SPOKEN LANGUAGE IDENTIFICATION IN RESOURCE-SCARCE ENVIRONMENTS

by

Marius Peché

Submitted in partial fulfillment of the requirements for the degree

Masters of Engineering (Applied Science)

in the

Department of Electrical, Electronic and Computer Engineering
UNIVERSITY OF PRETORIA

April 2009

ABSTRACT

South Africa has eleven official languages, ten of which are considered "resource-scarce". For these languages, even basic linguistic resources required for the development of speech technology systems can be difficult or impossible to obtain.

In this thesis, the process of developing Spoken Language Identification (S-LID) systems in resource-scarce environments is investigated. A Parallel Phoneme Recognition followed by Language Modeling (PPR-LM) architecture is utilized and three specific scenarios are investigated: (1) incomplete resources, including the lack of audio transcriptions and/or pronunciation dictionaries; (2) inconsistent resources, including the use of speech corpora that are unmatched with regard to domain or channel characteristics; and (3) poor quality resources, such as wrongly labeled or poorly transcribed data. Each situation is analysed, techniques defined to mitigate the effect of limited or poor quality resources, and the effectiveness of these techniques evaluated experimentally.

Techniques evaluated include the development of orthographic tokenizers, bootstrapping of transcriptions, filtering of low quality audio, diarization and channel normalization techniques, and the human verification of miss-classified utterances.

The knowledge gained from this research is used to develop the first S-LID system able to distinguish between all South African languages. The system performs well, able to differentiate among the eleven languages with an accuracy of above 67%, and among the six primary South African language families with an accuracy of higher than 80%, on segments of speech of between 2s and 10s in length.

Key Terms: Human Language Technologies; Spoken Language Identification; Automatic Speech Recognition; Parallel Phoneme Recognition followed by Language Modeling; Incomplete Resources; Mismatched Resources; Suboptimal Resources.

ABSTRAK

Suid-Afrika het elf amptelike tale waarvan tien as hulpbron-skaars beskou word. Vir die tien tale kan selfs die basiese hulpbronne wat benodig word om spraak tegnologie stelsels te ontwikkel moeilik wees om te bekom.

Die proses om ‘n Gesproke Taal Identifisering stelsel vir hulpbron-skaars omgewings te ontwikkel, word in hierdie tesis ondersoek. ‘n Parallelle Foneem Herkenning gevolg deur Taal Modellering argitektuur word ingespan om drie spesifieke moontlikhede word ondersoek: (1) Onvolledige Hulpbronne, byvoorbeeld vermiste transkripsies en uitspraak woordeboeke; (2) Teenstrydige Hulpbronne, byvoorbeeld die gebruik van spraak data-versamelings wat teenstrydig is in terme van kanaal kenmerke; en (3) Hulpbronne van swak kwaliteit, byvoorbeeld foutief geklasifiseerde data en klank opnames wat swak getranskribeer is. Elke situasie word geanaliseer, tegnieke om die negatiewe effekte van min of swak hulpbronne te verminder word ontwikkel, en die bruikbaarheid van hierdie tegnieke word deur middel van eksperimente bepaal.

Tegnieke wat ontwikkel word sluit die ontwikkeling van ortografiese ontleders, die outomatiese ontwikkeling van nuwe transkripsies, die filtrering van swak kwaliteit klank-data, klank-verdeling en kanaal normalisering tegnieke, en menslike verifikasie van verkeerd geklassifiseerde uitsprake in.

Die kennis wat deur hierdie navorsing bekom word, word gebruik om die eerste Gesproke Taal Identifisering stelsel wat tussen al die tale van Suid-Afrika kan onderskei, te ontwikkel. Hierdie stelsel vaar relatief goed, en kan die elf tale met ‘n akkuraatheid van meer as 67% identifiseer. Indien daar op die ses taal families gefokus word, verbeter die persentasie tot meer as 80% vir segmente wat tussen 2 en 10 sekondes lank.

Sleutel Terme: Menslike Taal Tegnologie; Gesproke Taal Identifisering; Outomatiese Spraak Herkenning; Parallelle Foneem Herkenning gevolg deur Taal Modellering; Onvolledige Hulpbronne; Teenstrydige Hulpbronne; Ondergeskikte Hulpbronne.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

Human Language Technologies (HLT), such as Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) systems, are becoming more part of our daily lives. Such technologies can be found in many computerized systems for various reasons that may range from more effective customer support in telephonic systems to accessibility functions on a personal computer.

The ability to identify language automatically is of great importance in systems that offer functionality in more than one language. For instance, a Textual Language Identification system can identify which set of pronunciation rules a polyglot Text-to-Speech (TTS) system has to use for any given sample of text, and Spoken Language Identification is of great importance in an audio data management system which has to handle massive amounts of data. We are specifically interested in Spoken Language Identification, and the development of such systems in resource-scarce environments.

This chapter is structured as follows: A literature review in Section 1.1 provides an overview of prior work related to this thesis. This is followed by a discussion of the problem statement in Section 1.2 and research methodology in Section 1.3. The chapter concludes with a summary of the contribution that is made through this research, in Section 1.4.

1.1 LITERATURE REVIEW

Prior work related to language identification is reviewed in this section. The language identification task is defined in Section 1.1.1, before the possible sources of information that can be used to distinguish among languages are discussed in Section 1.1.2. An overview of common S-LID techniques is provided in Section 1.1.3, with special attention paid to the Parallel Phoneme Recognition followed by Language Modeling technique in Section 1.1.4, as this is the most suitable technique for our envisaged research. The different variables that influence the accuracy of the Language Identification task are described in Section 1.1.5, before the current benchmark results obtained using state-of-the-art systems are introduced in Section 1.1.6. The section concludes with a brief overview of South Africa's official languages in Section 1.1.7

1.1.1 THE LANGUAGE IDENTIFICATION TASK

Language Identification (LID) is the process whereby the most likely candidate language in which some sample of speech is delivered is chosen from a set of possible target languages. The sample of speech that is considered may be recorded either in textual format or may be an audio segment. This difference therefore leads to two closely related yet sufficiently distinct branches of LID:

- Textual Language Identification (T-LID), and
- Spoken Language Identification (S-LID).

T-LID can be assumed to have been mostly solved, since n-gram techniques have yielded high accuracies on fairly small test-sets, as reported by Cavnar and Trenkle [1] in 1994. Similar results have been reported on a system built to distinguish between South African languages as well [2]. This can be attributed to the properly defined and accurate tokens in the form of alphabetical letters that T-LID utilizes as a source of information. S-LID, on the other hand, still presents a more complex problem since the speech sample exists as an audio signal, and the process to extract phonetic tokens from such audio is itself prone to errors.

1.1.2 INFORMATION SOURCES

When we consider a segment of audio speech, it is found that there are several features that differ from language to language. These are utilized to perform S-LID, with varying results. Initial attempts include focusing on prosodic [3] and spectral [4] information sources. Prosodic information involves variances such as the rhythm and intonation of a language, whereas spectral resources define the level of power used at different frequencies to produce speech. The proposition is that the same phonemes are pronounced differently enough between languages for the system to identify.

Other previous approaches to S-LID focus their attention on other sources as well, including reference sounds [5], pitch contours [6] and other raw waveform features [7]. However, these features represent a low level of linguistic knowledge, and systems utilizing such features may barely perform satisfactory.

Having said this, it has been hypothesized that the higher the level of knowledge presented within the extracted features is, the better the results will be [8]. However, there is a clear trade-off between complexity and performance. In other words, systems that depend on prosodic information do not perform as well as a system utilizing phonotactic information, and systems that extract information from the lexicon or even the syntax perform much better that their lower counterparts [9]. However, such systems that focus on information sources with a higher linguistic knowledge representation have proven to have a greater computational cost, as the design of the system itself is more complex and they also require much more labeled training data. As an example, a syntax-based system must posses the capability to identify not only the possible phonemes, but must have a dictionary to determine the possible words spoken as well as a proper, high-level language model to determine the validity of the resulting sentence.

A balance between the complexity and performance of a desired system has to be decided upon. For this reason, most researchers prefer to utilize acoustic resources [3], especially in the field of phonetic tokens as is the case with Parallel Phoneme Recognition followed by Language Modeling (PPR-LM), which is discussed later.

Even though many S-LID systems focus on only one source of information, it has been shown that results from more than one source can be combined and that such hybrid systems do show an improved result [8, 10].

1.1.3 COMMON TECHNIQUES

Initial systems tried to model the complexities of prosodic information such as pitch and rhythm [3]. Though humans appear to display some success in this approach, it has proven difficult to extract meaningful information from the audio signal and thus these approaches have not been very successful to date [11].

Word-spotting is another technique widely used by humans which has proven difficult to implement [12]. Word-spotting involves predefining key words that usually occur frequently within a target language and then trying to identify these key words within the speech sample. However, such an approach can easily fail when the required key words are not present.

Classification methods are also regularly considered as part of S-LID systems, and include approaches such as clustering algorithms [6], expert systems [13], Gaussian Mixture Models (GMM), Hidden Markov Models (HMM) [14] and Artificial Neural Networks [7] that try to represent the entire language. An entire language has proven to be too complex for these models [15], and it has been found that combining techniques from T-LID and other speech processing techniques can yield superior results. Usable tokens can be extracted from a sample of audio speech on a phonotactical, lexical or syntactic level [3] and used to train a classifier.

Good results are obtained by using the Parallel Phoneme Recognition followed by Language Modeling (PPR-LM) architecture [15], and systems that extract information from the lexicon or even the syntax perform even better [3]. Since there exists a clear connection between performance on the one hand and complexity on the other, the PPR-LM architecture has proven to be very popular within the research community.

Many state-of-the-art systems today consist of hybrid systems that combine various approaches (for example general and gender-specific acoustic models in PPR-LM or systems built using different sources of information) [8, 10, 16, 17].

1.1.4 PARALLEL PHONEME RECOGNITION FOLLOWED BY LANGUAGE MODELING

Parallel Phoneme Recognition followed by Language Modeling (PPR-LM) utilizes Automatic Speech Recognition (ASR) systems to extract tokens in the form of phonotactical information. Results from this front-end are then passed to a back-end where the system uses any form of language modeling (eg. n-grams) to determine the most probable language from the set of target languages [18].

Tokens are usually language-dependent phonemes. Though PPR-LM systems can function with only one target language tokenizer to process the audio signal [4], it appears that language-dependent phonemes in several of the target languages seem to work best [19]. In such a case, the typical PPR-LM system usually requires a set of phoneme recognizers, one in each of the target languages. These acoustic models then run in parallel to produce the tokens required by the back-end. However, to create effective acoustic models, a large quantity of transcribed audio data as well as accurate pronunciation dictionaries are required. The addition of a new language is more difficult [3], especially for languages with insufficient resources. It is also logical to assume that any classifier used to score the possible languages needs to be adjusted accordingly.

The better results achieved with more than one tokenizer may be because similar phonemes vary from language to language, and performance should increase as more tokenizers are added to the front-end [20]. Another suggestion to make the addition of languages to an existing system easier involves the use of a multilingual tokenizer. Such a tokenizer involves the use of all language-dependent and language-independent phones brought together into a super set [21, 22]. However, to create such a phoneme set, a large and balanced set of transcribed data is needed to ensure that the multilingual tokenizer does not favor one language's phoneme set above the rest. Therefore rarely seen languages remain difficult to add.

It has also been suggested to use phoneme-independent recognizers. Although such units can be bootstrapped from phonetically similar tokens, it appears that many of these units are still required to achieve similar results, thereby increasing the computational cost unnecessarily. It may also be difficult to augment the token set for both the multilingual and the phoneme-independent sets [21].

In order to develop a phoneme recognizer for a PPR-LM system, the following resources are typically required:

- Audio resources, labeled according to language.
- Written transcriptions of the audio resources.
- A pronunciation dictionary.

Once the front-end has processed the audio signal, the resulting token strings are scored by a classifier in the back-end and the most probable language returned. Though language models are usually employed to distinguish between languages, it has been shown that extracting the n-gram frequencies from the phoneme strings as a vector and using a Support Vector Machine (SVM) [23] to classify these vectors into groups representing the languages can prove to be more successful [19].

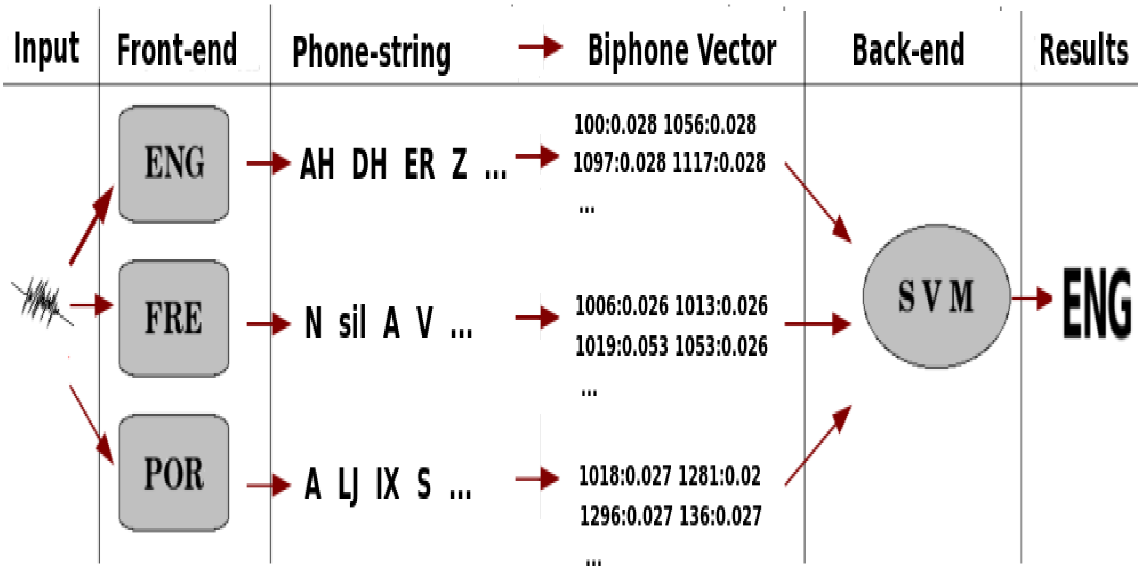

Figure 1.1: Visual representation of the PPR-LM architecture.

Figure 1.1 provides a visual representation of the PPR-LM architecture. An utterance given as input to the system is passed to three ASR systems (English, French and Portuguese phoneme recognizers in the image) that together form the front-end of the system. These ASR systems produce phoneme strings which are then passed as a vector of biphone frequencies to the language model at the back-end (an SVM classifier in the image) which then predicts the language spoken in the utterance.

1.1.5 VARIABLES THAT INFLUENCE THE ACCURACY OF THE S-LID TASK

It has been shown that for T-LID a number of factors play an important role in the accuracy of the system as a whole. These include: size of the text-fragment, amount and variety of training data, and classification algorithm. It also has been discovered that the composition of the target languages play an important role in the overall accuracy. It is much harder to distinguish between similar languages such as isiZulu and isiXhosa [24] than between unrelated languages. The same factors influence the accuracy of S-LID systems as well, most notably the size of the audio segments and the composition of the training sets [15].

The number of target languages known to the system also has an influence on the accuracy of the system as a whole. Researchers have been able to achieve much higher accuracies using language-pair recognition than trying to recognize ten languages at once [4]. In 1994, systems that tried to distinguish between a large number of languages have proven unreliable [15], though such systems have become much more accurate recently as more training data is becoming available.

In summary, S-LID results can only be compared accurately if the following variables are specified:

- The length of the audio segment used during testing.

- The number of target languages the system has to distinguish between.
- The use of an open or closed set of languages during testing. *Open* sets refer to test sets which contain unseen languages as well, which are unknown to the system.
- The specific languages distinguished amongst (specifically the extent to which these languages are related).

1.1.6 CURRENT BENCHMARK RESULTS

The National Institute of Standards and Technology (NIST) Language Recognition Evaluation (LRE) evaluates a series of experimental systems in order to establish the current baseline performance of language recognition [25]. Though initially started in 1996, the next evaluation was only in 2003, after which it has been repeated every two years. At the time of this thesis, the most recent results were published in 2007, with another evaluation planned for later in 2009.

In the 2007 evaluation two sets of experiments were conducted: A closed set, which contained only samples from predetermined languages; and an open set which also contained unknown (out-of-set) languages for which there were no training data available. However, out-of-set languages fall outside the scope of this thesis.

The *General Language Recognition* evaluation covered the following fourteen languages: Arabic, Bengali, English, Farsi, German, Hindustani, Japanese, Korean, Russian, Spanish, Tamil, Thai, Vietnamese and Chinese. In the cases of English, Chinese, Hindustani and Spanish, separate systems which determined the dialect were also evaluated. All audio data was part of the CallAFriend telephonic corpora.

The NIST-LRE measures the system achievements based only on pair-wise language recognition performance. Accuracy scores are calculated for each target language, based on the probability that the system incorrectly classifies an audio segment. The following expression gives the accuracy score for each target-non-target language-pair:

$$C\left(L_T, L_N\right) = \begin{aligned} & C_{Miss} P_{Target} P_{Miss}\left(L_T\right) \\ & + C_{FA}\left(1 - P_{Target}\right) P_{FA}\left(L_T, L_N\right) \end{aligned} \tag{1.1}$$

where L_T and L_N are the target and non-target languages respectively. C_{Miss} and C_{FA}, are the weights (costs) for each miss-classifications (*false reject* and *false accept* respectively), which are set to $C_{Miss} = C_{FA} = 1$ for the NIST evaluations. P_{Target}, is the probability of the audio segments being the target language and is set to $P_{Target} = 0.5$. P_{Miss} and P_{FA} are the system-specific measured *false reject* (incorrectly rejecting a true statement as false) and *false accept* (incorrectly accepting a false statement as true) values (expressed as probabilities).

An overall score is then computed for each system by adding together all these values for each target-non-target language pair, which is calculated for each target language. The following is the mathematical equation:

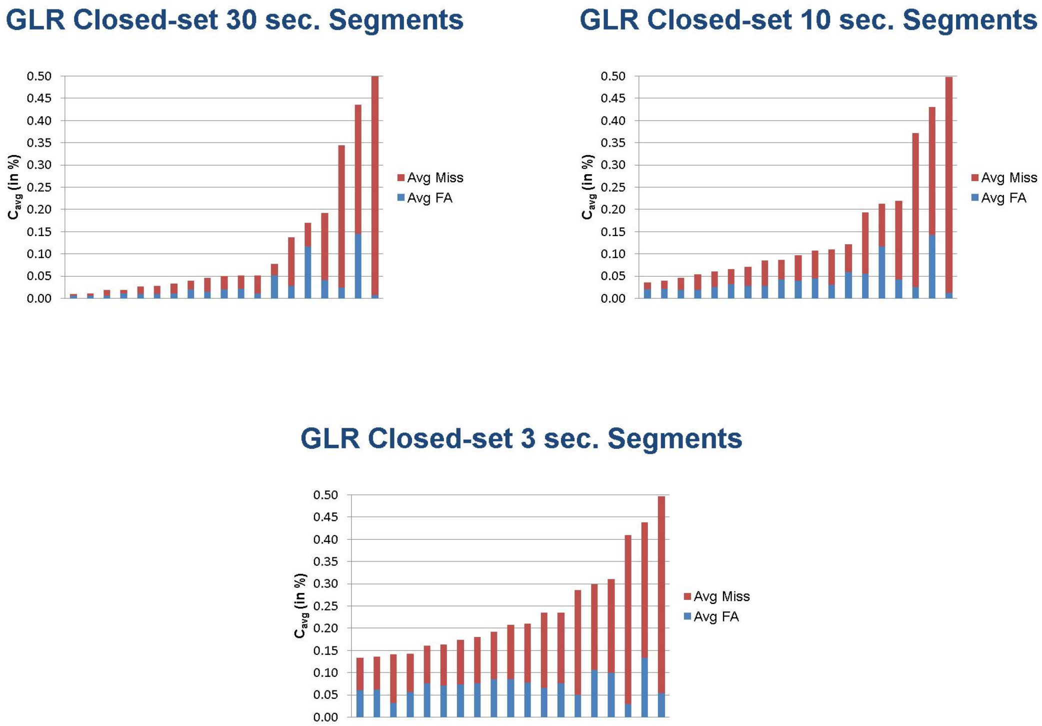

Figure 1.2: C_{avg} scores for the 2007 NIST LRE systems when evaluated with the general language closed set on 30, 10 and 3 second long test samples. Figures reproduced from [25].

$$C_{avg} = \frac{1}{N_L} \sum_{L_T} \left(\begin{array}{c} C_{Miss} P_{Target} P_{Miss} \left(L_T \right) \\ + \sum_{L_N} C_{FA} P_{Non-Target} P_{FA} \left(L_T, L_N \right) \\ + C_{FA} P_{Out-of-Set} P_{FA} \left(L_T, L_O \right) \end{array} \right) \tag{1.2}$$

where N_L is the number of languages in the closed-set, and L_O an out-of-set (or unknown) language.

Twenty-one organizations or teams from around the world partook in the evaluation. As can be seen in Figure 1.2, most of the systems that were tested on the general language recognition's closed set, achieved average costs of below 0.05 for the 30 sec segments. However, performance across the board is poorer for the 3 sec segments, as most systems only achieve a score of below 0.25.

Many of these systems are propriety, therefore their implementation details are considered sensitive. Figure 1.2 also shows that the system-specific performance during the NIST LRE is not made public either. The best systems' performance for the evaluations in both 2005 and 2007 are given in Table 1.1 [26].

The systems presented for the NIST LRE utilize several pre-processing techniques, such as *Voice Activity Detection* to remove any long silences or non-speech signals and *Speaker Clustering* which

NIST LRE	3-sec	10-sec	30-sec
2005	0.1569	0.0715	0.0419
2007	0.1335	0.0363	0.0103

Table 1.1: The best C_{avg} scores for both NIST LRE05 and NIST LRE07 [26].

assumes that one speaker only speaks in one language, before detecting the language of the test segment. For instance, the systems built by the International Computer Science Institute in the USA, as well as the Brno University of Technology in the Czech Republic, preprocessed the data by reducing the amount of noise as well as removing all of the silences beforehand.

Another common technique is to combine several of the systems mentioned in Section 1.1.3 into one. The language recognition of the ICSI system itself consisted of the common PPR approach, combined with an SVM. However, their front-end utilizes four different phoneme recognizers, one based on gender-specific HMMs and the other three language-specific Artificial Neural Networks based on Multilayer perceptron (MLP) phone classifiers [16]. The MLP classifiers were trained on English, Arabic and Mandarin. The ICSI system achieved an average cost of 0.076 for the 30-sec samples, for the general language recognition task. The BUT system utilizes a combination of 4 HMM-based acoustic and 9 GMM-based phonetic systems, achieving an average cost of 0.075 on 30-sec long samples [17].

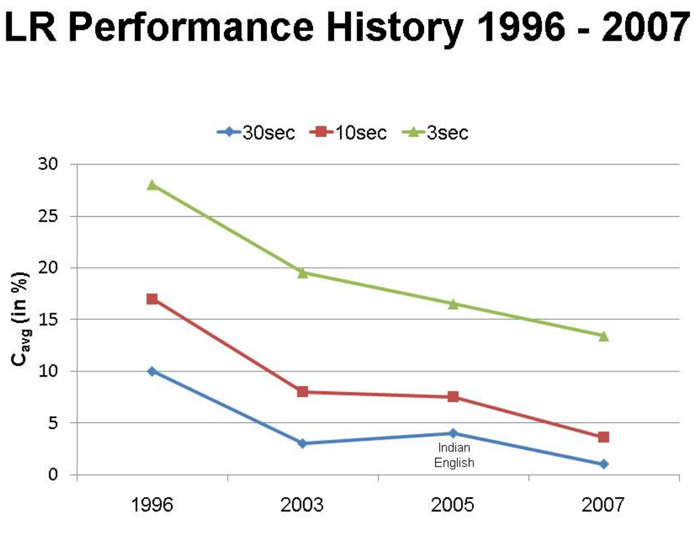

Figure 1.3: Historic C_{avg} scores for the 1996 - 2007 NIST LRE in the general language recognition task, grouped according to test-segment lengths. Figure reproduced from [25].

As can be seen from Figure 1.3, the performance of the LRE systems keeps increasing over the years. Also notice that the results clearly indicate that the length of the test segments influences the accuracy, with both tests using 30-sec and 10-sec audio segments achieving C_{avg} scores of below 0.05 (5% as indicated in Figure 1.3, using percentages), and the test using 3-sec segments lagging behind with a cost of just below 0.15 (15%, as indicated) during the 2007 NIST LRE.

1.1.7 OFFICIAL LANGUAGES OF SOUTH AFRICA

Since 1994, South Africa has eleven recognized official languages. These are listed in Table 1.2, along with each language's international ISO language code, the number of native speakers (in millions) and the language family it is classified in. As can be seen from Table 1.2, several of South Africa's official languages do not have a large speaker population, which makes the gathering of audio resources difficult.

Of particular importance is the language families. These families represent languages which exhibit similarities with regard to grammar, vocabulary and pronunciation. Three major language families are present in South Africa, namely the Germanic languages, the Nguni languages and the Sotho-Tswana languages. Whereas the Germanic languages are of Indo-European origin, both the Nguni and the Sotho-Tswana language families represent two of the major branches of the Southern Bantu languages which originated in Central to Southern Africa. Tswa-Ronga and Venda are also part of the Southern Bantu languages.

Language	ISO Code	Native Speakers	Language Family
isiZulu	zul	10.7	Nguni
isiXhosa	xho	7.9	Nguni
Afrikaans	afr	6.0	Germanic
Sepedi	nso	4.2	Sotho-Tswana
Setswana	tsn	3.7	Sotho-Tswana
Sesotho	sot	3.6	Sotho-Tswana
SA English	eng	3.6	Germanic
Xitsonga	tso	2.0	Tswa-Ronga
siSwati	ssw	1.2	Nguni
Tshivenda	ven	1.0	Venda
isiNdebele	nbl	0.7	Nguni

Table 1.2: A list of South Africa's eleven official languages [27].

1.2 PROBLEM STATEMENT

This section describes the *Research Questions* addressed and lists the *Motivations* and *Objectives* of this thesis.

1.2.1 RESEARCH QUESTION

In order to create a South African S-LID system, the following questions need to be explored first:

- Is it possible to create a reliable S-LID system with limited resources?
- What techniques can be used to increase the effectiveness of limited input data?
- How does poor quality data such as incorrectly labeled, poorly transcribed or poor audio quality data influence attempts to create such S-LID systems?
- What can be done to reduce the negative influence of poor quality data?

The hypothesis which will be tested is that it is indeed possible to create such an S-LID system, provided that a significant portion of the available data is at least correctly labeled (with the correct language identity) and that the poorly resourced languages can borrow tools and resources from better studied languages, such as English.

1.2.2 MOTIVATION

South Africa has the challenge of having eleven official languages, ten of which have not been studied from a language technology perspective in great detail. Demographically, several of these languages are not widely spoken as well, even when compared to other South African languages (e.g. Tshivenda is not as widely spoken as isiZulu). Thus resources are quite scarce and the electronic data that is available may be incorrect (such as wrongly labeled data or poorly transcribed data), incomplete (missing transcriptions or pronunciation dictionaries) or even inconsistent (audio files from one set of data may be recorded under different conditions than another set).

Being able to develop accurate S-LID systems, even with such limited resources, may pave the way towards incorporating additional languages in speech technology systems. S-LID systems in particular are important as they can play a useful role in data collection, enabling additional applications to be developed later, including spoken dialog systems in domains such as government service delivery or healthcare [28, 29].

1.2.3 OBJECTIVE

The objective of this research is to create a set of techniques for the development of an S-LID system when limited linguistic resources are available. It determines how limited, poor quality or incomplete audio data can be used best to improve S-LID accuracy or build completely new systems. Finally, the information gained from this thesis is put to use and an S-LID system is developed which is, to our knowledge, for the first time able to distinguish between all eleven of South Africa's official languages.

1.3 RESEARCH METHODOLOGY

This section provides an overview of the *Data Corpora* used in this thesis. It also explains the *Experimental Design*, focusing especially on the tokenizer and classifier used for the experiments throughout this thesis, before ending off with an overview of the *Research Focus*, introducing the topics which are covered in the rest of this thesis.

1.3.1 DATA CORPORA

The following corpora are used during this research:

- The GlobalPhone corpus [30]. Languages available include: French, Portuguese, and Japanese.
- The Wall Street Journal corpus provides American English [31]. The English from the Wall Street Journal is used in conjunction with the GlobalPhone corpus, since the two corpora have very similar acoustic characteristics.
- An in-house corpus (which is referred to as the African corpus) containing telephonic data in English, French and Portuguese, as spoken throughout Africa. This data is of a very poor quality and the corpus does not contain any written transcriptions.
- The Meraka Lwazi corpus [32]. Languages available are: Afrikaans, South African English, isiNdebele, isiXhosa, isiZulu, Setswana, Sepedi, Sesotho, siSwati, Tshivenda, and Xitsonga. This is a new corpus of telephonic data that is still under development.

1.3.2 EXPERIMENTAL DESIGN

The popular PPR-LM configuration, which is explained in Section 1.1.4 is used for all the experiments throughout this thesis, since it is widely used in the literature. The configuration is also relatively easy to develop, powerful enough to achieve satisfactory accuracies and does not require the development of any high level linguistic tools or data. The latter is critical as the focus is on languages with little resources. The following provides a brief overview of the system design. Further detail is provided in the individual chapters.

1.3.2.1 ACOUSTIC RECOGNIZERS

The tokenizers which are used to convert the audio signal into usable tokens are phoneme recognizers from ASR systems. These phoneme recognizers utilize Hidden Markov Models (HMM) which are trained to recognize biphones or triphones from Mel Frequencies Cepstral Coefficients (MFCC). The training of the HMMs as well as the extraction of the MFCCs from the audio signal, is performed by the HMM Tool Kit (HTK) [33].

In most cases, recognizers for each of the target languages are utilized, and run in parallel with one another, each yielding a phoneme string for a given speech sample.

1.3.2.2 LANGUAGE CLASSIFIER

Recently published results [19] indicate that the use of a Vector Space Model to distinguish between the target languages currently yields the best results. Therefore this thesis makes use of a Support

Vector Machine (SVM). Biphone frequencies are extracted from the phoneme strings which have been recognized from the audio. The frequency of each unique biphone is represented as a term in a vector, with the results of each phoneme recognizer concatenated one after the other to form a single vector for each utterance. This vector is used as an input function to the back-end classifier.

A Gaussian kernel is employed as the kernel function of the SVM. A grid search is utilized to determine the optimal values of the kernel width and the margin-accuracy trade-off parameter. Using this design, several systems are developed and analyzed in order to address the questions described in the next section.

1.3.3 RESEARCH FOCUS

During the course of this research, three key questions related to the resources required to develop an S-LID system are examined. These three questions are expected to be typical problems faced by developers who are trying to create S-LID systems in resource-scarce languages, especially when it comes to acquiring the necessary resources to develop many HLT systems.

1.3.3.1 INCOMPLETE RESOURCES

The resources that are usually required to develop ASR systems, which are typically used as tokenizers in PPR-LM systems, include:

- Audio resources, accurately identified according to language.
- Written transcriptions of the audio resources.
- A pronunciation dictionary.

Though the audio resources remain critical for development, Chapter 2 determines if it is possible to develop ASR systems without the use of transcriptions or a pronunciation dictionary. The first experiment attempts to build an ASR system without a dictionary, using the individual letters as possible tokens. The second experiment investigates the possibility to utilize an already existing ASR system to bootstrap transcriptions for incomplete corpora.

The GlobalPhone corpus is used for this section. Japanese is artificially limited to represent a poorly resourced language.

1.3.3.2 MISMATCHED RESOURCES

Another important problem faced by resource-scarce languages is that the different corpora which contain the few resources available to build a system may be mismatched. Chapter 3 examines the possible techniques which can be used to port a system from a resource rich environment to a poorly resourced environment. Well-known tools such as diarization and channel normalization are investigated.

The African corpus is used in conjunction with the GlobalPhone corpus for this section. Since the African corpus does not possess any transcriptions, new transcriptions are bootstrapped from well-trained ASR systems used in the previous chapter.

1.3.3.3 POOR QUALITY RESOURCES

Another important challenge for languages in a limited-resource environment is the quality of the available data. Chapter 4 investigates the effect of poor quality data on the overall performance of the system. An experiment investigates the effect of poor transcriptions on the overall performance by altering the amount and quality of training data independently.

The Meraka Lwazi corpus is used in this section. Data collected during different stages of the quality verification process is used. The transcriptions, as well as some of the audio data in the isiZulu corpus were originally of poor quality, and is therefor used to represent the language with poor quality resources.

1.3.3.4 DEVELOPMENT OF A SOUTH AFRICAN S-LID SYSTEM

Chapter 5 utilizes the knowledge gained from the previous experiments in order to develop an S-LID system specifically aimed at distinguishing between South African languages. Again, the PPR-LM configuration is implemented and the Meraka Lwazi corpus is used for the development of the final system.

1.4 CONTRIBUTION

Usually the answer to poorer-than-desired performance is to “throw more data at the system”. However, since this thesis is looking at languages that have very little data available, better alternatives have to be found to create even a basically usable system.

This thesis aims to develop a set of techniques to aid in the development of an S-LID system with limited linguistic resources or when the quality of available resources are questionable. Such techniques include bootstrapping usable transcriptions, and automatically filtering data that restricts system performance. This thesis determines how data of poor quality or even incomplete data affects the overall system accuracy.

Finally, the information gained from this research is put to use and the first publicly released S-LID system which is able to distinguish between all eleven of South Africa’s official languages, is developed.

CHAPTER TWO

INCOMPLETE RESOURCES

2.1 INTRODUCTION

Creating a phoneme recognizer in environments where languages have limited resources may prove extremely difficult. Chapter 1 described the Spoken Language Identification (S-LID) task, and introduced the concept of Parallel Phoneme Recognition (PPR). To create the necessary phoneme recognizers, properly transcribed audio data as well as a pronunciation dictionary are required. In this chapter, the following two important issues will be addressed:

- The absence of any written transcriptions.
- The unavailability of a pronunciation dictionary.

Transcriptions record what is being said in the audio data, whereas the pronunciation dictionary accurately determines which phonemes make up the words written down in the transcriptions. In other words, both the transcriptions and a pronunciation dictionary are of vital importance to define the actual phonemes, as well as their order within the audio data.

This chapter investigates an environment where only incomplete linguistic data is available. It defines and examines techniques to create a phonotactic S-LID system when the transcriptions are not trusted, if they are even available, and pronunciation dictionaries may be difficult to acquire. Firstly the necessity of a tokenizer specific to the new target language is determined. As part of the first experiment, the influence of multiple tokenizers on the overall performance of an S-LID system is investigated. Secondly, techniques to create a phoneme recognizer for a language without complete linguistic resources, as well as the influence of such tokenizers on the overall performance of the system, is also determined.

Language	Set	Speakers	Utterances	Hours
English	Train	83	10 219	20.0
	Test	19	2 555	4.9
French	Train	80	8 380	21.6
	Test	21	2 096	5.3
Portuguese	Train	77	6 037	14.4
	Test	25	1 511	3.5

Table 2.1: Statistics on the training and testing sets for each language in the baseline system.

The remainder of this chapter is structured as follows: An overview of the setup of the experiments for the rest of the chapter is provided in Section 2.2. Section 2.3 describes the first set of experiments where the effect of multiple recognizers is investigated, followed by experiments where a recognizer for the new target language is created in Section 2.4. Section 2.5 discusses the results obtained.

2.2 EXPERIMENTAL SETUP

This section describes the design of the system used for the experiments in the rest of this chapter. Firstly, it provides an overview of the corpora used to create the baseline system in Section 2.2.1. Section 2.2.2 describes the general system design in more detail, before an overview of the performance measurements used for the experiments is provided in Section 2.2.3. The performance of the baseline system is described in Section 2.2.4.

2.2.1 CORPUS STATISTICS

Data from two acoustically similar corpora is used to create the baseline system. The GlobalPhone corpus [30] provides data for the French and Portuguese recognizers, as well as Japanese which is used to simulate a resource scarce language later in the chapter. Data for the English recognizer is acquired from the Wall Street Journal corpus [31]. The audio segments in both corpora are on average 8 seconds in length. Table 2.1 provides statistics on the data used, specifically the number of speakers within the corpus, as well as the combined amount of utterances for each language. The length of all the audio data for each language is also given in hours.

Pronunciation dictionaries for French and Portuguese are also acquired from the GlobalPhone corpus, and the publicly available Carnegie Mellon pronunciation dictionary version 0.7a is used for the English data [34].

2.2.2 BASELINE SYSTEM DESIGN

The baseline system utilizes the popular Parallel Phoneme Recognition followed by Language Modeling (PPR-LM) configuration [15]. The PPR-LM is used because it is also relatively easy to develop, powerful enough to achieve satisfactory accuracies and does not require the development of any high level linguistic tools or data.

Phoneme recognizers, which utilize Hidden Markov Models (HMMs) are created for each of the three target languages. The HMMs are trained to recognize biphones and use three emitting states. Each state consists of a Gaussian Mixture Model (GMM) with four mixtures. The amount of states and mixtures have been chosen based on experiments done prior to the start of this experiment [35].

The HMMs are trained on audio data which is encoded as Mel Frequency Cepstral Coefficients (MFCCs). The MFCCs are extracted from the audio signal and consist of 36 values for each 10 microseconds. These values are 12 coefficients, which are derived from log spectra based on a fast Fourier transform. For each coefficient, the difference between each two subsequent frames is also calculated (12 delta coefficients) as well as the difference between each two subsequent delta coefficients (12 acceleration coefficients).

The actual recognition of the phonemes within the audio segments is done in conjunction with a flat language model. Unlike a properly trained language model which stipulates which phoneme may follow the previously recognized phoneme, a flat language model allows the recognizer complete freedom to decide which phoneme is currently being uttered (in other words, any phoneme may follow any other). Note that this language model forms part of the ASR system and should not be confused with the Language Modeling that will classify the phoneme strings to determine the possible target language.

The system utilizes a vector space to classify the different target languages, which is implemented with a Support Vector Machine (SVM). Biphone frequencies are extracted from the phoneme strings which have been recognized from the audio. The frequency of each unique biphone is represented as a term in a vector, with the results of each phoneme recognizer concatenated one after the other to form a single vector for each utterance. This utterance is then used as input to the SVM.

The SVM employs a Gaussian kernel as the kernel function to distinguish between the vectors. During the training process, boundaries between the different classes of vectors are calculated. When a new vector has to be classified, the SVM will determine on which side of the boundary it lies and classify the sample accordingly. Note that a SVM is actually designed to distinguish only between two classes, but can be expanded to a more complex problem by combining several classes into a super class, therefore distinguishing between classes through several levels. Since a vector has to be part of one of the two classes, an SVM alone cannot be used for open test sets as all samples will automatically be classified as one of the known target languages.

2.2.3 PERFORMANCE MEASUREMENTS

The performance of the system as a whole is measured by evaluating both the front-end as well as the back-end. The ASR systems in the front-end are evaluated using both the phoneme recognition accuracy and phoneme correctness. Accuracy and Correctness can be defined as follows:

$$\% \text{ Correctness} = \frac{N - D - S}{N} * 100\% \tag{2.1}$$

and

$$\% \text{ Accuracy} = \frac{N - D - S - I}{N} * 100\% \tag{2.2}$$

where N is the total number of labels, D is the number of deletion errors, S is the number of substitution errors, and I is the number of insertion errors.

As for the SVM, the overall accuracy of the S-LID system, as well as the precision and recall for each language is reported. The overall accuracy is simply the percentage of all utterances correctly identified by the SVM. Precision and recall of a specific language (l), are defined as follows:

$$\% \text{ Recall} = \frac{l_{correct}}{l_{correct} + o_{wrong}} * 100\% \tag{2.3}$$

and

$$\% \text{ Precision} = \frac{l_{correct}}{l_{correct} + l_{wrong}} * 100\% \tag{2.4}$$

where $l_{correct}$ is the number of utterances correctly classified as language l and l_{wrong} is the number of utterances incorrectly classified as language l. $o_{correct}$ is the number of utterances correctly classified as the other language and o_{wrong} the number of utterances of the language l incorrectly classified as the other language. l_{wrong} can also be called *false accepts*, and o_{wrong} *false rejects*. ($o_{correct}$ is not used directly in the calculations of precision and recall.) In other words, the precision is the percentage of all test utterances which were classified as language l that are correctly predicted, whereas the recall is the percentage of test utterances that are actually spoken in language l which are predicted correctly.

When Equation 1.1 is considered:

$$P_{Miss} = 1 - \frac{\text{recall}}{100} \tag{2.5}$$

and

$$P_{FA} = 1 - \frac{\text{precision}}{100} \tag{2.6}$$

Note that the precision and recall are calculated on a language-specific basis. The equations 2.3 and 2.4 hold true for a language-pair evaluation, but can be extended to a more complex system by assuming the language o to be a sum of all the incorrect languages in turn.

Since several factors can influence the performance of the ASR systems, such as the insertion factor and the amount of mixtures present in the GMMs, a grid search is used to select the optimal values. Similarly, the SVM results are also optimized using a grid search.

2.2.4 BASELINE PERFORMANCE

The performance of the system as described in sections 2.2.1 to 2.2.2 is provided in Table 2.2. The ASR systems are trained on the full training set as displayed in Table 2.1, whereas the SVM is trained only on a portion of the training data to ensure that the training set is balanced for all languages (approximately 2 500 utterances per language). It should be noted that the performance of the ASR system as given in Table 2.2 can be increased if the HMMs are used in conjunction with a properly generated language model, which is where much of the strength of current ASR systems lie. A flat language model is used as this provides the most language neutral phoneme string for later S-LID classification.

	English	French	Portuguese
Front-end Performance			
Correctness	68.67%	73.88%	52.98%
Accuracy	61.84%	65.90%	34.11%
Back-end Performance			
Precision	99.70%	99.50%	99.80%
Recall	99.70%	100.00%	99.30%
Overall S-LID accuracy : 99.70%			

Table 2.2: The performance of the ASR systems in the front-end as well as the SVM classifier in the back-end of the baseline system.

It can also be seen in Table 2.2 that the SVM classifier is still capable of classifying the languages with a high degree of accuracy, achieving an overall accuracy of 99.7% when classifying the 10 seconds long test utterances of GlobalPhone. Note that though the accuracy compares well with the systems mentioned in Section 1.1.6, this system only distinguishes between three languages instead of the thirteen of the NIST Language Recognition Evaluations of 2007.

2.3 THE EFFECT OF MULTIPLE PHONEME RECOGNIZERS

As stated in Section 1.1, the basic PPR-LM architecture uses a set of tokenizers (in this case phoneme recognizers) to process the audio signal. It is not a requirement that there is a phoneme recognizer for each of the target languages present, though more recognizers are believed to increase the overall performance of a system [19].

This section investigates the effectiveness of a system, which is extended with a new, resource-scarce target language, when only the original tokenizers are used. Attention is given to the influence of an increasing number of tokenizers, therefore the available phoneme recognizers are artificially restricted.

2.3.1 EXPERIMENTAL DESIGN

Using the sufficient resources described in Section 2.2.1, an S-LID system is successfully created to distinguish between English, French and Portuguese with an overall accuracy of 99.7%. This represents our existing system, which is trained on clean and complete resources. For experimental purposes, Japanese, also from the GlobalPhone Corpus, is added to the system. However, a phoneme recognizer is not created for Japanese at this time.

In preparation for the experiment, the Japanese training data is divided into four sets of increasing sizes, with each increased set containing the total previous set. These different sets of Japanese training data is artificially limited to 500, 1 000, 1 500 and 2 500 utterances, each about 8 seconds long. This is done in order to be able to analyze the trends observed when the available training data increases. A test set of 500 utterances is kept aside.

The next step in the experiment involves creating three different environments by artificially restricting the number of available phoneme recognizers. First only English is utilized, then a combination of English-French and finally all three existing recognizers are used. Each of the four Japanese training sets is then combined with a similar amount of training data from the three ‘well-resourced’ languages, and recognized by all three environments. The classifier at the back-end of the system is then retrained with the new training data set, thereby creating twelve different systems. (A different system is created for each combination of the three tokenizer configurations and four training set sizes.) These twelve systems are then tested with a universal test set which incorporates all four languages.

2.3.2 RESULTS

The initial system, utilizing only the English phoneme recognizer, reports an overall accuracy of 93.20% on the smallest training set defined for Japanese. As expected, the performance of the system increases as more phoneme recognizers are added to the front-end, eventually reporting 98.28% for the same training and test set.

Similarly, as more training data becomes available for Japanese, the system’s performance also increases. For the system with only the English recognizer, the performance increases from an overall accuracy of 93.20% to 96.45%, whereas the EFP system increases less dramatically from 98.28% to 99.17%.

Table 2.3 provides a confusion matrix that indicates the performance of all the target languages for the best performing system. This is the EFP system trained on the full Japanese training set. The columns represent the correct language of the utterances whereas the rows represent the language as predicted by the system. The number of correctly classified utterances on the main diagonal of the matrix is boldfaced for clarity.

Figure 2.1 represents the results of this experiment, with graph (a) showing the increase in accuracy performance, and (c) the increase in recall performance. It is interesting to note that the

	English	French	Portuguese	Japanese
English	**99.47%**	0.00%	0.20%	0.00%
French	0.00%	**99.50%**	0.10%	0.00%
Portuguese	0.53%	0.21%	**98.50%**	1.43%
Japanese	0.00%	0.29%	1.20%	**98.57%**

Table 2.3: A confusion matrix which summarizes the performance of the EFP system, when trained on 2 500 utterances per language.

performance increases with smaller intervals as more recognizers are added, and as more Japanese training data becomes available. This suggests that at some point the gain in performance will not be worth the effort of creating another recognizer. Also interesting to note is that Figure 2.1 (b), which plots the precision achieved by the systems, actually reports a decrease in performance.

2.4 CREATING A NEW PHONEME RECOGNIZER WITH INADEQUATE RESOURCES

Now that the effect of an increasing number of tokenizers is better understood, it can be expected that a tokenizer for the new target language will also improve the performance of the system in general. However, the new target language does not have the resources to create an optimal phoneme recognizer. This section investigates techniques to create a tokenizer, as well as the influence such suboptimal tokenizers have upon the system as a whole.

2.4.1 EXPERIMENTAL DESIGN

The next set of experiments continues by adding a Japanese phoneme recognizer to the complete EFP system. Two approaches to the creation of the phoneme recognizer for Japanese are investigated:

- A Japanese phoneme recognizer which is created *orthographically* (EFPJ-orth hereafter) from the available transcriptions. In other words, instead of creating recognizers based on the native phonemes of the new target language, the recognizers are based on the graphemes found in the transcriptions. Therefore the pronunciation dictionary (which is typically required) is not used in this experiment, as the transcriptions are assumed to be spelled phonetically.

- A Japanese phoneme recognizer which is created with transcriptions *bootstrapped* from one of the original phoneme recognizers (EFPJ-boot hereafter). In other words, all the audio data from the new language is recognized by an existing phoneme recognizer (English for this experiment) and the resulting phoneme strings are then used as transcriptions. The existing transcriptions are ignored and since the new transcriptions are already in phonetic format, the pronunciation dictionary is also not required. (In principle, this process can be repeated with

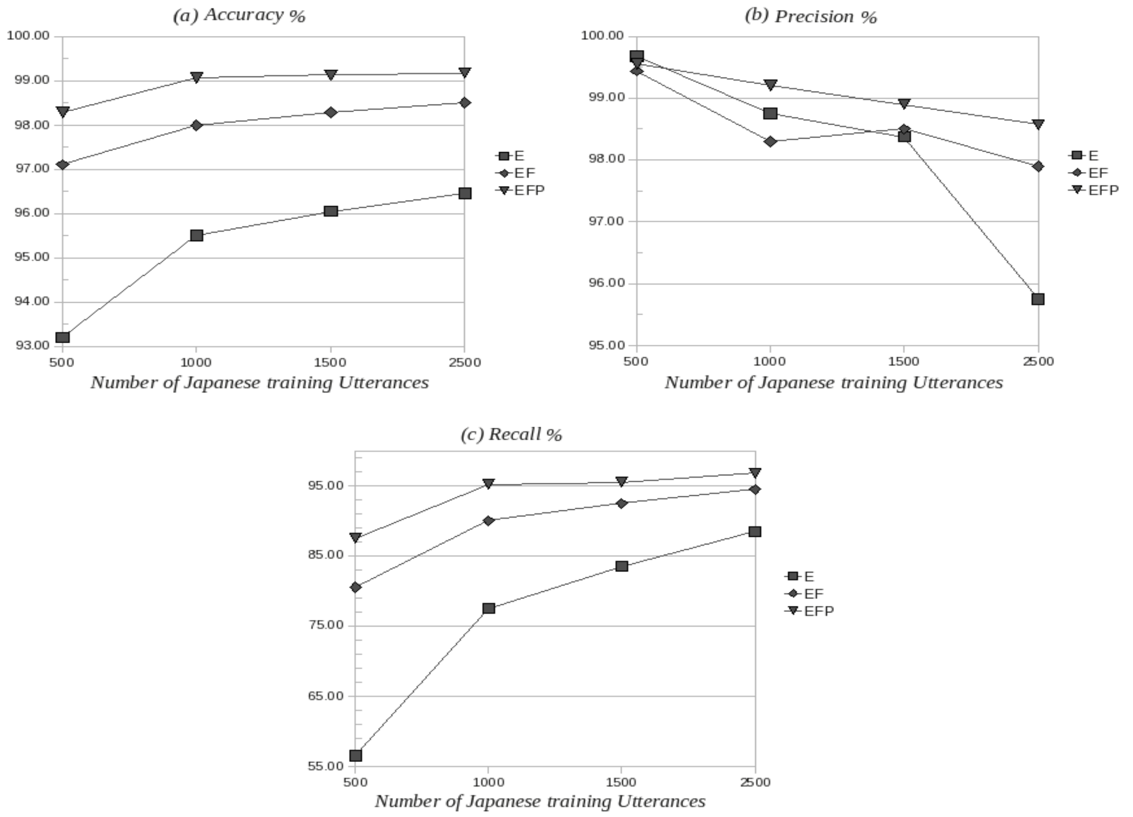

Figure 2.1: Overall Accuracy (a); Precision (b); and Recall (c) of the different S-LID systems when the number of tokenizers are increased from English-Only (E), to English-French (EF) and finally English-French-Portuguese (EFP).

transcriptions repeatedly derived from the updated acoustic models. However for this experiment only one iteration is used.)

These two newly created phoneme recognizers are then employed as tokenizers alongside the existing recognizers. The training sets from the previous experiment as defined in Section 2.3 are used again. Eight more systems (EFPJ-orth and EFP-boot for four training set sizes each), each of which is compared to the best results from the original experiment, are created.

2.4.2 RESULTS

All the results from the new set of systems created for this experiment are compared with the respective results from the best performing system from the initial experiment, namely the EFP system. The results from the three systems, when trained on the smallest training set for Japanese, appear to suggest that the addition of a weaker recognizer does more damage than good to the performance of the system. The original EFP system reports an overall accuracy of 98.28%, whereas both the EFPJ-orth system and the EFPJ-boot system report poorer results with accuracies of 98.25% and 98.09% respectively.

As more data becomes available, the two new configurations finally do overtake the original in performance. Where the EFP system achieves a final accuracy of 99.17% on the complete Japanese training set, the EFPJ-orth system does slightly better with an accuracy of 99.20%. The EFPJ-boot system, on the other hand, outperforms both systems with an accuracy of 99.37%.

Table 2.4 provides an overview of the performance of the Japanese testing set for all three experimental systems. These systems are trained on the complete Japanese training set. Table 2.5 provides a confusion matrix that stipulates the performance of all the target languages when classified with the EFPJ-boot system, which is trained on the entire Japanese training set. The columns represent the correct language of the utterances whereas the rows represent the language as predicted by the system. The number of correctly classified utterances on the main diagonal of the matrix is boldfaced for clarity.

System	Overall Accuracy	Japanese Precision	Japanese Recall
EFP	99.17%	98.57%	96.79%
J-orth	99.20%	99.38%	97.19%
J-boot	99.37%	99.39%	97.59%

Table 2.4: SVM performance for three of the systems, when 2 500 utterances per language are used to train the system.

It is interesting to note that, whereas the EFPJ-boot system performs better than the EFP system almost immediately, the EFPJ-orth system is only able to do so at the very end of the experiment. Therefore, though the addition of the orthography-based Japanese recognizer does not seem to im-

	English	French	Portuguese	Japanese
English	**99.60%**	0.10%	0.20%	0.20%
French	0.00%	**99.50%**	0.00%	0.20%
Portuguese	0.30%	0.40%	**99.80%**	2.10%
Japanese	0.10%	0.00%	0.10%	**97.50%**

Table 2.5: A confusion matrix which summarizes the performance of the EFPJ-boot system, when trained on 2 500 utterances per language.

prove the performance of the system as a whole, the results from the second configuration indicate that a bootstrapped acoustic model can in fact be used to increase the accuracy of an existing system.

Figure 2.2 represents the results of this experiment, comparing performance of the three different configurations in terms of (a) overall accuracy, (b) precision and (c) recall.

2.5 CONCLUSION

The experiments in this chapter show that adding more tokenizers to the front-end of an existing S-LID system does increase the performance of the specific system. Even phoneme recognizers for languages with limited available resources can indeed be added successfully to a phonotactic S-LID system with almost no loss in accuracy to the existing target languages with abundant resources.

The bootstrapping technique which is described in this section makes it possible to create phoneme recognizers for languages for which neither transcriptions nor pronunciation dictionaries can be acquired. Though the bootstrapping of new transcriptions works better than creating an orthographic system, it still proves only beneficial once the resources are above a certain threshold (more than 1 000 utterances in this experiment), otherwise such tokenizers may actually be detrimental to the system's performance. Still, as the last experiment in this chapter shows, the addition of such tokenizers when enough resources are available can be beneficial to the system as a whole, though they are not as effective as when the same language is fully resourced.

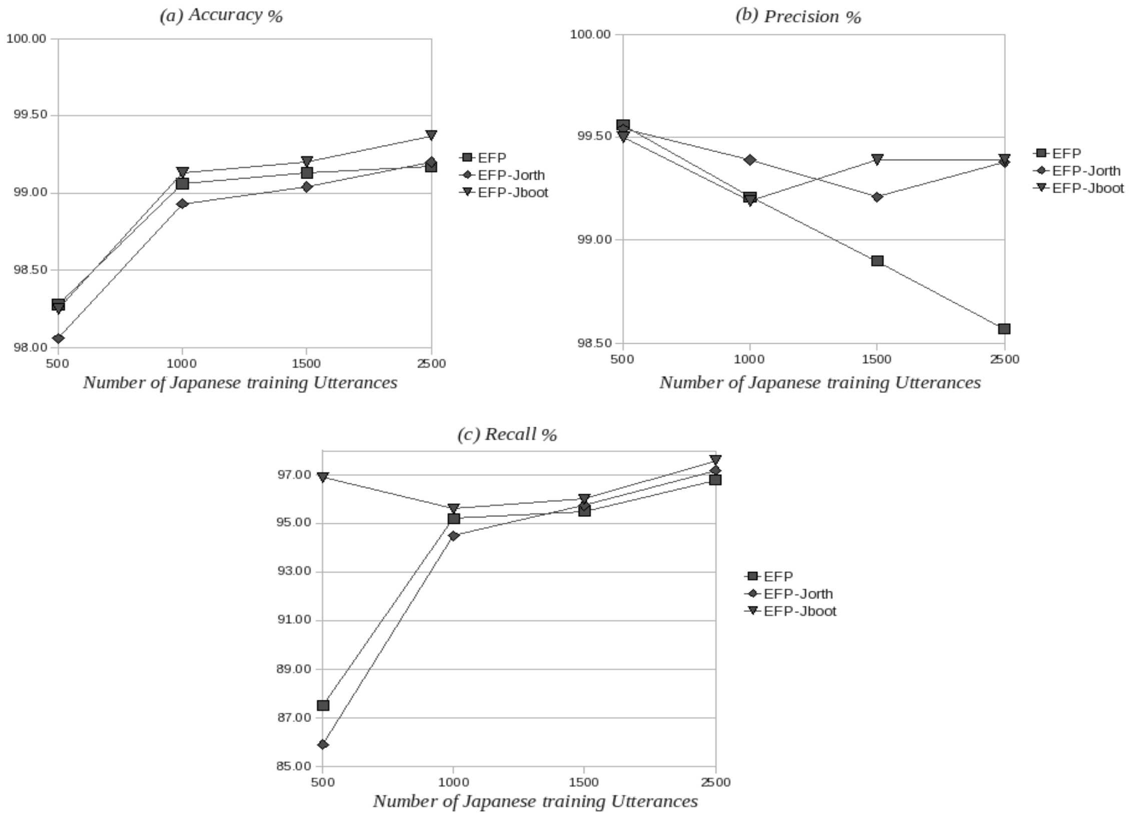

Figure 2.2: Overall Accuracy (a); Precision (b); and Recall (c) of S-LID systems when a Japanese tokenizer is added Orthographically (J-orth) or with bootstrapped transcriptions (J-boot) when compared to the English-French-Portuguese-Only (EFP) system.

CHAPTER THREE

MISMATCHED RESOURCES

3.1 INTRODUCTION

In a resource-scarce environment, it is often required to use data from different sources in order to develop a functional Spoken Language Identification (S-LID) system. An existing system, which is trained on a corpus of carefully selected data, may also need to be ported to a different environment for which data is more limited in quantity and may possibly be of poorer quality.

This chapter investigates the process of porting an existing S-LID system, which was developed in a well-resourced environment, to a different, under-resourced environment. It describes methods to prepare the system for more effective use with unmatched data, possibly of poorer quality.

This chapter is structured as follows: Various causes of data mismatch are explored in Section 3.2. An overview of the experimental design is provided in Section 3.3. A detailed discussion of the various main steps then follow: Data preparation (Section 3.4), initial adaptation, including retraining of the classifier at the back-end (Section 3.5) and the final retraining of the entire system in Section 3.6. Conclusions are summarized in Section 3.7.

3.2 CAUSES OF DATA MISMATCH

There is a variety of reasons why data can differ from one corpus to a next. These can be summarized using the following categories:

Channel conditions. Channel conditions depend on the physical environment in which the audio data is recorded. The *recording medium* has a major influence, for instance using a microphone will produce clearer sounds than recording files over a telephone. *Sound pollution*, such as *background noise* and other *non-speech signals* also play an important role in the quality of the audio (consider studio conditions vs a public environment). The *recording conditions*, such as sampling frequency, bandwidth and volume are also important to consider. The *encoding protocol* (mono vs. stereo as an example) is also important, as the compression of the analog signal into electronic data can change the acoustic characteristics of the speech sample.

Speaker conditions. The demographic variables are also important. *Dialects*, *strong accents* and *second-language speakers* have a strong influence on the performance of the ASR systems as the same words are pronounced with different phonemes. Speaker characteristics such as age and gender also have an effect.

Domain conditions. The *vocabulary* can also be considered to be responsible for differences in corpora, as different terminology is used in different domains. The speaking style is also important as people's continuous speech is spoken differently from speech produced by reading, for example.

3.3 EXPERIMENTAL SETUP

This section examines the design of the system used for the experiments that are found in the rest of this chapter. The same setup as described in Section 2.2 is also used for the experiments in this chapter. A more complete discussion of the corpora used in this chapter is supplied in Sections 3.3.1. After a brief overview of the system design in Section 3.3.2, the performance of the baseline system for this Chapter is provided in 3.3.3.

3.3.1 CORPORA

The data from the GlobalPhone [30] as well as the Wall Street Journal [31] corpora is used to represent the cleaner, well-resourced environment from which an existing S-LID system is to be ported. Alongside these two corpora, an in-house corpus (which will be referred to as the African corpus) will be used to represent the under-resourced environment. However, all the audio data from the well-resourced corpora is downsampled from 16kHz to 8kHz in order to make it comparable to the poorer African corpus.

Since statistics of the cleaner corpora are listed in Table 2.1, only the statistics of the African corpus are listed here in Table 3.1. The number of utterances is given, as well as the length of all the audio data for each language. The difference between the training and test sets is also shown.

Unfortunately, the African corpus does not have any transcriptions. Therefore, the technique of bootstrapping transcriptions, as defined in Chapter 2, is utilized in this chapter as well. Since the audio data of the African corpus also does not have any speaker information, the assumption is made that one channel corresponds to one unique speaker, and that one speaker is not shared between different utterances. While these assumptions are valid for the majority of the utterances, a few individual exceptions do occur. The utterances of the African corpus are split into different channels, in order to

Language	Set	Utterances	Hours
English	Train	250	16.77
	Test	25	4.30
French	Train	109	8.25
	Test	26	2.07
Portuguese	Train	108	11.18
	Test	28	2.84

Table 3.1: Statistics on the training and testing sets for each language in the African corpus.

	English	French	Portuguese
	Front-end Performance		
Correctness	64.68%	74.09%	55.43%
Accuracy	52.81%	66.19%	48.10%
	Back-end Performance		
Precision	98.91%	94.87%	97.66%
Recall	99.26%	98.99%	90.60%
Overall S-LID accuracy	97.18%		

Table 3.2: The performance of the ASR systems in the front-end as well as the SVM classifier in the back-end of the baseline system, which is developed with the downsampled GlobalPhone corpus.

separate the different speakers from each other. These channels are then further separated into minute long segments.

3.3.2 SYSTEM DESIGN

The system implements the popular Parallel Phoneme Recognition followed by Language Modeling (PPR-LM) architecture [15]. Phoneme recognizers for all three target languages are trained on the cleaner GlobalPhone corpus. These phoneme recognizers utilize biphone-based Hidden Markov Models (HMM) as in Chapter 2. The HMMs also consist of three emitting states, with Gaussian Mixture Models (GMM) of four mixtures in each state. A Support Vector Machine (SVM) at the back-end classifies the languages based on biphone frequencies.

For a more complete description, please refer to Section 2.2.2.

3.3.3 BASELINE PERFORMANCE

The performance of the baseline system for this chapter's experiments is provided in Table 3.2, which lists the performance measurements of the system which is developed with the downsampled GlobalPhone corpus. The results are comparable to that of the baseline system used in Chapter 2, though the training and test sets are not exactly the same.

3.4 DATA PREPARATION

Preprocessing of data from two completely different environments is a vital step which must be completed before any attempt to combine resources can even be considered. If this is not done, the system may be biased towards the acoustic characteristics of the corpus with the most audio data. For languages with limited resources, the available corpora may even be of such poor quality that it is rendered useless unless some preprocessing steps are taken.

The audio data from the African corpus also contains not only speech, but telephonic signals and in many cases long stretches of silence and background noise as well. The signal quality itself is also much poorer than that of GlobalPhone, so it is critical that some preprocessing takes place.

The preprocessing process takes place as follows:

- Common diarization techniques are used to separate the different telephonic channels from each other. The diarization also removes any non-speech signals and long sections of silence from the audio.
- Amplitude normalization is also applied to the new data to diminish the effect of background clicks and other channel related noises that the diarization is unable to remove.
- Channel normalization is applied to the GlobalPhone data in order to approximate the channel conditions to the African corpus, now that the African corpus is prepared for use.

3.4.1 DIARIZATION OF AUDIO DATA

Diarization is a preprocessing step which can be applied to the audio data of a poor quality corpus. It is used to separate the speech signals from any other competing noises that may be found in the sound files, or the lack thereof in the case of silence removal. It is a very useful tool to improve the quality of the audio data before any training or testing is performed.

A diarization module can be described as a series of units that process the sound files in a series of steps. The first step is to remove any long silences and for the experiment in this chapter this is accomplished with the “praat” toolkit [36]. The silence removal is followed by segmentation, during which the stream of audio is segmented into different parts. Speech and non-speech segments are most commonly used. This is done with the Bayesian Information Criterion (BIC). BIC simply looks for differences in the covariance matrix and segments the audio on large changes. These segments are then passed to the last step of the diarization process, where a classifier determines the type of each segment based on the pitch information.

In this chapter, the different channels found in the sound files of the African corpus is separated before the audio data is run through the diarization process. An SVM is used in the final classification to distinguish between speech and non-speech segments.

3.4.2 CHANNEL NORMALIZATION OF AUDIO DATA

When audio data is recorded under different channel conditions, it will lead to a mismatch between the acoustic characteristics of the audio data. Therefore the technique of channel normalization can be applied in order to minimize the spectral differences.

Channel normalization is performed by estimating the magnitude spectrum for each frame across each data set. A map filter is then calculated which can be applied across each frame of the environment that has to be normalized in turn, so that the channel characteristics of the one channel is also visible in the other.

Channel normalization can be applied in either direction. In this chapter, the cleaner GlobalPhone data is channel normalized to the weaker African corpus. This is done because it is easier and more effective to approximate the clean data to the corpus that is littered with noise, than to try and clear the noise from the weaker corpus.

Amplitude normalization on the other hand, scales each utterance's sample values so that the absolute maximum value is one and thus is used to reduce the effect of any extreme spikes in the raw wave-form. Unlike channel normalization it is performed on a per-file basis and utterances do not influence each other, even within the corpus.

3.5 INITIAL PORTING OF THE S-LID SYSTEM

This section investigates the initial effects of adapting the system to the new environment. The overall performance of the system, when only the original audio data is adapted to resemble the new environment, is examined. The results achieved when audio data from the new environment is classified with this new system is also reported on, both for the system as it is, as well as after the system has been adapted initially.

3.5.1 ORIGINAL TRAINING DATA ADAPTATION

With the data from the new corpus preprocessed, the audio data from the original corpus is also adapted to correlate more accurately with the channel conditions of the African corpus. As already stated, the African corpus is used alongside the much cleaner GlobalPhone corpus. Therefore, all the audio data from the cleaner corpora is downsampled from 16kHz to 8kHz in order to make it comparable to the African corpus. The original data is also amplitude normalized and then channel normalized in order to better match the new environment [37].

Verification reveals that the new system, which is retrained on the adapted GlobalPhone data, performs relatively similar to the original downsampled system, when tested on the adapted GlobalPhone test data. However, as can be seen from Figure 3.1 the system performance drops considerably when data from the adapted African corpus is used to test the system. This clearly shows that only approximating the data used to train a system is not effective at all. The system itself must also be adapted to the new corpus before it can be used in the new environment.

	English	French	Portuguese
Front-end Performance			
Correctness	56.00%	62.06%	45.91%
Accuracy	45.52%	52.38%	38.95%
Back-end Performance			
Precision	99.37%	95.06%	99.19%
Recall	99.64%	99.52%	94.50%
Overall system accuracy : 98.25%			

Table 3.3: The performance of the ASR systems in the front-end as well as the SVM classifier in the back-end of the system which is trained on the downsampled data, after channel normalization has been applied.

A more detailed examination of the results follows in Section 3.5.3, after classifier adaptation is also considered.

3.5.2 CLASSIFIER ADAPTATION

The first logical attempt to adapt the system for use in the new environment is to retrain the back-end. Retraining of the classifier recalculates the boundaries of the target languages within the vector space, according to the new set of audio data. The adaptation of the classifier is achieved by recognizing the audio data from the African corpus with the latest phoneme recognizers which are trained on the adapted audio data of the GlobalPhone corpus. The resulting phone strings are then used to retrain and test the SVM. As can be seen in Figure 3.1, even the basic adaptation of the system achieves improved results. A more detailed examination of the results now follows in Section 3.5.3.

3.5.3 ANALYSIS

Though the audio data from the GlobalPhone corpus is normalized to approximate the channel conditions of the much poorer quality African corpus, the system's performance is not greatly effected when tested on data from the original environment. As can be seen from Table 3.3, the system as a whole performs nearly on par with the original baseline system described in Section 3.3. While ASR accuracy decreases, the performance of the system, which is now trained on the newly adapted audio data, remains high, reporting an overall accuracy of 98.26%. Table 3.3 gives a more complete overview of system performance, both for the front-end and the back-end.

Unfortunately, though the system is now trained on data which approximate the channel conditions of the new environment, performance is still poor when test data from the African corpus is used. A poor accuracy of 47.02% is reported when no additional adaptation is applied to the existing system. Table 3.4 gives a more complete overview of the performance of the back-end after the initial adaptation when tested with data from the African corpus.

When the back-end of the system is also adapted to the new environment (i.e after retraining the

	English	French	Portuguese
Precision	56.27%	42.20%	36.46%
Recall	57.14%	37.10%	38.82%
Overall system accuracy : 47.02%			

Table 3.4: The performance of the SVM classifier at the back-end of the system when test data from the African corpus is used using the same system as Table 3.3.

	English	French	Portuguese
Precision	64.71%	62.38%	58.22%
Recall	76.45%	50.81%	50.00%
Overall system accuracy : 62.56%			

Table 3.5: The performance of the SVM classifier at the back-end of the system when test data from the African corpus is used after the SVM itself is retrained.

classifier with phone strings generated by the adapted back-end), the performance does increase again to a accuracy of 62.57%. Though still quite low, it is a promising improvement when compared to the initial attempt to identify the languages spoken in the African corpus. Table 3.5 gives a more complete overview of the performance of the back-end after further adaptation.

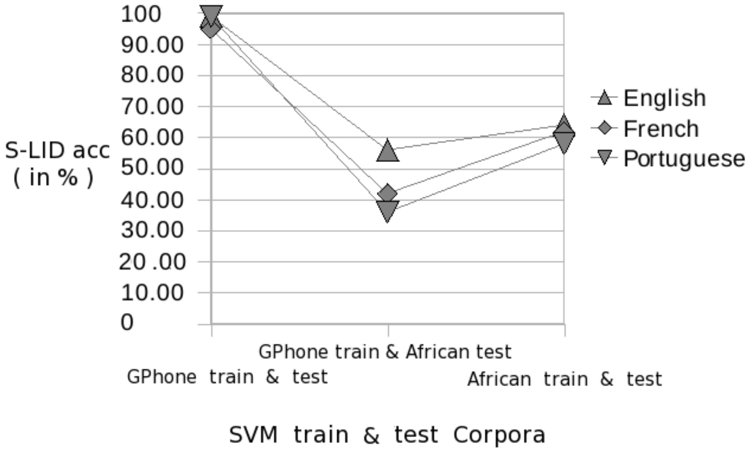

Figure 3.1: SVM performance with different train and test sets when the phoneme recognizers are trained only on the downsampled GlobalPhone corpus, and the SVM is trained and tested on sets from the specified corpora.

Figure 3.1 depicts the overall performance of the SVM at the back-end in percentage accuracy for the specified systems graphically.

3.6 FURTHER ADAPTION OF THE S-LID SYSTEM

With the promising improvement achieved by retraining the classifier, this section now explores a more complete porting of the existing system to the new environment. Attention is given to the adaptation of the ASR systems at the front-end of the system.

3.6.1 PHONEME RECOGNIZER ADAPTATION

In order to further improve the performance of the system, attention is given to the front-end, in combination with the adaptation of the back-end. Though the current phoneme recognizers are trained with data that is approximated to the new channel conditions, the data remained different from the African corpus with regard to speaking style, domain and speaker conditions. Note that in Section 3.5.1, the system performed on par with the original. Therefore it is logical to assume that adapting the recognizers to the new environment will improve system accuracy within this environment.

Adaption is performed by retraining the phoneme recognizers, using only audio data from the African corpus. As there is no available transcriptions in the new environment, the bootstrapping technique from Section 2.4.1 is implemented to generate new transcriptions for the audio data. After the new phoneme recognizers are created, the classifier is retrained again, as is done in Section 3.5.2.

The results achieved by the new system are disappointing, as the accuracy falls slightly to 60.58%. Even the phoneme recognizers at the front-end are struggling. This is believed to be caused by audio data of much poorer quality found in the new environment, as some of the utterances appear to have a very poor phoneme recognition. Table 3.6 gives a more complete overview of the performance for both the front-end and the back-end.

	English	French	Portuguese
Front-end Performance			
Correctness	16.99%	23.75%	37.30%
Accuracy	11.04%	17.71%	28.07%
Back-end Performance			
Precision	60.25%	69.23%	57.55%
Recall	91.89%	29.03%	35.88%
Overall system accuracy : 60.57%			

Table 3.6: The performance of the ASR systems in the front-end as well as the SVM classifier in the back-end of the system which is trained on data from the African Corpus.

3.6.2 ADAPTATION THROUGH FILTERING OF THE PHONEME STRINGS

A technique to automatically filter out these segments of poor quality audio is developed: All segments that recorded a small frequency of phonemes are removed from the train and test sets. As can be seen from Table 3.7, this again improves the overall performance of the system as results peak at an

	English	French	Portuguese
All utterances $<$ 0.5 phonemes/sec			
Front-end Performance			
Correctness	32.34%	32.20%	38.36%
Accuracy	25.76%	24.20%	30.73%
Back-end Performance			
Precision	63.31%	58.51%	65.00%
Recall	79.70%	44.35%	56.52%
Overall system accuracy : 62.86%			
All utterances $<$ 1.0 phonemes/sec			
Front-end Performance			
Correctness	33.10%	32.69%	37.94%
Accuracy	26.77%	25.35%	31.18%
Back-end Performance			
Precision	65.75%	68.97%	66.67%
Recall	86.75%	50.00%	57.34%
Overall system accuracy : 66.66%			
All utterances $<$ 1.5 phonemes/sec			
Front-end Performance			
Correctness	32.71%	31.79%	38.03%
Accuracy	26.59%	24.40%	31.49%
Back-end Performance			
Precision	67.36%	73.75%	68.07%
Recall	83.87%	54.63%	62.79%
Overall system accuracy : 68.87%			
All utterances $<$ 2.0 phonemes/sec			
Front-end Performance			
Correctness	40.47%	40.02%	44.02%
Accuracy	34.15%	33.96%	36.53%
Back-end Performance			
Precision	64.95%	77.78%	69.64%
Recall	85.71%	52.83%	62.4%
Overall system accuracy : 68.78%			

Table 3.7: The performance of the ASR systems in the front-end as well as the SVM classifier in the back-end. The system is trained on data from the African corpus, after all utterances with less than the given amount of phonemes per second is removed from both the training and test sets.

overall accuracy of 68.88%. Figure 3.2 depicts the overall performance of the SVM at the back-end in percentage accuracy for the specified systems graphically.

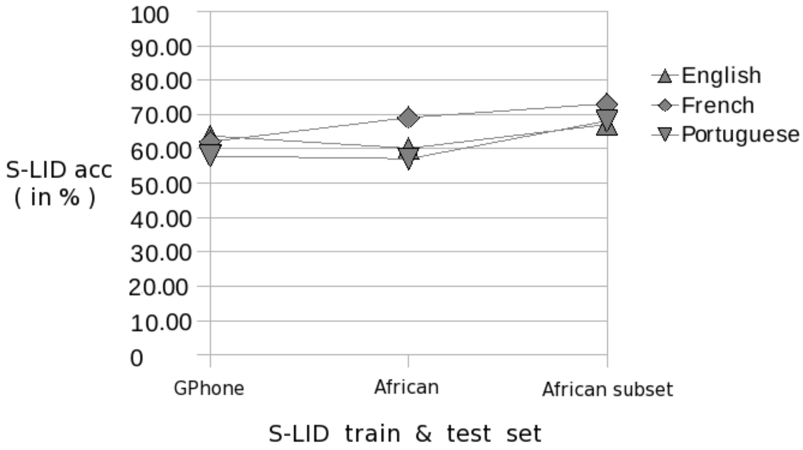

Figure 3.2: Overall SVM performance when the phoneme recognizers is trained on the different corpora.

3.6.3 DISCUSSION

Though increasing the accuracy from 47.02% to 68.88% is a great improvement, the current system still does not perform sufficiently to suggest that it is ported to the new environment successfully. Therefore further analysis of the new environment is required.

This is done by having human listeners systematically listen to random subsets of the African corpus, focusing particularly on the segments which were classified incorrectly by the system. The human verifiers identified the following problematic subsets within the corpus:

- Audio data labeled with the incorrect language.
- Audio data of such poor quality or with such a level of competing noise that the speech is barely audible and is rendered unusable. This, together with the incorrectly labeled data is included in the *Unusable* subset.
- Correctly labeled data, but spoken with a strong accent. This is included in the *Accented* subset.

Though not always as clean as the GlobalPhone corpus, all the usable segments are included in the *Correct* subset. Table 3.9 gives a brief summary of the findings of the human verifiers, showing the percentage of audio segments for each subset per language. The surprisingly high percentage of *Unusable* data present within the subset given to the human verifiers may be the reason for the initial fall in performance of the system when the phoneme recognizers are adapted, as well as be responsible for the poorer than desired results in the end.

Language	Unusable	Accented	Correct
English	36.45%	24.08%	39.46%
French	40.29%	1.46%	58.24%
Portuguese	24.14%	0.68%	75.17%

Table 3.8: Percentage of samples per language in each category, as reported by human verifiers.

If the system is used as developed in Section 3.6.2 and only the "correctly" labeled utterances are recognized, a different picture emerges, with an overall accuracy of 80.30% affirmed. The entire adaptation process is summarized in Table 3.9, displaying results on the African test corpus.

	Language	GlobalPhone channel normalization	SVM retrained	Recognizer adaptation	Recognizer adaptation after filtering	Tested on correct subset
Phoneme Recognition Accuracy (in %)	English French Portuguese	45.52 52.38 38.95	= = =	11.04 211.71 28.07	26.59 24.40 31.49	= = =
SVM Classification Precision (in %)	English French Portuguese	56.27 42.20 36.46	64.71 62.38 58.22	60.25 69.23 57.55	67.36 73.75 68.07	71.43 78.70 78.06
SVM Classification Recall (in %)	English French Portuguese	57.14 37.10 38.82	76.45 50.81 50.00	91.89 29.03 35.88	83.87 54.63 62.79	74.16 76.53 76.82
Overall Accuracy(in %)		47.02	62.56	60.57	68.87	80.30

Table 3.9: The entire adaptation process.

3.7 CONCLUSION

This chapter demonstrates that an S-LID system cannot function properly outside its original environment without some adaptation. Techniques to port an existing S-LID system to a much poorer environment is described and the feasibility of these techniques are demonstrated.

The importance of verifying the quality of the data from the new environment before any adaptation is attempted, is also emphasized. The quality of the audio data can be verified by examining the phoneme strings that are produced by the front-end. Thereafter poorly performing training data can be automatically filtered out before the adaption process continues. In theory, this can be improved by iterating through the process, but for this experiment only one iteration was applied.

The preparation of the data is an important step before an S-LID system can be ported to a new environment. Experiments also show in particular that though a great improvement in system performance can be gained by retraining the classifier, the best result is still achieved by retraining both the tokenizers as well as the classifier.

CHAPTER FOUR

SUBOPTIMAL RESOURCES

4.1 INTRODUCTION

When a Spoken Language Identification (S-LID) system needs to be created for a completely new environment, new linguistic resources have to be gathered before speech processing tools can be created. However, as was seen at the end of Chapter 3, it is often the case that during the initial stages of this development, the available data in a new corpus may be of very poor quality.

This chapter investigates the effect that resources of suboptimal quality have on the overall performance of an S-LID system. Specifically the effect of transcription errors is investigated. This chapter is structured as follows: An overview of the setup of the experiments for the rest of the chapter is provided in Section 4.2. Section 4.3 describes the experiment itself before a conclusion is provided in Section 4.4.

4.2 EXPERIMENTAL SETUP

This section examines the design of the system used for the experiments in the rest of this chapter. The setup used for these experiments is very similar to that of previous chapters. However, the ASR systems at the front-end are implemented a bit differently than for previous experiments.

This section is structured as follows: The Meraka Lwazi corpus is discussed in Section 4.2.1, and the difference between the two versions of the isiZulu corpora discussed in Section 4.2.2. Section 4.2.3 explains the setup of the experiment, and the performance of the baseline system is provided in Section 4.2.4.

4.2.1 CORPUS STATISTICS

This chapter uses the Meraka Lwazi corpus [32]. The Meraka Lwazi corpus was still under development at the time of this research, but is earmarked for public release. It was jointly developed by the Meraka Institute and by the North-West University for the Lwazi project, sponsored by the Department of Arts and Culture of the Government of South Africa.

The Meraka Lwazi corpus consists of telephonic audio data, which is collected in all eleven of South Africa's official languages. Therefore, unlike the previous chapters, the target languages are now South African languages. For this chapter specifically, Afrikaans, SA English, Setswana and isiZulu are selected. Table 4.1 gives some statistics on the available data. The number of speakers per language is given, as well as the combined number of utterances and the length of the audio data in hours. The difference between the train and test set is also provided.

Language	Set	Speakers	Utterances	Hours
Afrikaans	Train	170	4 383	3.32
	Test	30	787	0.60
SA English	Train	175	4 287	3.25
	Test	30	728	0.55
Setswana	Train	176	4 257	3.23
	Test	30	686	0.52
isiZulu	Train	170	4 091	3.10
	Test	30	711	0.54

Table 4.1: Statistics on the training and testing sets for each of the languages used in this chapter.

Unlike the African corpus used in Chapter 3, the audio data from the Lwazi corpus is clearly audible, and background and other non-speech noises have been kept to an absolute minimum.

4.2.2 DIFFERENCES IN ISIZULU CORPORA

After collection, audio data is curated and transcribed by human annotators, as described in more detail in [27]. Annotation is not error-free, and the corpus undergoes a number of quality verification cycles prior to completion. The results from two such cycles are selected - an initial version and a later verified version - in order to compare the effect of the poorer transcriptions on system performance. Table 4.2 gives a brief summary of the differences between the original and the refined isiZulu corpora.

Table 4.2 describes the differences between the two versions of the isiZulu corpus. It lists the number of files per category as a percentage of the refined isiZulu version. Different levels of alterations concerning the transcriptions are given attention to, namely minor changes (which mostly concerns small changes such as correction of spelling errors and other changes of two or less characters per sentence), noise-tag reduction (correction of previously erroneous noise-tag markings as well as removal of markings referring to noise now removed from actual audio files during the latest

Transcription Changes	
Major Changes	16.91%
Noise-tag Reduction	16.45%
Minor Changes	35.16%
Unchanged	28.89%
Audio Changes	
Major Changes	28.17%
Minor Changes	44.75%
Unchanged	24.48%
File Changes	
Files Gained	2.59%
Files Lost	2.09%

Table 4.2: A brief summary of the differences between the original and the refined isiZulu corpora.

verification) and major changes (such as truncating words at the end of the utterance which are not completely pronounced in the audio and other changes of three or more characters per sentence). Major changes in the audio files include all files that have been altered to differ in size with more than 10% (of the largest file between versions) and mostly consists of the removal of operator interruptions. Minor changes consists mostly of shifting utterance boundaries in order to complete final words in utterances which had been cut off too soon. It should also be noted that about 2% of the files do not have corresponding counterparts in the other version. Since both versions are available, isiZulu will be used as the new target language which is being added to the system and the effects of better and poorer transcriptions can be compared directly. Five isiZulu corpora are defined, with varying percentages of the different versions of isiZulu data. These corpora are explained in Table 4.3. It should be noted that the number of training utterances remains the same for each of the corpora, and that speaker distribution remains constant across the artificial corpora.

isiZulu Corpus	Original Version	Refined Version
Corpus-A	0%	100%
Corpus-B	25%	75%
Corpus-C	50%	50%
Corpus-D	75%	25%
Corpus-E	100%	0%

Table 4.3: A number of artificial isiZulu corpora are created by combining different percentages from the original (poor quality) and the refined (better quality) corpora.

As can be seen from table 4.3, corpus-A is by definition the refined version whereas corpus-E remains the original version.

4.2.3 BASELINE SYSTEM DESIGN

The system implements the popular Parallel Phoneme Recognition followed by Language Modeling (PPR-LM) architecture [15]. Automatic Speech Recognition (ASR) systems for all target languages are trained, and again utilizes biphone-based Hidden Markov Models (HMM) for acoustic models. The HMMs consists of three emitting states as for the rest of the thesis, but the number of mixtures of the GMMs in each state have been increased to seven mixtures. An SVM at the back-end classifies the languages, as in the rest of the thesis.

For a more complete explanation of the rest of the system, please refer to section 2.2.2.

4.2.4 BASELINE PERFORMANCE

The baseline system for this chapter performs satisfactorily. As can be seen from Table 4.4, the ASR systems perform on par and above with other phoneme recognizers in the rest of the thesis. The SVM itself also performs well, achieving an overall accuracy of 85.13%. This is lower than any of the baseline systems used in previous chapters because the Meraka Lwazi corpus consists of much shorter audio segments, as can be seen in the figures of Appendix 1.

	Afrikaans	SA English	Setswana
Front-end Performance			
Correctness	70.03%	57.87%	66.89%
Accuracy	65.16%	52.76%	55.32%
Back-end Performance			
Precision	81.79%	84.53%	89.07%
Recall	86.79%	81.04%	86.73%
Overall system accuracy : 84.87%			

Table 4.4: The performance of the ASR systems in the front-end as well as the SVM classifier in the back-end of the baseline system.

4.3 THE EFFECT OF SUBOPTIMAL TRANSCRIPTIONS

The purpose of this section is to examine the effect of suboptimal data on the performance of an S-LID system. This is done by building different S-LID systems, each with a set of data that increases in quality.

Details of the experiment are provided in Section 4.3.1. The results of both the ASR systems and the SVM are examined separately in Section 4.3.2 and Section 4.3.3 respectively, and a discussion then follows in Section 4.3.4.

4.3.1 EXPERIMENTAL DESIGN

Section 4.2.2 defines five isiZulu corpora, with varying percentages of suboptimal and refined data. The corpus which contains only data from the refined version is labeled as *corpus-A*. From *corpus-A* the data is replaced with data from the original isiZulu version, 25% at a time until the entire original version is defined as *corpus-E*.

Corpora *A* to *E* are further divided into three sets with increasing amounts of training data, with each smaller set being a subset of the larger ones. The smallest set only contains 25% of the training data, with the second set containing 50% of the training data, and the largest set containing the entire training set. Therefore, a total of 15 S-LID systems are created.

The training sets which are used to train the ASR systems, are also artificially restricted to 1 020, 2 039 and 4 091 training utterances (numbers respectively equal to the different isiZulu training set sizes) for the other three languages when used to train the SVM classifier. This ensures a balanced classifier at the back-end. Each of these 15 S-LID systems is then tested using only one, predefined test set from *corpus-A*. The test set remains the same for both the ASR systems and the SVM classifier.

4.3.2 ASR RESULTS

The isiZulu phoneme recognizers also perform on par with other phoneme recognizers used in the rest of this thesis. The correctness of the recognizer achieves a percentage of 66.38%, and the accuracy a percentage of 58.31%. Table 4.5 provides a more complete performance of the ASR systems in the front-end of the S-LID systems, according to the training data used.

	25% Training Data	50% Training Data	100% Training Data
isiZulu Corpus A			
Correctness Accuracy	60.43% 54.00%	65.21% 57.02%	66.38% 58.31%
isiZulu Corpus B			
Correctness Accuracy	62.08% 54.00%	64.83% 56.99%	66.37% 58.38%
isiZulu Corpus C			
Correctness Accuracy	61.57% 53.87%	64.69% 55.78%	66.00% 58.03%
isiZulu Corpus D			
Correctness Accuracy	58.87% 52.23%	63.03% 55.72%	65.40% 56.77%
isiZulu Corpus E			
Correctness Accuracy	59.23% 52.14%	63.12% 55.93%	63.71% 56.67%

Table 4.5: The performance of the ASR systems in the front-end of the S-LID systems, according to the training data used.

As can be seen from Table 4.5, the performance of the ASR systems continue to increase as more data is added to the training set, regardless of which isiZulu corpus is used. Comparing the performance based on the isiZulu corpora, it is also clear that *corpus-A* outperforms *corpus-E* on all training set sizes. The isiZulu corpora *B* to *D* is a bit more ambiguous, though a clearly increasing trend can still be seen in performance as the corpus used becomes more refined. In summary, while the effects of the improvements to the corpus is visible, this is much less then initially anticipated.

4.3.3 SVM RESULTS

As mentioned in Section 1.1.5, the addition of isiZulu to the baseline system is expected to decrease the performance of the SVM classifier. Though the S-LID systems with the smallest training sets report an overall accuracy of between 68.37% and 69.09%, the systems with the full training sets still manage to achieve an average overall accuracy of between 80.66% and 80.97%. Table 4.6 provides a more complete overview of the performance of the ASR systems in the front-end of the S-LID systems, according to the training data used. The precision as well as the recall for each language, grouped according to both the training set size and the isiZulu corpus used to train the system, is provided.

It should be noted in Table 4.6 that the performance of isiZulu remains on par with Afrikaans, SA English and Setswana. Though all the S-LID systems appear to be unaffected by the different corpora, the size of the training sets clearly influence the system as a whole.

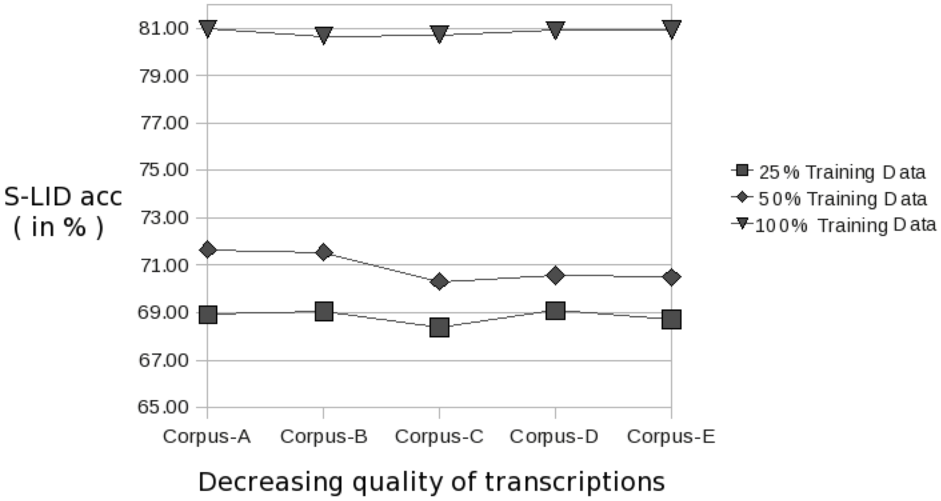

Figure 4.1: Overall SVM accuracy when the phoneme recognizers are trained on the different corpora.

Figure 4.1 represents the overall performance of the S-LID systems individually. As can be seen from Figure 4.1, the performance of the SVM classifiers also continue to increase as more data is added to the training set, regardless of which isiZulu corpus is used. Though the performance of the *corpus-A* SVM still outperforms the *corpus-E* SVM, the results remain very similar across the different isiZulu corpora.

4.3.4 DISCUSSION

The varying quality of the data has an effect on the ASR systems of the front-end, as Section 4.3.2 reveals but this effect is smaller than anticipated, given the large number of refinements implemented. Fortunately, due to the strength of the SVM at the back-end the difference in ASR accuracy has limited effect on the S-LID accuracy. It is clear that more data improves the performance of the system for each of the corpora defined and that, while the refined corpus does perform better than the original suboptimal data, this effect is minimal.

4.4 CONCLUSION

This chapter demonstrates that some transcriptions of suboptimal quality do not influence the overall performance of an S-LID system greatly. Though the ASR systems at the front-end do report a visible improvement in performance as the transcriptions are refined, the SVM classifier at the back-end appears to be resilient enough to function properly with transcriptions of varying quality. In particular, small decreases in performance suffered by the ASR systems are not carried through to a decrease in performance of the system as a whole. This implies that the quantity of audio data is more important than the quality of the transcriptions (a result that reinforces the conclusion reached in Section 2.4).

It should be kept in mind that Chapter 3, on the other hand, reveals that quality of the audio data and label accuracy have a more substantial effect on the system as a whole. It remains important to verify the quality of the audio data before an S-LID system is created. Therefore in summary, sufficient data, audio quality and label accuracy prove to be more important to an S-LID system than perfect transcriptions.

	25% Training Data				50% Training Data				100% Training Data			
	afr	eng	tsn	zul	afr	eng	tsn	zul	afr	eng	tsn	zul
	isiZulu Corpus A											
Precision	68.61%	69.97%	73.07%	65.31%	71.60%	72.27%	75.12%	68.37%	80.58%	79.38%	81.76%	82.30%
Recall	74.71%	66.89%	60.93%	72.29%	79.79%	63.73%	69.09%	73.27%	84.37%	77.75%	81.05%	80.45%
	Overall system accuracy : 68.92%				Overall system accuracy : 71.67%				Overall system accuracy : 80.97%			
	isiZulu Corpus B											
Precision	67.88%	71.46%	72.42%	65.71%	70.81%	74.91%	73.51%	68.22%	80.88%	79.31%	81.35%	81.15%
Recall	75.47%	68.13%	60.49%	71.16%	81.70%	61.53%	68.36%	73.69%	83.86%	78.43%	80.75%	79.32%
	Overall system accuracy : 69.05%				Overall system accuracy : 71.56%				Overall system accuracy : 80.67%			
	isiZulu Corpus C											
Precision	68.23%	69.61%	72.26%	64.47%	69.66%	71.72%	73.10%	67.67%	80.58%	79.80%	80.29%	82.32%
Recall	73.69%	67.03%	61.51%	70.46%	79.66%	60.98%	67.34%	72.43%	84.37%	78.15%	80.17%	79.88%
	Overall system accuracy : 68.37%				Overall system accuracy : 70.33%				Overall system accuracy : 80.73%			
	isiZulu Corpus D											
Precision	68.12%	68.56%	72.97%	67.48%	70.69%	73.12%	72.84%	66.49%	80.17%	80.58%	81.14%	81.88%
Recall	74.96%	67.72%	62.97%	69.76%	81.82%	61.67%	66.47%	71.16%	83.73%	79.25%	80.90%	79.46%
	Overall system accuracy : 69.09%				Overall system accuracy : 70.57%				Overall system accuracy : 80.91%			
	isiZulu Corpus E											
Precision	67.89%	70.12%	71.42%	66.17%	70.02%	71.54%	73.37%	67.81%	80.60%	80.58%	80.58%	82.05%
Recall	74.71%	66.07%	63.41%	69.90%	81.32%	60.43%	67.49%	71.72%	83.98%	79.25%	80.46%	79.74%
	Overall system accuracy : 68.71%				Overall system accuracy : 70.50%				Overall system accuracy : 80.94%			

Table 4.6: The performance of the SVM classifier in the back-end of the S-LID systems, according to the training data used.

CHAPTER FIVE

A SOUTH AFRICAN SPOKEN LANGUAGE IDENTIFICATION SYSTEM

5.1 INTRODUCTION

Now that the research questions mentioned in Section 1.2.1 have all been explored, the development of a South African Spoken Language Identification (S-LID) system can proceed. This chapter utilizes the knowledge gained from the previous experiments in order to develop an S-LID system specifically aimed at distinguishing between South African languages.

Again, the Parallel Phoneme Recognition followed by Language Modeling (PPR-LM) configuration [15] is implemented and the Meraka Lwazi corpus is used for the development of the proposed system.

This chapter is structured as follows: Section 5.2 provides a brief summary of the techniques which were investigated throughout this thesis and may be used in the development of the South African S-LID system. Section 5.3 explains the setup of the South African S-LID system in more detail. The initial South African S-LID system is introduced in Section 5.4 and refined in Section 5.5. The focus shifts from all eleven languages to a simplified version in which only the language families are focused upon in Section 5.6 before a final discussion of the South African S-LID system concludes the chapter in Section 5.7.

5.2 SUMMARY OF TECHNIQUES

Throughout this thesis, several techniques were proposed to answer the challenge of limited or incomplete resources. The effectiveness of these techniques were investigated, with varying results. Though not all of these techniques will be needed for the development of the final South African S-LID system, the most important techniques are listed here.

5.2.1 ORTHOGRAPHIC TOKENIZERS

Orthographic recognizers are created by defining each letter within the available transcriptions as a unique label. This is done with the assumption that the transcriptions are written phonetically (the Japanese transcriptions used the Roman alphabet). This technique will not work well for a language with a particularly irregular spelling system, that it, where many different graphemes can map to the same phoneme (compare the *f* from 'fish' with *gh* from 'enough') or visa verse (compare both *c*s in 'access').

It was found that the use of an orthographic-based recognizer built using very small training sets can be detrimental towards the overall performance of an S-LID system. When more training utterances are used (more than 2 500 samples) performance increases, but remains inferior to the use of the bootstrapped techniques (discussed next). It was decided that this method of creating a tokenizer with inadequate resources is not optimal, especially in the light of the technique described in Section 5.2.2.

5.2.2 BOOTSTRAPPED TRANSCRIPTIONS

Bootstrapped recognizers are developed with transcriptions created by another recognizer. The bootstrapping process involves recognizing the audio data with another phoneme recognizer. The existing recognizer can be one developed for a different language in the same environment, or one of the same language but developed with a better corpus. Note that if the recognizer was developed for another environment, the audio data should preferably be *channel normalized* (see Section 3.4.2) to the new environment, before the recognizer which will be used to bootstrap the new transcriptions can be used.

Such transcriptions, even when they are bootstrapped from audio data of very poor quality, can be beneficial to an S-LID system. Though the recognizers appear to decrease the performance of the S-LID system when trained using a very small training set, they augment the system positively very quickly, surpassing the original system's performance with a training set of only 1 000 utterances.

This technique has also proven invaluable in environments where transcriptions are not available.

5.2.3 ADAPTING AN EXISTING S-LID SYSTEM BEFORE USE IN A NEW ENVIRONMENT

One of the important experiments during this research was to investigate the concept of porting an existing S-LID system to a new environment. This is an important question regarding the re-usability of existing tools in the creation of an S-LID system. The technique of bootstrapping new transcriptions from existing ASR systems, as described in Section 5.2.2 has already proven useful. Therefore the re-usability of the ASR systems and the SVM was investigated.

It is found that an S-LID system can typically not be used outside of its original environment without adaptation. Creating phoneme strings with the adapted ASR systems before retraining the

SVM proved an important step in the porting process. For this technique to be successful, ASR system adaptation is required prior to phoneme string generation if channel conditions are mismatched (otherwise bootstrapping may fail completely).

The best results are obtained by retraining the entire S-LID system from the transcriptions bootstrapped from the original ASR systems. Therefore it is concluded that, if for whatever reason only the classifier can be retrained, an S-LID system will be usable but it remains preferable to retrain both tokenizers as well as the classifier.

5.2.4 DIARIZATION AND NORMALIZATION

Many of the systems presented to the NIST Language Recognition Evaluations (LRE) of 2007 utilized a data-preprocessing step before even attempting to extract any linguistic information. In Chapter 3, diarization, amplitude normalization and channel normalization were used as preprocessing steps in order to refine the audio data and to adapt the one corpus in order to resemble the other more closely.

Diarization is the process whereby silences and other non-speech noises are removed from the audio signal. Diarization is a very important technique that can be used to clean up a particularly noisy corpus. Amplitude normalization scales each utterance's sample values so that the absolute maximum value is one. It is done on a per-file basis and is used to reduce the effect of any extreme spikes in the raw wave-form of a *scratchy* corpus. Channel normalization reduces the difference in acoustic characteristics between two corpora. This is of great importance in a resource-scarse environment where data from different corpora have to be used in conjunction with one another.

5.2.5 QUANTITY OF DATA VS. QUALITY OF TRANSCRIPTIONS

The pattern recognition and classification algorithms implemented in the PPR-LM systems used for the experiments throughout this research have proven to be quite resilient against corpora with suboptimal (Chapter 4) or even completely missing (Chapter 2) transcriptions. Though the Hidden Markov Models (HMMs) implemented in the phoneme recognizers have shown the detrimental effect of suboptimal transcriptions, these small decreases in performance are not carried through to a decrease in performance of the system as a whole.

The quality of the audio data on the other hand, is shown in Chapter 3 to be more important, as well as the accuracy of the labels used to train the system. Therefore it should be stressed that though the quantity of data is clearly more important than the quality of the transcriptions, the quality of the audio data and the accuracy of the labels remains critical to the improvement of the S-LID system.

5.2.6 HUMAN VERIFICATION OF MISS-CLASSIFIED UTTERANCES

As mentioned in Section 5.2.5, the quality of the audio data used to train the S-LID system plays an important role in the eventual performance of the system. Therefore it may be of importance that at least a portion of the audio data that is to be used during the development of the system, for both training and testing purposes, should be verified beforehand.

This is done for the African corpus in Section 3.6.3. Initially, a subset of *suspicious* audio segments were generated by having the system perform language identification on the entire corpus (both training and testing data). Each verifier was then given a set of randomly selected audio segments that were chosen at random from this *suspicious* subset.

The verifiers identified the following four subsets: Usable quality, unusable quality, heavily accented and incorrectly labeled. When both the incorrectly labeled and unusable quality subsets were removed from the corpus, the performance of the system increased drastically. This suggests that verification of the audio data, especially the data used during training, is an important step before the development of an S-LID system can continue.

5.2.7 FILTERING LOW QUALITY AUDIO

A technique is developed to automatically filter out audio segments of low quality, based on the frequency of phonemes within the output provided by the phoneme recognizers for each utterance. Any segment that fails to achieve a phonemes-per-second count of higher than a predetermined threshold is automatically removed from the training set.

This had an immediate and positive effect, even with a minimum phonemes-per-second count of 0.5 seconds. In the specific experiment, the ASR systems' performance continued to increase for higher phonemes-per-second values and the S-LID system as a whole reached a peak performance at around 1.5 phonemes-per-second. This number may possibly be unique to the specific environment in which the tests were conducted, but the process itself is data-neutral.

The systematic evaluation of the phoneme strings produced by the recognizers is definitely a valuable technique, especially in an environment where the quality of the audio data is of a very poor or unknown quality. Utterances with weak phoneme recognition can then be discarded to improve the system's performance.

5.3 EXPERIMENTAL SETUP

This section describes the design of the South African S-LID system that is presented in this chapter. The setup used for the South African S-LID system developed in this chapter is very similar to that used in the rest of the thesis. The acoustic models of the ASR systems at the front-end are implemented a bit differently, as described in Section 5.3.2 and a more complete overview of the corpora is provided in Section 5.3.1.

5.3.1 CORPUS STATISTICS

The South African S-LID system is trained and tested with the Meraka Lwazi corpus, which was introduced in Section 4.2.1. For the development of the South African S-LID system, all eleven languages in the corpus are utilized. Table 5.1 provides some statistics on the available data. The number of speakers per language is given, as well as the combined number of utterances and the length of the audio data in hours. The differences between the train and test set is also provided.

Language	Set	Speakers	Utterances	Hours
Afrikaans	Train	170	4382	3.32
	Test	30	787	0.60
SA English	Train	175	4287	3.25
	Test	30	728	0.55
isiNdebele	Train	170	4878	3.69
	Test	30	846	0.64
isiZulu	Train	171	4852	3.23
	Test	29	685	0.52
isiXhosa	Train	180	4458	3.38
	Test	30	669	0.51
Sepedi	Train	169	3674	2.78
	Test	30	664	0.50
Sesotho	Train	170	4642	3.52
	Test	30	815	0.62
Setswana	Train	176	4257	3.10
	Test	30	686	0.54
siSwati	Train	178	4778	3.62
	Test	30	811	0.62
Tshivenda	Train	171	4414	3.34
	Test	30	770	0.58
Xitsonga	Train	168	4230	3.21
	Test	30	755	0.57

Table 5.1: Statistics on the training and testing sets for each of the South African official languages, as used for the South African Spoken Language Identification system.

5.3.2 SYSTEM DESIGN

The South African S-LID system implements the popular Parallel Phoneme Recognition followed by Language Modeling (PPR-LM) architecture, with the phoneme recognizers of Automatic Speech Recognition (ASR) systems being used as tokenizers.

The phoneme recognizers utilize context-dependent Hidden Markov Models (HMMs). These HMMs consist of three emitting states with seven mixtures per Gaussian Mixture Model (GMM) within each state. The HMMs are trained using the HTK program [33] on Mel Frequencies Cepstral Coefficients (MFCCs) which encode thirty nine features (13 MFCC, 13 delta costs and 13 acceleration costs). Cepstral Mean Normalization (CMN) as well as Cepstral Variance Normalization (CVN) are used as a feature-domain channel normalization technique. Semi-tied transforms are applied to the HMMs and a flat phone-based language model is employed for phone recognition [27].

A Support Vector Machine (SVM) at the back-end classifies the languages, based on biphone frequencies which are extracted from the output of the phoneme recognizers in the form of a vector. For a more complete description of the SVM, please refer to Section 2.2.2.

5.4 THE INITIAL SOUTH AFRICAN S-LID SYSTEM

In this section, the first attempt to develop a successful South African S-LID system is described. The system is described in Section 5.4.1 and the results are provided in Section 5.4.2

5.4.1 SYSTEM DESCRIPTION

For the development of the South African S-LID system, three system designs are compared with one another. Each design implements the PPR-LM architecture, but utilizes an increasing number of phoneme recognizers, starting with only one language, then four and finally all eleven. The languages chosen for the different designs are:

- SA English;
- Afrikaans, SA English, Setswana and isiZulu;
- All eleven languages.

Although Chapter 2 already confirms that an S-LID system's performance will increase as more tokenizers are added, the addition of each new tokenizer also increases the computational cost to create the system. It is therefore necessary to determine a feasible number of phoneme recognizers for the system design.

5.4.2 RESULTS

As expected, the overall performance of the system increases with the addition of more phoneme recognizers. The system with only one phoneme recognizer struggles as it achieves an overall accuracy of just below 35%. Utilizing four recognizers, the system's performance increases to just above 50% and with all eleven tokenizers to just below 60%. Therefore, though requiring a significant amount of computational effort to create, the system with all eleven recognizers is the preferred configuration for the South African S-LID system.

It is also not surprising that the best performing South African S-LID system performs poorer than the baseline system of Chapter 4 (please refer to Section 4.2.3) as the addition of new target languages is expected to have a negative impact on the system's overall performance. Table 5.6 at the end of the chapter displays the performance of all three configurations of the South African S-LID system in more detail. The accuracy and correctness of the phoneme recognizers as well as the precision and recall for each language are given.

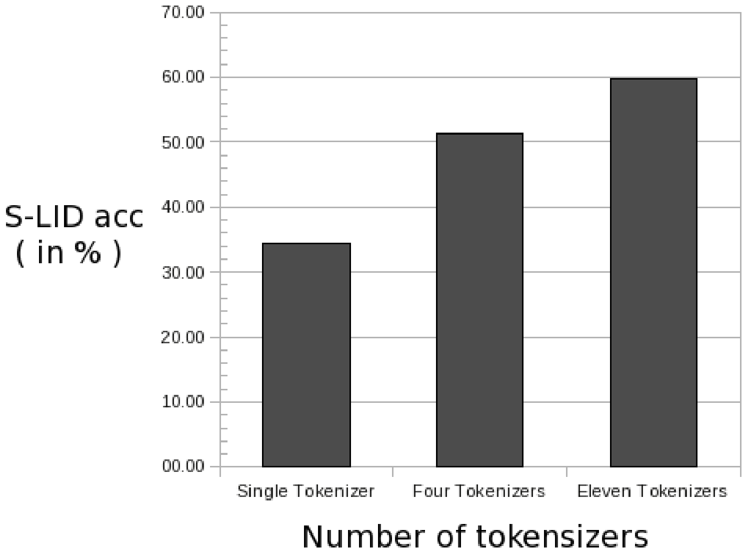

Figure 5.1: Overall SVM performance of all three configurations for the South African S-LID systems.

Figure 5.1 represents the overall performance of the SVM at the back-end in percentage accuracy for all three configurations for the South African S-LID systems visually. Table 5.2 displays the confusion matrix for the best performing South African S-LID system. The columns represent the correct language of the utterances whereas the rows represent the language as predicted by the system. The number of correctly classified utterances on the main diagonal of the matrix are boldfaced for clarity.

	afr	eng	nbl	nso	sot	tsn	ssw	ven	xho	tso	zul
afr	**646**	92	38	50	44	42	29	31	51	58	31
eng	70	**552**	29	41	47	35	35	28	43	31	54
nbl	5	3	**521**	13	19	13	53	34	68	36	73
nso	4	3	8	**295**	58	39	7	18	5	22	12
sot	12	18	19	87	**427**	77	24	18	21	34	16
tsn	6	5	16	75	82	**389**	8	21	11	19	17
ssw	7	6	55	20	38	10	**513**	21	40	38	68
ven	18	4	24	33	23	25	26	**520**	16	63	11
xho	2	8	52	8	15	9	42	13	**312**	17	52
tso	6	11	26	16	21	16	18	35	22	**399**	19
zul	11	26	58	26	41	31	56	31	80	38	**332**

Table 5.2: The confusion matrix for the initial South African S-LID system.

As expected, the two Germanic languages are mostly classified correctly, as they are not part of any of the family of Southern Bantu languages that make up the rest of the target languages. Instead these two appear to affect each other more severely, though the large number of Southern Bantu utterances that are classified as either Afrikaans or English is quite surprising. This effect is analyzed in the next section.

5.5 IMPROVING THE SOUTH AFRICAN S-LID SYSTEM

In this section, an attempt is made to improve the accuracy of the initial South African S-LID system of Section 5.4. The corpus is examined more closely in Section 5.5.1 after which an additional experiment is developed in Section 5.5.2. The results of the improved South African S-LID system are provided in Section 5.5.3.

5.5.1 CORPUS EXAMINATION

In order to determine the cause of the unusually high difference between the precision and recall percentages of the two Germanic languages, a more detailed investigation is made concerning the audio data of the Meraka Lwazi corpus. As the images in Appendix 1 reveal, it is found that most of the audio segments for both Germanic languages are shorter than 6 seconds. Therefore it is theorized that the SVM is biased towards the Germanic languages, especially Afrikaans, when it comes the such short segments.

The accuracy of the S-LID system is also analyzed according to utterance length (grouped together into time frames that increase with intervals of 3 seconds each) and language identity. Figures 5.2 and 5.3 both show these results, with the former displaying the accuracies calculated on language-specific performance and the latter displaying the average accuracies across all languages. As can be seen in Figure 5.2, the few values that relate to Afrikaans achieve accuracies of above 80%. From Figure 5.2 it can be seen that the accuracies of each time frame (increasing with an interval of three seconds each) are on average above that of the shorter utterances, except for the longest utterances which appear to be struggling.

5.5.2 DATA FILTERING

In order to improve the South African S-LID system, the data used to train the SVM on is balanced more equally across the different lengths of utterances for all eleven languages.

This is accomplished by removing all audio files that are shorter than 2 seconds and longer than 10 seconds from both the classifiers training and test sets. These boundaries are chosen to ensure that a single language does not dominate the available amount of training samples within a specific utterance length. (These files are removed from the test set as well, as we are not interested in S-LID performance on ultra-short segments.) After examining the average phoneme recognition rates, it is also decided to implement the technique of filtering all utterances with a phoneme recognition of less than four phonemes-per-second, as is done in Section 3.6.2.

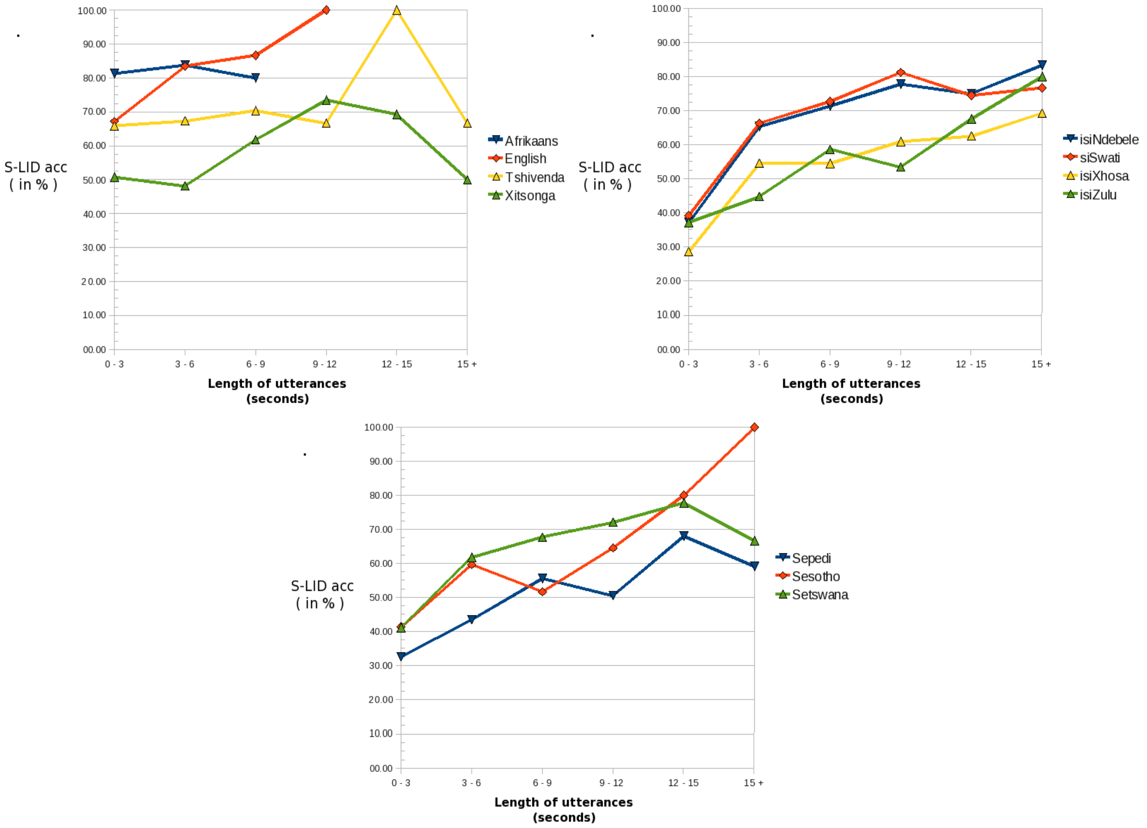

Figure 5.2: The language specific performance of the initial South African S-LID system, considered on different lengths of testing utterances.

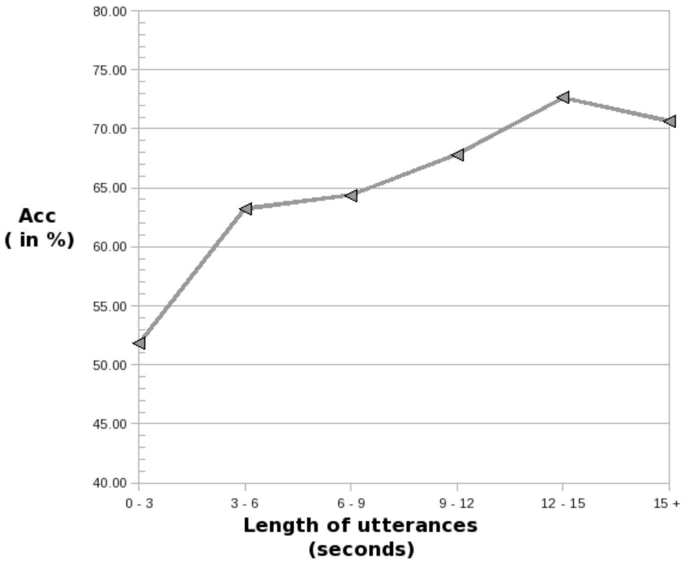

Figure 5.3: The average performance of the initial South African S-LID system, considered on different lengths of testing utterances and calculated across all languages.

The remaining audio files are restricted so that each language has 2 112 utterances in the training set, and 361 utterances in the test set. This is merely done to ensure that one language does not have more training data than another, as these values represents the smallest number of utterances any language has for the two sets. These new training and test sets are then recognized by all eleven phoneme recognizers before the resulting phoneme strings are used to retrain and test the SVM classifier at the back-end.

5.5.3 RESULTS

The techniques described in Section 5.5.2 result in a clear improvement, as can be seen in Table 5.7 at the end of the chapter. The accuracy of the system improves from an overall accuracy of 59.71% to 66.73%. It should also be noted that the precision of the Germanic languages has improved significantly, suggesting that the balancing of the training set has indeed created a better SVM.

Table 5.3 displays the confusion matrix for the improved South African S-LID system. The columns represent the correct language of the utterances whereas the rows represent the language as predicted by the system. The number of correctly classified utterances on the main diagonal of the matrix is boldfaced for clarity.

	afr	eng	nbl	nso	sot	tsn	ssw	ven	xho	tso	zul
afr	**318**	34	3	11	14	13	10	6	14	8	8
eng	16	**302**	1	9	5	0	2	4	11	11	12
nbl	0	0	**258**	10	4	6	24	13	26	14	31
nso	4	2	5	**226**	46	32	7	15	6	20	5
sot	5	6	2	28	**197**	28	9	9	17	19	6
tsn	3	2	4	38	42	**244**	8	7	3	9	6
ssw	1	1	23	9	11	6	**247**	10	26	14	43
ven	3	3	10	10	9	8	4	**254**	8	29	16
xho	2	5	28	7	7	3	18	8	**207**	9	38
tso	3	2	7	7	16	12	10	21	14	**212**	11
zul	6	4	20	6	10	9	22	14	29	16	**185**

Table 5.3: The confusion matrix for the improved South African S-LID system.

Figure 5.4 displays this improvement visually. The average accuracy percentages calculated across all target languages and considered on the different lengths of testing utterances are provided.

5.6 IDENTIFYING LANGUAGE FAMILIES

When it is considered that the South African S-LID system has to distinguish between eleven, closely related languages with test samples of between 2 and 10 seconds long, the system performs reasonably well. This can be illustrated by combining the results according to language families, instead of representing the languages individually. The languages can be combined as follows:

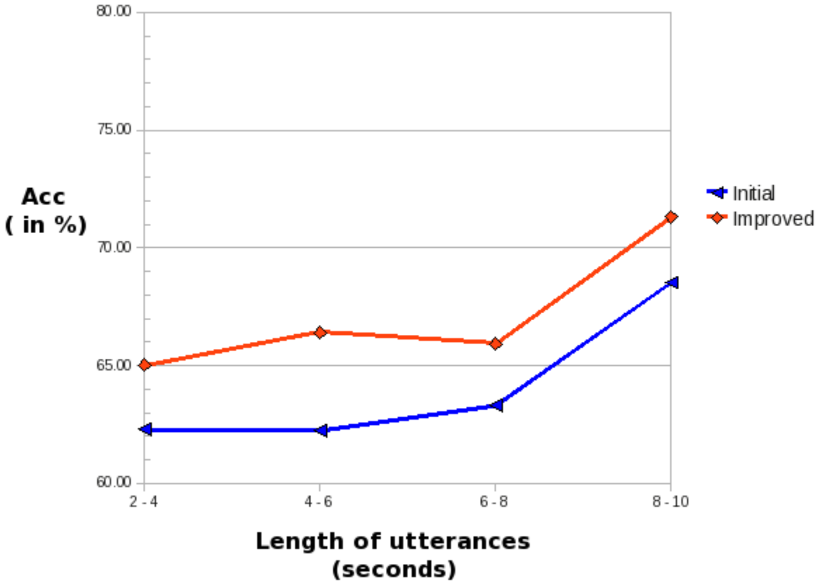

Figure 5.4: The average performance of the initial South African S-LID system compared to the improved SVM. Accuracy is considered on different lengths of testing utterances and calculated across all languages.

Afrikaans Though part of the Germanic languages, Afrikaans can be classified further as "Low Franconian".

English Though part of the Germanic languages, English can be classified further as "AngloFrisian".

Sotho-Tswana The languages classified into this group are Sepedi, Setswana and Sesotho.

Nguni The languages classified into this group are siSwati, isiNdebele, isiXhoza and isiZulu.

Tswa-Ronga Xitsonga is the only language within this family group.

Venda Tshivenda is the only language within this family group.

When the ambiguity of closely related target languages is removed, the system's performance increases significantly to an overall accuracy of 80.38%. Note that the results are still acquired from the improved South African S-LID system of Section 5.5.2, and have not been generated by a newly trained system. Table 5.4 displays the precision and recall for each language family.

	Afrikaans	English	Sotho-Tswana	Nguni	Tswa-Ronga	Venda
Precision	72.43%	80.96%	83.11%	85.66%	67.30%	71.75%
Recall	88.08%	83.65%	81.34%	84.83%	58.72%	70.36%
	Overall system accuracy : 80.38%					

Table 5.4: The performance of the SVM classifier in the back-end of the South African system, when only language families are considered.

Table 5.5 displays the confusion matrix for the best performing South African S-LID system. The columns represent the correct language of the utterances whereas the rows represent the language as predicted by the system. The number of correctly classified utterances on the main diagonal of the matrix are boldfaced for clarity.

	Afrikaans	English	Sotho-Tswana	Nguni	Tswa-Ronga	Venda
Afrikaans	**318**	34	38	35	8	6
English	16	**302**	14	26	11	4
Sotho-Tswana	12	10	**881**	78	48	31
Nguni	9	10	88	**1225**	53	45
Tswa-Ronga	3	2	35	42	**212**	21
Venda	3	3	27	38	29	**254**

Table 5.5: The confusion matrix for the South African S-LID system with eleven phoneme recognizers when only language families are considered.

Figure 5.5 displays the average accuracies the improved South African S-LID system achieves on the reduced test set (same set used to achieve the results as given in in Table 5.5).

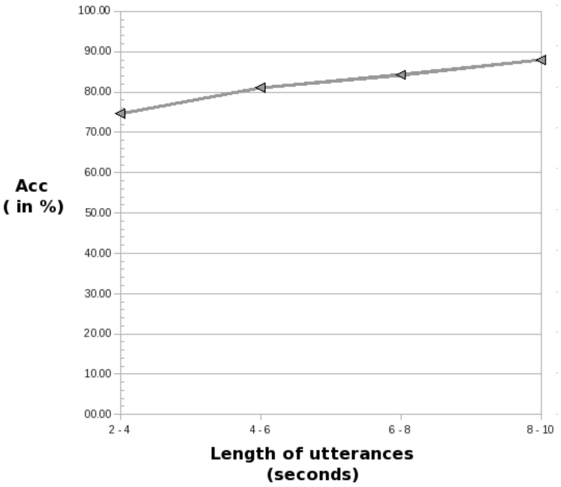

Figure 5.5: The average performance of the improved South African S-LID system, considered on different lengths of testing utterances and calculated across the various language families.

5.7 CONCLUSION

This chapter reviewed a set of newly developed techniques for S-LID system development with limited resources and demonstrated that it is possible to successfully develop a South African S-LID system, capable to distinguish between all of the country's eleven languages on relatively short test utterances. As expected, the languages within the different language families prove to influence each other negatively.

However, a high degree of accuracy is achieved when only language families are considered, resulting in around 81% classification accuracy for test samples between 4 and 6 seconds long when compared to 67% for the same utterance lengths when the proper languages are considered. As expected, the longer lengths of utterances achieves better results, with classification accuracies for the 8 to 10 second long test samples of 71% and 88% for proper languages and language families respectively.

	afr	eng	nbl	nso	sot	tsn	ssw	ven	tso	xho	zul
	ASR Systems										
Correct	70.49%	58.72%	73.02%	68.00%	67.76%	70.64%	74.04%	75.76%	68.53%	68.54%	69.81%
Accuracy	65.47%	52.30%	66.61%	57.78%	57.17%	57.08%	65.77%	67.37%	60.92%	58.57%	63.06%
	Single Tokenizer										
Precision	42.47%	75.48%	35.64%	28.64%	26.89%	27.42%	37.81%	35.66%	36.92%	27.56%	25.37%
Recall	71.02%	43.13%	35.22%	25.75%	26.99%	28.42%	33.29%	40.38%	30.86%	26.15%	24.81%
	Overall system accuracy : 35.49%										
	Four Tokenizers										
Precision	72.24%	63.67%	48.57%	41.70%	40.99%	58.03%	49.71%	51.15%	49.20%	42.63%	48.29%
Recall	74.08%	61.40%	54.25%	42.77%	47.73%	44.75%	52.65%	57.40%	44.90%	44.09%	35.18%
	Overall system accuracy : 51.28%										
	All Tokenizers										
Precision	58.09%	57.20%	62.17%	62.63%	56.71%	59.94%	62.86%	68.15%	67.74%	58.86%	45.48%
Recall	82.08%	75.82%	61.58%	44.43%	52.39%	56.71%	63.25%	67.53%	52.85%	46.63%	48.47%
	Overall system accuracy : 59.71%										

Table 5.6: The performance of the ASR systems in the front-end as well as the SVM classifier in the back-end of the initial South African S-LID system.

	afr	eng	nbl	nso	sot	tsn	ssw	ven	tso	xho	zul
	Single Tokenizer										
Precision	43.38%	57.62%	46.78%	38.64%	32.74%	41.14%	50.38%	50.00%	49.15%	42.35%	70.00%
Recall	65.37%	65.92%	56.50%	37.67%	30.74%	41.82%	53.73%	48.47%	40.16%	39.88%	29.08%
	Overall system accuracy : 46.31%										
	Four Tokenizers										
Precision	72.63%	80.12%	65.92%	58.40%	41.05%	55.14%	51.62%	54.25%	50.00%	47.38%	42.64%
Recall	83.10%	70.36%	57.34%	40.44%	47.64%	57.89%	57.34%	61.77%	50.41%	47.64%	40.16%
	Overall system accuracy : 55.82%										
	All Tokenizers										
Precision	72.44%	80.96%	66.83%	61.41%	60.43%	66.66%	63.17%	71.75%	67.30%	62.35%	57.63%
Recall	88.09%	83.65%	71.46%	62.60%	54.57%	67.59%	68.42%	70.36%	58.73%	57.34%	51.25%
	Overall system accuracy : 66.73%										

Table 5.7: The performance of the SVM classifier in the back-end of the improved South African system.

CHAPTER SIX

CONCLUSION

6.1 INTRODUCTION

The objective of this research was to create a set of techniques for the development of a Spoken Language Identification (S-LID) system when limited linguistic resources are available, as is the case with many South African languages. In order to achieve this, the effect of limited, poor quality or incomplete data was investigated. Finally, the information gained from this thesis was put to use and an S-LID system was developed which is able to distinguish between all eleven of South Africa's official languages.

6.2 SUMMARY OF CONTRIBUTION

Three key questions related to the resources required for the development of an S-LID system were identified. These three questions are of specific importance to phoneme recognizers in the *Parallel Phoneme Recognition followed by Language Modeling (PPR-LM)* architecture, and were examined during the course of this research. The findings are listed below.

6.2.1 INCOMPLETE RESOURCES

Techniques to create phoneme recognizers with incomplete resources were proposed and the possibility to implement such techniques investigated. A new language was added to an existing S-LID system, at first with no native recognizer included in the front-end. The result from this system was then compared to results from systems with phoneme recognizers created with incomplete resources. It was found that adding more tokenizers to an existing S-LID system increases the performance in accuracy of the S-LID system. The technique of bootstrapping new transcriptions was also described and shown to be effective when transcriptions cannot be acquired. (Some of these results were published in [38].)

6.2.2 MISMATCHED RESOURCES

The use of corpora with mismatched audio to create an S-LID system was also investigated. Attention was given to the porting of an existing system from a resource-rich environment to a poorly resourced environment. It was found that an S-LID system may be unable to function properly outside its original environment without the adaptation of at least the classifier to the conditions of the new environment. Techniques such as diarization (which removes any non-speech noise, digital signals and silences from the audio data) and channel normalization (which minimizes the differences of the acoustic characteristics between different corpora) can be applied to the audio data as a preprocessing step. The best result was still achieved by retraining both the tokenizers as well as the classifier. It was also found that verifying the quality of the audio data from the new environment is important and that filtering out utterances that record a poor frequency of phonemes can be beneficial to the performance of the system. (Some of these results were published in [39].)

6.2.3 SUB-OPTIMAL RESOURCES

The effect of poor quality transcriptions on an S-LID system was investigated as well. Different corpora containing increasingly accurate transcriptions were defined and added to an existing system. It was found that, though the ASR systems appear to be sensitive to the quality of the corpora, the classifier was robust enough to function equally well with any of the corpora evaluated. It should be kept in mind that large changes in the audio data between corpora will still influence the S-LID system as a whole, as is described in Section 6.2.2.

6.2.4 DEVELOPMENT OF A SOUTH AFRICAN S-LID SYSTEM

The knowledge gained from the above-mentioned research was reviewed and utilized in order to develop an S-LID system that distinguishes between South African languages. The main techniques identified and investigated during the course of this research were:

- Orthographic tokenizers
- Bootstrapped transcriptions
- Adapting an existing S-LID system before use in a new environment
- Diarization and channel normalization
- Utilizing poor-quality transcriptions
- Human verification of miss-classified utterances

- Filtering low quality audio

These are all described in more detail in Section 5.2.

The South African S-LID system was implemented with the PPR-LM design and configured to use phoneme recognizers for all eleven official languages. Some loss in overall accuracy was expected as several, closely related languages were added to the system. However, a high degree of accuracy is achieved when only language families are considered, resulting in above 80% classification accuracy for test samples between 2 and 10 seconds long.

6.3 FUTURE WORK

This thesis successfully created an S-LID system, capable of distinguishing between all eleven of South Africa's official languages. This system, as described in Chapter 5, performs well, achieving an overall accuracy of 80.38% when classifying between language families on short segments and achieving 66.73% when all eleven languages are recognized individually. On audio segments of between 2 and 10 seconds long, it is not yet effective enough to be used in a real world environment, but on longer segments, the system becomes practically usable. Future work includes evaluating performance on longer segments (which would include additional data collection) and investigating other methods in order to increase on the effectiveness of the system.

Bootstrapping of new transcriptions, as well as filtering of low quality audio are techniques that could have been implemented as an iterative process to refine results. However, for this thesis only one iteration was implemented for each experiment. It may be interesting to see what the effect of multiple iterations on the results will be and how many iterations will be required to result in a clear improvement.

Though only the PPR-LM architecture have been used throughout this thesis, Section 1.1.3 states that combining different S-LID techniques improves overall results. Therefore it may also be worthwhile expand on the South African S-LID system developed in Chapter 5 to include other S-LID techniques that utilizes information sources of low linguistic knowledge. Of particular interest would be to incorporate the acoustic scores a particular utterance achieves with each phoneme recognizer into the language classification process, whether it is as part of the vector space, or in conjunction with it. It would also be interesting to determine whether all eleven tokenizers are indeed required for the optimal system, and whether family-specific tokenizers could possibly result in similar performance.

6.4 CONCLUSION

This thesis has investigated the effects of limited resources on the development of an S-LID system and introduced several techniques to decrease the negative effects of suboptimal data. These techniques were proven to be feasible and were successfully implemented to create a South African S-LID system. It is believed that some of the lessons learned from this thesis, as well as the techniques developed during this research, can also be used for Automatic Speech Recognition systems in environments with limited resources.

APPENDIX A

MERAKA LWAZI CORPUS: DISTRIBUTION OF UTTERANCE LENGTHS.

Here follows a set of graphs which illustrates the distribution of utterances within the Meraka Lwazi corpus.

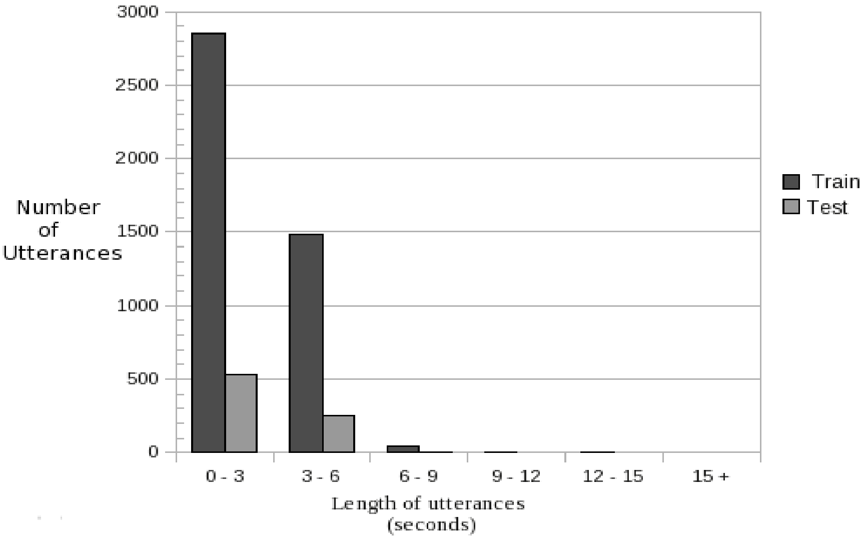

Figure A.1: Distribution of utterances of the specified lengths for Afrikaans. Both the training and the test sets are displayed.

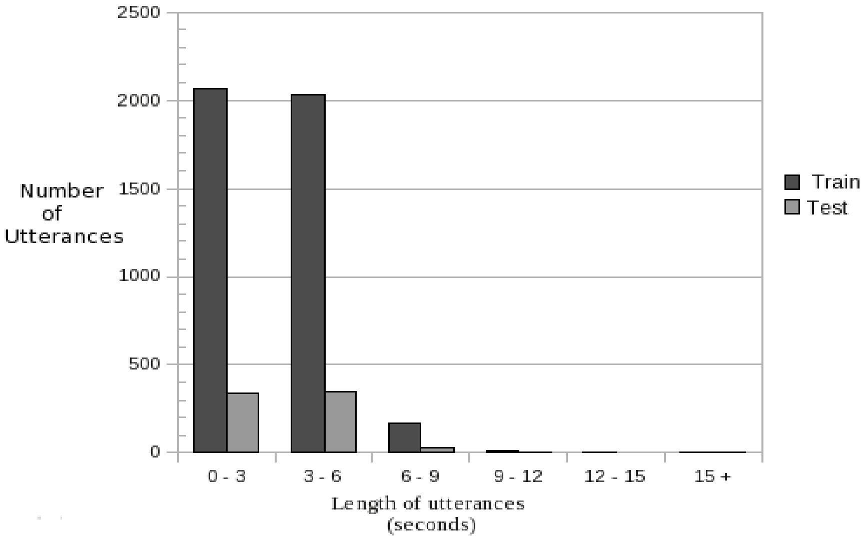

Figure A.2: Distribution of utterances of the specified lengths for English. Both the training and the test sets are displayed.

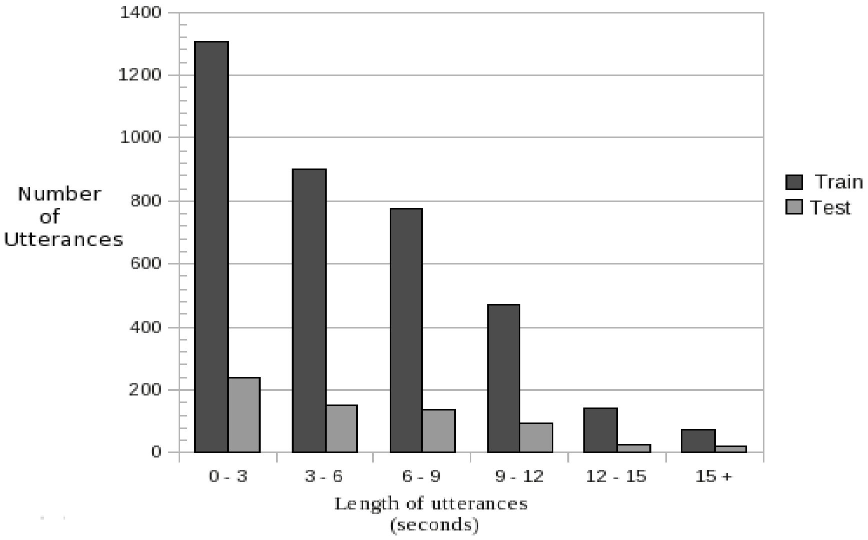

Figure A.3: Distribution of utterances of the specified lengths for Sepedi. Both the training and the test sets are displayed.

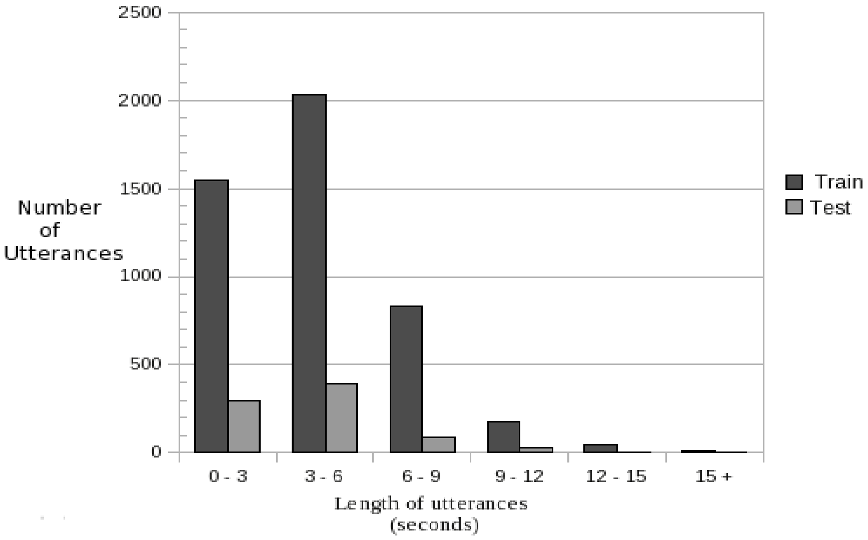

Figure A.4: Distribution of utterances of the specified lengths for Sesotho. Both the training and the test sets are displayed.

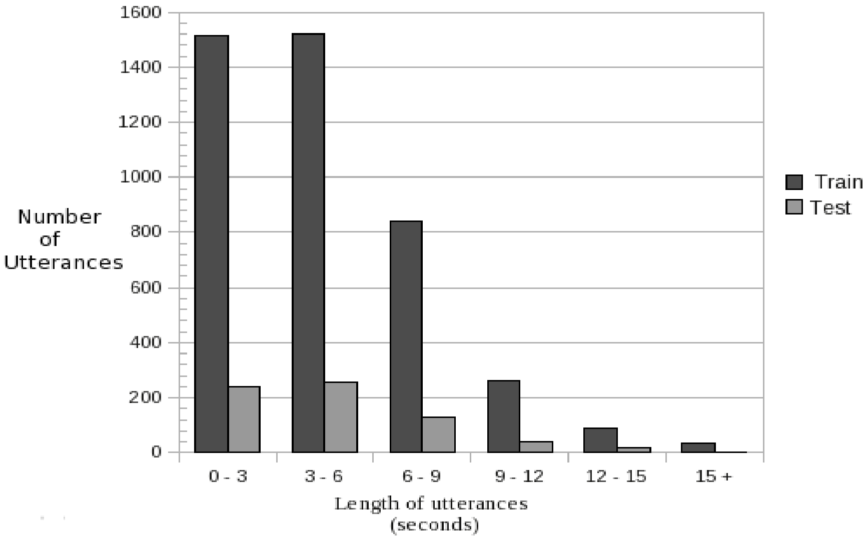

Figure A.5: Distribution of utterances of the specified lengths for Setswana. Both the training and the test sets are displayed.

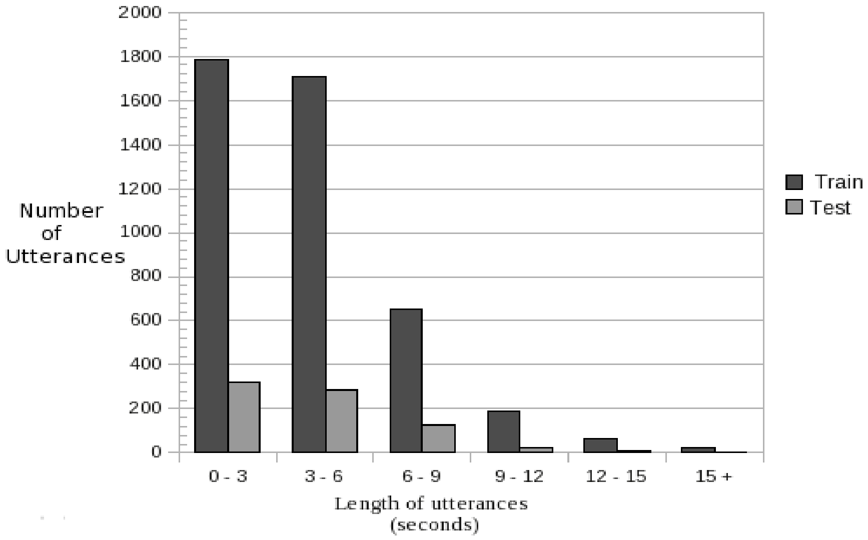

Figure A.6: Distribution of utterances of the specified lengths for Tshivenda. Both the training and the test sets are displayed.

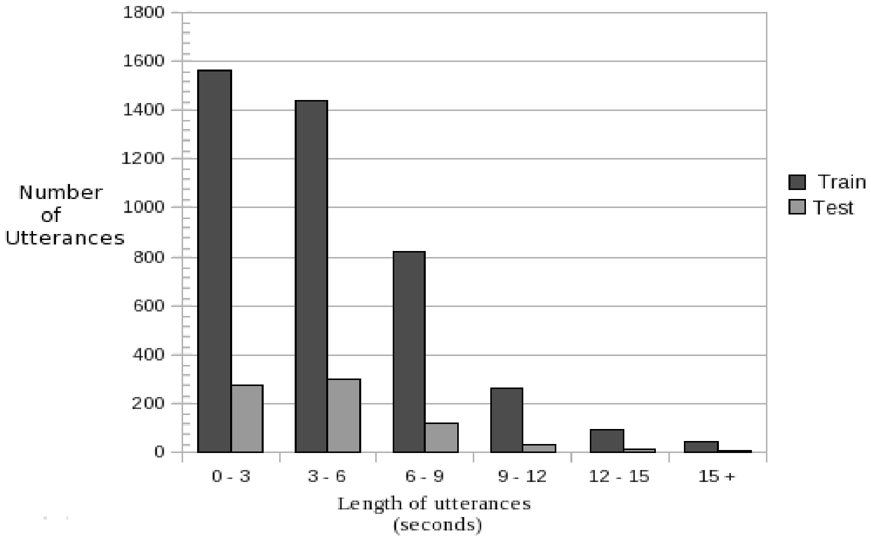

Figure A.7: Distribution of utterances of the specified lengths for Xitsonga. Both the training and the test sets are displayed.

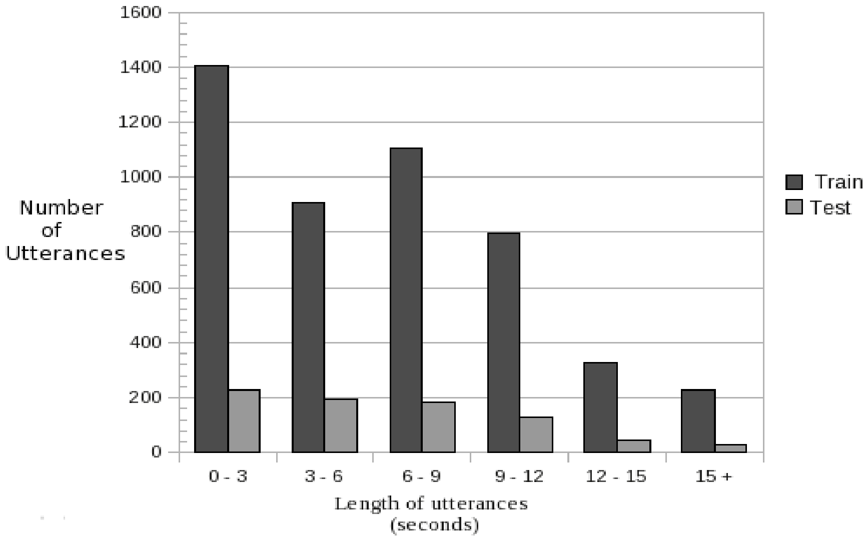

Figure A.8: Distribution of utterances of the specified lengths for siStwati. Both the training and the test sets are displayed.

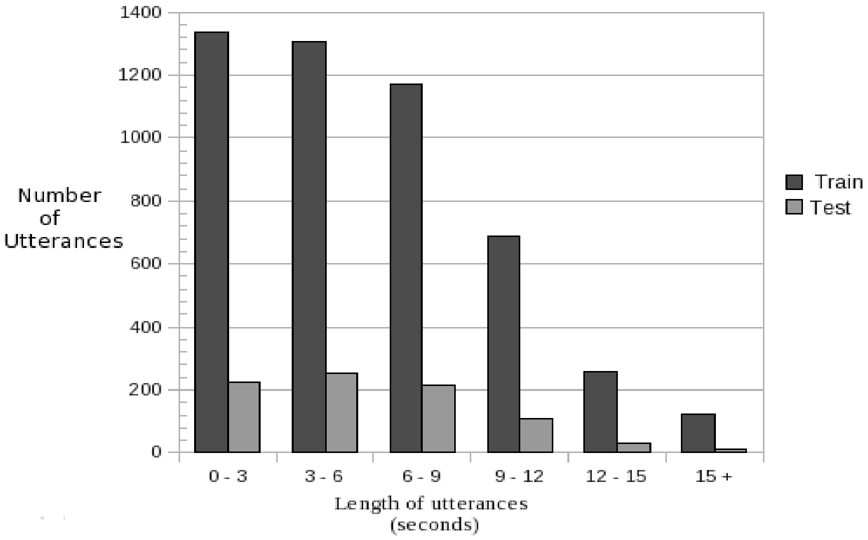

Figure A.9: Distribution of utterances of the specified lengths for isiNdebele. Both the training and the test sets are displayed.

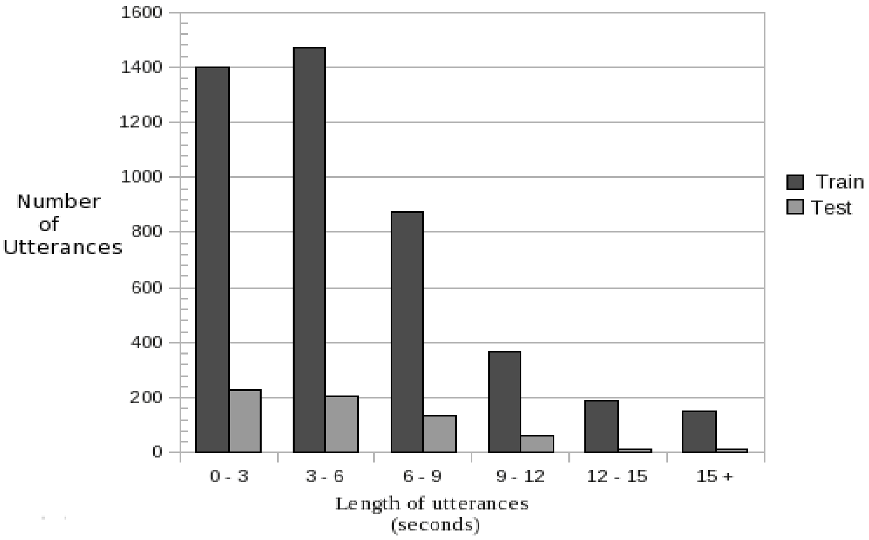

Figure A.10: Distribution of utterances of the specified lengths for isiXhosa. Both the training and the test sets are displayed.

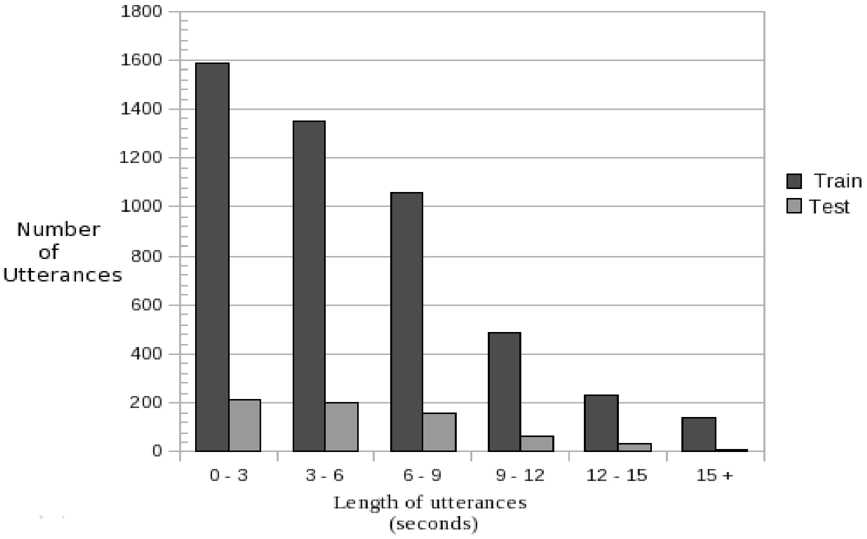

Figure A.11: Distribution of utterances of the specified lengths for isiZulu. Both the training and the test sets are displayed.

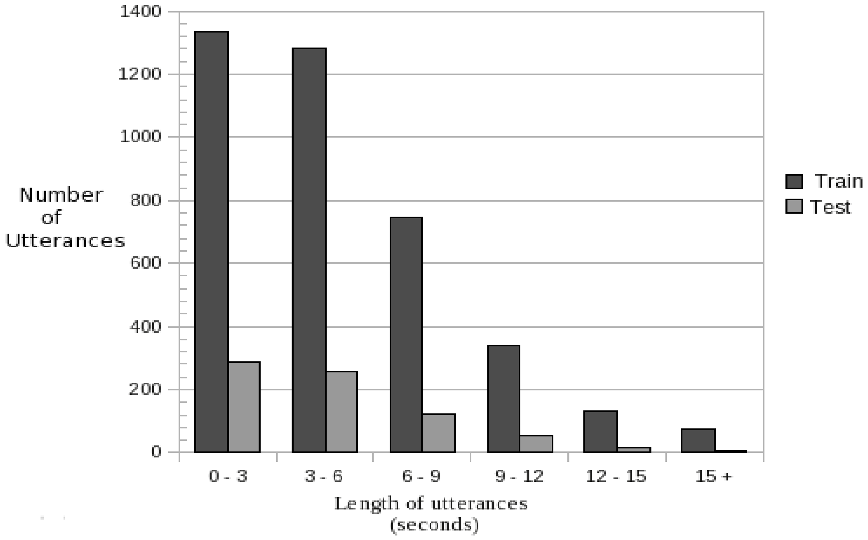

Figure A.12: Distribution of utterances of the specified lengths for all eleven languages combined. Both the training and the test sets are displayed.

APPENDIX B

CONTACT INFORMATION

Postal Address : 514 Banket Drive, Helderkruin, Roodepoort, 1724
E-mail : mpeche@csir.co.za
Tel Number : 012 841 4633
Fax Number : 012 841 4829
Cell Number : 072 172 4815

REFERENCES

[1] W. Cavnar and J. M. Trenkle, “N-gram based text categorization,” in *Proceedings of the Annual Symposium on Document Analysis and Information Retrieval*, 1994, vol. 3, pp. 161 – 169.

[2] G. Botha, V. Zimu, and E. Barnard, “Text-based language identication for the South African languages,” in *Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, 2006, vol. 17, pp. 46 – 52.

[3] Marc A. Zissman and Kay M. Berkling, “Automatic Language Identification,” *Speech Communication*, vol. 35, pp. 115 – 124, 2001.

[4] M.A. Zissman, “Comparison of four approaches to automatic language identification of telephone speech,” *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31 – 44, 1996.

[5] R.G Leonard and G.R. Doddington, “Automatic language identification. Technical report RADC-TR-74-200,” *Air Force Rome Air Development Center*, 1974.

[6] J.T Foil, “Language identification using noisy speech,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processung (ICASSP)*, 1986, pp. 861 – 864.

[7] S.C Kwasny, B.L. Kalman, W. Wu, and A.M. Engebretson, “Identifying language from speech: An example of high-level statistically-based feature extraction,” in *Proceedings of the Annual Conference of the Cognitive Science Society*, 1992, vol. 14, pp. 53 – 57.

[8] Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li, and Eng Siong Chng, “Integrating Acoustic, Prosodic and Phonotactic Features for Spoken Language Identification,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processung (ICASSP)*, 2006, pp. 205–208.

[9] D. Matrouf, M. Adda-Decker, L. Lamel, and J. Gauvain, “Language Identification Incorporating Lexical Information,” *International Conference on Spoken Language Processing (ICSLP)*, pp. 181 – 184, 1998.

[10] Bo Yin, Eliathamby Ambikairajah, and Fang Chen, “Combining cepstral and prosodic features in LID,” *International Conference on Pattern Recognition*, vol. 18, pp. 254 – 257, 2006.

[11] A.E. Thyme-Gobbel and S.E. Hutchins, “On using prosodic cues in automatic language identification,” *International Conference on Spoken Language Processing (ICSLP)*, vol. 3, pp. 1768 – 1772, 1996.

[12] K. M. Berkling and E. Barnard, “Theoretical error prediction for a language identification system using optimal phoneme clustering,” in *Proceedings of the European Conference on Speech Communication and Technology(EuroSpeech)*, 1995, vol. 4, pp. 351 – 354.

[13] R.B. Ives, “A minimal rule AI expert system for real-time classification of natural spoken languages,” in *Proceedings of the Annual Artificial Intelligence and Advanced Computer Technology Conferance*, 1986, vol. 2.

[14] A.S. House and E.P. Neuberg, “Toward automatic identification of the language of an utterance,” *Preliminary methodological consiferations Journal of the Acoustical Society of America*, pp. 708 – 713, 1977.

[15] Yeshwant K Muthusamy, Ethienne Barnard, and Ronald A Cole, “Reviewing Automatic Language Identification,” *IEEE Signal Processing Magazine*, 1994.

[16] Christian Mul̈ler and Joan-Isaac Biel, “The ICSI 2007 language recognition system,” *Poceedings of the IEEE International Conference on Speaker and Language Recognition, (IEEE Odyssey)*, 2008, paper 013.

[17] Pavel Matejka, Lukas Burget, Ondrej Glembek, Petr Schwarz, Valiantsina Hubeika, Michal Fapso, Tomas Mikolov, and Oldrich Plchot, “BUT system description for NIST LRE 2007,” 2007, http://www.fit.vutbr.cz/research/groups/speech/lid/2007/BUT.pdf.

[18] Haizhou Li and Bin Ma, “A phonotactic language model for spoken language identification,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005, vol. 43, pp. 515 – 522.

[19] Bin Ma and Haizhou Li, “A Comparative Study of Four Language Identification Systems,” *Computational Linguistics and Chinese Language Processing*, vol. 11, no. 2, pp. 980 – 985, 2006.

[20] A. Montero-Asenjo, D.T. Toledano, J. Gonzalez-Dominguez, J. Gonzalez-Rodriguez, and J. Ortega-Garcia, “Exploring PPR-LM Performance for NIST 2005 Language Recognition Evaluation,” in *Poceedings of the IEEE International Conference on Speaker and Language Recognition, (IEEE Odyssey)*, 2006, pp. 1 – 6.

[21] B. Ma., H. Li, and C.H. Lee, “An Acoustic Segment Modeling Approach to Automatic Language Identification,” *Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 2829 – 2832, 2005.

[22] C. Corredor-Ardoy, J.L. Gauvain, M. Adda-Decker, and L.Lamel, "Language identification with language-independent acoustic models," in *Proceedings of the European Conference on Speech Communication and Technology(EuroSpeech)*, 1997, pp. 355 – 358.

[23] Li, H., and B. Ma, "A Phonotactic Language Model for Spoken Language Identification," *Meeting of the Association for Computational Linguistics*, vol. 43, pp. 515 – 522, 2005.

[24] Gerrit R. Botha and Etienne Barnard, "Factors that affect the accuracy of text-based language identication," in *Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, 2007, vol. 18, pp. 7 – 10.

[25] "NIST website: The 2007 NIST language recognition evaluation results," 2008, http://www.nist.gov/speech/tests/lre/2007/lre07_eval_results_vFINAL/index.html.

[26] Alvin F Martin and Audrey N Le, "NIST 2007 language recognition evaluation," *Poceedings of the IEEE International Conference on Speaker and Language Recognition, (IEEE Odyssey)*, 2008, paper 016.

[27] Etienne Barnard, Marelie Davel, and Charl van Heerden, "ASR corpus design for resource-scarce languages," *Submitted to Annual Conference of the International Speech Communication Association (Interspeech)*, 2009.

[28] Aditi Sharma Grover, Madelaine Plauché, Etienne Barnard, and Christiaan Kuun, "HIV health information access using spoken dialogue systems: Touchtone vs. Speech," *IEEE/ACM International Conference on Information and Communication Technologies for Development (ICTD)*, 2009.

[29] Tebogo Gumede and Madelaine Plauché, "Initial fieldwork for LWAZI: A telephone-based spoken dialog system for rural South Africa," in *Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages*, 2009, pp. 59 – 65.

[30] Tanja Schultz, "GlobalPhone: A multilingual speech and text database developed at Karlsruhe University," *International Conference on Spoken Language Processing (ICSLP)*, pp. 345 – 348, 2002.

[31] Douglas B. Paul and Janet M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the workshop on Speech and Natural Language*, 1992, pp. 357 – 362.

[32] Meraka-Institute, "Lwazi ASR corpus," 2009, Online: http://www.meraka.org.za/lwazi.

[33] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland, "The HTK book. Revised for HTK version 3.3," 2005, Online: http://htk.eng.cam.ac.uk/.

[34] "The CMU pronunciation dictionary," 1998, http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[35] Marelie Davel, Etienne Barnard, and Louis Joubert, "Research and development of a phoneme-recognition approach to spoken language identification," *CSIR Technical Report*, 2006.

[36] Paul Boersma and David Weenink, "Praat: doing phonetics by computer (Version 5.1.04) [Computer program]," 2009, Retrieved from http://www.praat.org/.

[37] Neil Kleinhans, "Channel normalization for speech recognition in mismatched conditions," *Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, vol. 19, pp. 66 – 70, 2008.

[38] Marius Peché, Marelie Davel, and Etienne Barnard, "Phonotactic spoken language identication with limited training data," *Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 1537 – 1540, 2007.

[39] Marius Peché, Marelie Davel, and Etienne Barnard, "Porting a spoken language identification system to a new environment," *Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, pp. 58 – 62, 2008.