZI text, GT for the Pedi NCHLT text, ND for the Ndebele LWAZI text, and GT for the Ndebele NCHLT text.
    - Working with the CMU-Cam SLM toolkit, AD smoothed unigrams and bigrams together with WB smoothed trigrams to hexagrams gave better PPL results for texts belonging to the two languages.
- When there is improvement from higher-order n-gram estimation of the test data, n-gram performance does not significantly improve beyond hexagrams.
- Language models trained with the interpolation techniques enabled yielded improved PPL results.
- In general, mixing/interpolating models at the testing phase using SRILM does not produce a better language model. Slight exceptional PPL improvements from mixed Ndebele and Nguni LWAZI best performing models were found.
- Lower PPL values were found associated with language models of the Pedi text, whilst Ndebele text models gave relatively high PPL values.
- The disjunctiveness of Pedi writing leads to longer sentences or word sequences, thus more text in terms the number of words. The conjunctive writing of the Ndebele language leads to relatively small text for the training of language models. As found in the preceding observation, the more the LM text data, the better the estimation of the developed languages models.
- The length of the modelling unit is relatively longer for the Ndebele language and shorter for the Pedi language, LM development coped relatively better with the shorter modelling unit of the Pedi language.

With the LM results reported and analysed thus far, it can be concluded that the orthography of the two languages does have effect on the quality of language models developed on their text.

For LM of texts belonging to the two languages, a differing LM approach is thus recommended given their differing writing systems. The following recommendations are made as part of LM development.

- Firstly, specially prepare and clean the text before LM development – paying attention to within sentence text markers and annotation tags that may incorrectly form part of LM sentences, word sequences, and n-gram contexts.
- Secondly, enable interpolation during LM training.
- Thirdly, develop with quadrigrams to hexagrams for Pedi texts, and trigrams and quadrigrams for the Ndebele texts.
- Lastly, and not the least, investigate the efficient smoothing method for different text sizes (e.g., LWAZI is smaller than NCHLT in terms of the amount of text), text domain (e.g., language subject and/or topics of the LWAZI and NCHLT text are different), and change in languages (e.g., in the pooling data experiments, other languages were incorporated).
    - In particular: GT, WB, and UKN smoothing methods appear to well smooth the Pedi texts; and GT, ND and WB methods better smooth the Ndebele texts.

6.2. Summary

This study proposed an investigation in LM for speech recognition. In further determining the influence that the unique and inherent nature of a language has on the language models created for text of that language, we undertook to study the influence of the orthography of selected languages. The unique conjunctive and disjunctive writ- ing systems of the South African Ndebele and Pedi languages were studied.

Ultimately, we sought to contribute an in-depth analysis and study of LM work for the selected under-resourced languages. This was earmarked to become a significant addition to the body of HLT research in the context of under-resourced languages of the world.

LM as art of determining probabilities and/or likelihood of word sequences is fundamental in the workings of a speech recognition system as it helps quantify which word sequence, amongst all estimated, the system outputs as its best estimate to the input speech signal. Having pre-processed the text data: language models are trained on the training set and then evaluated on a separate testing set. The quality of the models could be evaluated using the PPL metric, where the model with the lowest PPL score is most accurate in the modeling task. There exist various methods and techniques for developing quality and low PPL language models such as smoothing, factored language models, neural network LM, and approximate inference LM.

This study used the data from the LWAZI and NCHLT speech corpora projects. The LM techniques that were implemented included: word-based LM, various LM smoothing methods, LM interpolation and higher-order n-gram LM. The toolkits used in development were: HTK LM, SRILM, and CMU-Cam SLM toolkits. Four main experiments were conducted under the designs: baseline LM, pooled data, higher-order n-grams, and n-gram interpolation.

With the LM results reported and analysed thus far, it can be concluded that the orthography of the two languages does have effect on the quality of language models developed on their text.

For LM of texts belonging to the two languages, a differing LM approach is recommended given their differing writing systems. The following recommendations are made as part of LM development. Firstly, specially prepare and clean the text before LM development – paying attention to within sentence text markers and annotation tags that may incorrectly form part of sentences, word sequences, and n-gram contexts. Secondly, enable interpolation during LM training. Thirdly, develop with quadrigrams to hexagrams for Pedi texts, and trigrams and quadrigrams for the Ndebele

texts. Lastly, investigate the efficient smoothing method for different text sizes, text domain, and change in languages.

Parts of the work of this study were shared with the research community in presentations and publications as referred to by [68], [69], [70], [71], and [72].

6.3. Future Work

Possible next phase(s) of this study may partly be guided by the questions raised from the findings. Some of these questions were the following: 1) what causes the LM development setup not to support estimation of KN smoothed models in certain experiments? 2) Why could no benefits be derived, in general, from the models mixing experiment? 3) What causes the big differences in PPL performance for the Nguni languages?

On-going research in the field of LM has led to new developments. Such as: factored language models, neural network language models, investigating sub-word language models for under-resourced languages, and wider morphology-based language models for morphology rich languages. Adaptation and interpolation of these models is an interesting LM research perspective. These developments could be explored even for languages of this study.

What will give more significance to the work of this study would be the eventual incorporation and the evaluation thereafter of the developed language models in an ASR application system developed for the concerned languages.

References

[1] D. Jurafsky and J. H. Martin, Speech and Language Processing, 2nd ed., Upper Saddle River, New Jersey 07458: Prentice-Hall, 2009.

[2] J. R. Bellegarda, “Statistical language model adaptation: review and perspectives,” *Speech Communication,* vol. 42, no. 1, pp. 93-108, 2004.

[3] L. Besacier, E. Barnard, A. Karpov and T. Schultz, “Automatic speech recognition for under-resourced languages: A survey,” *Speech Communication,* vol. 56, no. 1, pp. 85-100, 2014.

[4] R. H. Gouws , U. Heid , W. Schweickard and H. E. Wiegand , An International Encyclopedia of Lexicography, Supplementary Volume: Recent Developments with Focus on Electronic and Computational Lexicography, Berlin, Boston: De Gruyter Mouton, 2013.

[5] J. Jones, S. E. Bosch, L. Pretorius and D. Prinsloo, “Development of reusable resources for Human Language Technologies (HLT) applications: practice and experience,” *South African Journal of African Languages,* vol. 25, no. 2, pp. 141-159, 2005.

[6] E. Eiselen and M. Puttkammer, “Developing text resources for ten South African languages,” in *Language Resources and Evaluation Conference*, pp. 3698-3703, 2014.

[7] E. Taljard and S. E. Bosch, “A Comparison of Approaches to Word Class Tagging: Disjunctively vs. Conjunctively Written Bantu Languages,” *Nordic Journal of African Studies,* vol. 15, no. 4, pp. 428-442, 2006.

[8] S. Young, E. Gunnar, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland, The HTK Book (for HTK Version 3.4), Cambridge: Cambridge University Engineering Department, 2009.

[9] I. Oparin, “Language models for automatic speech recognition of Inflectional languages,” PhD thesis, University of West Bohemia, Pilsen, Czech Republic, 2008.

[10] D. Falavigna and R. Gretter, “Focusing language models for automatic speech recognition,” in *The 9th International Workshop on Spoken Language Translation*, pp. 171-178, 2012.

[11] A. Karpov, K. Markov, I. Kipyatkova, D. Vazhenina and A. Ronzhin, “Large vocabulary Russian speech recognition using syntactico-statistical language modeling,” *Speech Communication,* vol. 56, pp. 213-228, 2014.

[12] C. Martins, A. Teixeira and J. Neto, “Language models in automatic speech recognition,” *Electrónica e Telecomunicações,* vol. 4, no. 4, pp. 428-432, 2005.

[13] F. S. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Technical Report TR-10-98, Computer Science Group, Harvard University, 1998.

[14] K. G. Santosh, W. G. Bharti and Y. Pravin, “A Review on Speech Recognition Technique,” *International Journal of Computer Applications,* vol. 10, no. 3, pp. 16-24, 2010.

[15] F. Sadaoki, “Automatic speech recognition and its application to information extraction,” in *37th Meeting of ACL*, pp. 11-20, 1999.

[16] A. L. Buchbaum and R. Giancarlo, “Algorithmic aspects in speech recognition: an introduction,” *ACM Journal of Experimental Algorithms,* vol. 2, no. 1, pp. 1-44, 1997.

[17] G. Salvi, “Developing acoustic models for automatic speech recognition,” MSc thesis, TMH, KTH, Stockholm, Sweden, 1998.

[18] A. Acero, “Speech Recognition and Understanding,” 2003. [Online]. Available: http://research.microsoft.com/en-us/um/redmond/groups/srg/videos/tutorial.ppt. [Accessed 04 August 2015].

[19] W. Ghai and N. Singh, “Literature review on automatic speech recognition,” *International Journal of Computer Applications,* vol. 41, no. 8, pp. 42-50, 2012.

[20] Z. Ghahramani, “An introduction to Hidden Markov Models and Bayesian Networks,” *International Journal of Pattern Recognition of Artificial Intelligence,* vol. 15, no. 1, pp. 9-42, 2001.

[21] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE,* vol. 77, no. 2, pp. 257-386, 1989.

[22] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose , V. Tyagi and C. Wellekens, “Automatic speech recognition and variability: a review,” *Speech Communication,* vol. 49, pp. 763-786, 2007.

[23] J. N. Holmes, W. J. Holmes and P. N. Garner, “Using formant frequencies in speech recognition,” in *EUROSPEECH*, pp. 2083-2086, 1997.

[24] C. J. Chen , R. A. Gopinath, M. D. Monkowski, M. A. Picheny and K. Shen , “New methods in continuous speech recognition,” in *EUROSPEECH*, pp. 1543-1546, 1997.

[25] T. A. Stephenson, “Speech recognition with auxiliary information,” PhD thesis, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2004.

[26] Wikipedia contributors, “Acoustic model,” Wikipedia, The Free Encyclopedia., 2015. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Acoustic_model&oldid=721809231. [Accessed 21 July 2016].

[27] X. Huang, A. Acero and H. W. Hon, Spoken language processing, Upper Saddle River, New Jersey 07458: Prentice-Hall, 2001.

[28] E. Arisoy, T. N. Sainath, B. Kingsbury and B. Ramabhadran, “Deep Neural Network Language Model,” in *NAACL-HLT 2012 Workshop: Will We Ever Really*

Replace the N-gram Model? On the Future of Language Modeling for HLT, pp. 20-28, 2012.

[29] A. G. Adami, “Automatic speech recognition: from beggining to the Portuguese language,” in *International Conference on Computational processing of the Portuguese language (PROPOR)*, pp. 1-73, 2010.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, G. Stemmer and K. Vesely, “The Kaldi speech recognition toolkit,” in *IEEE ASRU Workshop*, Hilton Waikoloa Village, Big Island, Hawaii, US, 2011.

[31] A. Lee, T. Kawahara and K. Shikano, “Julius - an open source real-time large vocabulary recognition engine,” in *EUROSPEECH*, pp. 1691-1694, 2001.

[32] K. Samudravijaya, “Toolkits for ASR: Sphinx,” in *Workshop on fundamentals of automatic speech recognition CDAC Noida*, pp. 1-31, 2011.

[33] C. Gaida, P. Lange, R. Petrick, P. Proba, A. Malatawy and D. Suendermann-Oeft, “Comparing open-source speech recognition toolkits,” Technical Report, DHBW Stuttgart, 2014.

[34] F. Brett, “The top five uses of speech recognition technology,” Call Centre Helper magazine, 2008. [Online]. Available: https://www.callcentrehelper.com/the-top-five-uses-of-speech-recognition-technology-1536.htm. [Accessed 21 July 2016].

[35] J. T. Goodman, “A bit of progress in language modeling,” *Computer Speech and Language,* vol. 15, no. 4, pp. 403-434, 2001.

[36] F. Jelinek, “Up from trigrams! The struggle for improved language models,” in *EUROSPEECH*, pp. 1037-1040, 1991.

[37] H. Schwenk, “Continuous space language models,” *Computer Speech and Language,* vol. 21, no. 3, pp. 492-518, 2007.

[38] X. Chen, “Scalable Recurrent Neural Network Language Models for Speech Recognition,” PhD thesis, University of Cambridge, Cambridge, England, 2017.

[39] I. Kipyatkova and A. Karpov, “Recurrent neural network-based language modeling for an automatic Russian speech recognition system,” in *AINL-ISMW-FRUCT*, pp. 33-38, 2015.

[40] H. Schwenk and J.-L. Gauvain, “Training Neural Network Language Models on Very Large Corpora,” in *Human Language Technology Conference and Conference on Empirical Methods in Natural LanguageProcessing (HLT/EMNLP)*, pp. 201-208, 2005.

[41] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky and S. Khudanpur, “Recurrent neural network based language model,” in *INTERSPEECH*, pp. 1045-1048, 2010.

[42] M. Sundermeyer, I. Oparin, J. Gauvain, B. Freiberg, R. Schlüter and H. Ney, “Comparison of Feedforward and Recurrent Neural Network Language Models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8430-8434, 2013.

[43] T. Kaufmann and B. Pfister, “Syntactic language modeling with formal grammars,” *Speech Communication,* vol. 54, no. 6, pp. 715-731, 2012.

[44] A. Deoras, T. Mikolov, S. Kombrink and K. Church, “Approximate inference: A sampling based modeling technique to capture complex dependencies in a language model,” *Speech Communication,* vol. 55, no. 1, pp. 162-177, 2013.

[45] R. Rosenfeld, “Two Decades of Statistical Language Modeling: Where do we go from here?,” *Proceedings of the IEEE,* vol. 88, no. 8, pp. 1270 - 1278, 2000.

[46] I. Kipyatkova, V. Verkhodanova and A. Karpov, “Rescoring N-best lists for Russian speech recognition using factored language models,” in *SLTU*, pp. 81-86, 2014.

[47] S. Seng, S. Sam, V.-B. Le, B. Bigi and L. Besacier, “Which units for acoustic and language modeling for Khmer automatic speech recognition?,” in *SLTU*, pp. 33-38, 2008.

[48] D. Yuret and E. Biçici, “Modeling morphologically rich languages using split words and unstructured dependencies,” in *ACL-IJCNLP Conference Short Papers*, pp. 345-348, 2009.

[49] B. Allison, D. Guthrie and L. Guthrie, “Another look at the data sparsity problem,” in *Text, Speech and Dialogue*, pp. 327-334, 2006.

[50] T. Alumae and M. Kurimo, “Efficient Estimation of Maximum Entropy Language Models with N-gram features: an SRILM extension,” in *INTERSPEECH*, pp. 1820-1823, 2010.

[51] Y. M. Tachbelie, T. S. Abate and L. Besacier, “Using different acoustic, lexical and language modeling units for ASR of an under-resourced language – Amharic,” *Speech Communication,* vol. 56, no. 1, pp. 181-19, 2014.

[52] S. Virpioja, P. Smit and S.-A. Grönroos, “Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline,” Aalto University publication series SCIENCE + TECHNOLOGY 25/2013, Helsinki, 2013.

[53] K. Kirchhoff, J. Bilmes and K. Duh, “Factored Language Models Tutorial,” Tutorial, Dept of EE, University of Washington, Seattle, Washington, 2008.

[54] K. Kirchhoff, D. Vergyri, J. Bilmes, K. Duh and A. Stolcke, “Morphology-based language modeling for conversational Arabic speech recognition,” *Computer Speech and Language,* vol. 20, no. 4, pp. 589-608, 2006.

[55] R. Rosenfeld and P. Clarkson, “Statistical Language Modeling Using the CMU-Cambridge Toolkit,” in *EUROSPEECH*, pp. 2707-2710, 1997.

[56] A. Stolcke, “SRILM-an extensible language modeling toolkit,” in *INTERSPEECH*, pp. 901-904, 2002.

[57] A. Stolcke , J. Zheng , W. Wang and V. Abrash, “SRILM at sixteen: Update and outlook,” in *IEEE ASRU Workshop*, Hilton Waikoloa Village, Big Island, Hawaii, USA, 2011.

[58] V. B. Le, L. Besacier, S. Seng, B. Bigi and D. Thi-Ngoc-Diep , “Recent advances in automatic speech recognition for Vietnamese,” in *SLTU*, pp. 47-52, 2008.

[59] B. Chen, “Introduction to SRILM Toolkit,” Tutorial Presentation, Department of Computer Science & Information Engineering, National Taiwan Normal University.

[60] J. C. Roux, P. H. Louw and T. R. Niesler, “The African Speech Technology Project: An Assessment,” in *Language Resources and Evaluation Conference*, pp. 93-96, 2004.

[61] J. Badenhorst, C. van Heerden, M. Davel and E. Barnard, “Collecting and evaluating speech recognition corpora for 11 South African languages,” *Language Resources and Evaluation,* vol. 45, no. 3, pp. 289-309, 2011.

[62] E. Barnard, M. H. Davel, C. van Heerden, F. De Wet and J. Badenhorst, “The NCHLT speech corpus of the South African languages,” in *SLTU*, pp. 194-200, 2014.

[63] E. Barnard, M. Davel and C. van Heerden, “Automatic Speech Recognition corpus design for resource-scarce languages,” in *INTERSPEECH*, pp. 2847-2850, 2009.

[64] C. van Heerden, E. Barnard and M. Davel, “Basic speech recognition for spoken dialogues,” in *INTERSPEECH*, pp. 3003-3006, 2009.

[65] E. Barnard, M. H. Davel and G. B. van Huyssteen, “Speech Technology for Information Access: a South African Study,” in *AAAI Spring Symposium: Artificial Intelligence for Development*, pp. 13-15, 2010.

[66] A. S. Grover, G. B. van Huyssteen and M. W. Pretorius, “The South African Human Language Technology Audit,” *Language Resources and Evaluation,* vol. 45, no. 3, pp. 271-288, 2011.

[67] CText, “Language Resource Management Agency (RMA),” South African Department of Arts and Culture (DAC), 2013. [Online]. Available: http://rma.nwu.ac.za/index.php/. [Accessed 25 January 2017].

[68] D. Sindana and M. J. Manamela , “Development of robust language models for recognition of under-resourced languages,” Poster Presentation, Southern Africa Telecommunication Networks and Applications Conference, 2015.

[69] D. Sindana and M. J. Manamela, “The Influence of orthography on language modelling for speech recognition,” in *Southern African Telecommunication Networks and Applications Conference*, pp. 302-307, 2016.

[70] D. Sindana, M. J. Manamela and T. I. Modipa, “Orthography-based language modelling for speech recognition,” Poster presentation, Center for High Performance Computing – National Meeting, 2016.

[71] D. Sindana, M. J. Manamela and T. I. Modipa, “Orthography-based language modelling for speech recognition,” Oral presentation, Inaugural Conference of the Digital Humanities Association of Southern Africa, 2017.

[72] D. Sindana, M. J. Manamela and T. I. Modipa, “The effect of orthography on language modelling,” in *Southern Africa Telecommunication Networks and Applications Conference*, pp. 240-245, 2017.