Lecture Transcription Systems in Resource-Scarce Environments

By

Pieter Theunis de Villiers

Dissertation submitted for the degree

Scientiae Magister in Computer Science

at the

Vaal Triangle campus

of the

NORTH-WEST UNIVERSITY

Advisor: Professor Etienne Barnard

May 2014

ACKNOWLEDGEMENTS

Great thanks go out to Dr. Charl van Heerden, Prof. Etienne Barnard, Dr. Marelie Davel and Mr. Petri Jooste for their valuable guidance and support. A great thank you also to the NRF (National Research Foundation) for the funding provided throughout the duration of this project.

OPSOMMING

Lecture Transcription Systems in Resource-Scarce Environments

deur

Pieter Theunis de Villiers

Adviseur: Professor Etienne Barnard

Noordwes-Universiteit

Scientiae Magister in Rekenaarwetenskap

Die neem van klasnotas in 'n lesingsaal is 'n fundamentele taak wat daagliks deur leerders uitgevoer word. Hierdie aantekeninge voorsien leerders van waardevolle studiemateriaal vir aflyn gebruik, veral in gevalle waar moeiliker onderwerpe bespreek word. Daar is bevind dat die gebruik van klasnotas beide studente se leerervaring verbeter, sowel as 'n algehele verbetering in akademiese prestasie teweeg bring. In 'n onlangse studie is 'n toename van 10.5% in studente se uitslae aangeteken nadat hulle voorsien is van multimedia klasnotas. Hierdie resultate is nie heel onverwags nie, aangesien daar al voorheen bevind is dat die suksesvolle oordrag van inligting aan die mens toeneem wanneer inligting auditief sowel as visueel verskaf word.

Alhoewel die neem van klasnotas dalk na 'n eenvoudige taak klink, sukkel studente met gehoor-, visuele-, fisiese- en leergestremdhede, of selfs andertalige luisteraars geweldig hiermee en vind dit soms selfs ondoenbaar. Daar is bevind dat selfs studente met geen gestremdhede die neem van klasnotas as tydrowend ervaar; hulle vind dit ook uitdagend om klasnotas te neem en terselftertyd te konsentreer op wat die dosent verduidelik. Hierdie bevinding word dan ook beaam deur 'n studie waar daar bevind is dat tersiêre studente slegs ~40% van 'n dosent se lesing kon aanteken. Dit is dus redelik om te verwag dat 'n outomatiese stelsel wat klasnotas neem voordelig sal wees vir alle leerders.

Lesingtranskripsiestelsels word gebruik in leeromgewings om hulp te verleen aan leerders deur intydse transkripsies van die lesing te voorsien, of selfs video opnames en transkripsies, vir aflyn gebruik beskikbaar te stel. Sulke stelsels is reeds suksesvol in ontwikkelde lande geïmplementeer waar al die nodige hulpbronne maklik verkrygbaar is. Hierdie stelsels word gewoonlik ontwikkel deur gebruik te maak van honderde tot selfs

duisende ure van spraak, terwyl die gepaardgaande taalmodelle afgerig word deur gebruik te maak van miljoene of selfs honderde miljoene woorde. Hierdie hoeveelhede data is oor die algemeen nie beskikbaar in ontwikkelende lande nie.

In hierdie verhandeling ondersoek ons ’n aantal benaderings vir die ontwikkeling van bruikbare lesingtranskripsiestelsels in hulpbron beperkte omgewings.

Ons fokus op verskillende benaderings om voldoende hoeveelhede goed getranskribeerde spraak vir die afrig van akoestiese modelle te bekom deur van datastelle gebruik te maak wat min tot geen transkripsies het nie. Een benadering ondersoek die gebruik van dinamiese programmering foneem-string belynings metodes, met die doel om soveel moontlik bruikbare transkripsies te onttrek vanuit benaderde transkripsies. Ons vind dat taal-spesifieke akoestiese modelle optimaal is vir hierdie doel, maar rapporteer ook belowende resultate wanneer akoestiese modelle van ’n ander taal gebruik word vir aanvanklike belynings.

’n Ander benadering behels die gebruik van “geen-toesig” metodes. Hier word ’n aanvanklike lae-akkuraatheid herkenner gebruik om ’n stel ongetranskribeerde data outomaties te transkribeer. Goed herkende segmente word dan geïdentifiseer en onttrek deur gebruik te maak van ’n woord waarskynlikheids grens. Die nuut herkende data word dan saam met die aanvanklik getranskribeerde data gebruik om ’n nuwe stelsel af te rig, ten einde die algehele akkuraatheid te verhoog. Die aanvanklike herkenner is afgerig deur van slegs 11 minute getranskribeerde data gebruik te maak. Na ’n paar iterasies van geen-toesig afrigting is ’n merkbare toename in akkuraatheid waargeneem (47.79% woord-fouttempo tot 33.44% woord-fouttempo). Soortgelyke resultate is egter ook gevind (35.97% woord-fouttempo) waar die aanvanklike stelsel op ’n groot spreker-onafhanklike korpus afgerig is.

Bruikbare taalmodelle is ook afgerig deur van so min as 17955 woorde gebruik te maak; dit het egter gelei tot heelwat veral tegniese woorde wat nie in die taalmodel woordeskat voorkom nie. Hierdie probleem is aangespreek deur middel van taalmodel interpolasie. Daar is gevind dat taalmodel interpolasie veral voordelig is in gevalle waar onderwerp-spesifieke data (soos “Powerpoint” lesings en boeke) beskikbaar is.

Ons stel ook ons NWU lesingtranskripsiestelsel bekend, wat ontwikkel is vir gebruik in leeromgewings en wat ontwerp is deur van ’n kliënt/bediener argitektuur gebruik te maak.

Gebaseer op die resultate in hierdie studie is ons vol vertroue dat bruikbare modelle vir gebruik in lesingtranskripsiestelsels, ontwikkel kan word in hulpbron-beperkte omgewings.

Sleutelwoorde - akoestiese modellering, outomatiese spraakherkenning, taal modellering, lesing transkripsie, geen-toesig afrigting

SUMMARY

Lecture Transcription Systems in Resource-Scarce Environments

by

Pieter Theunis de Villiers

Advisor: Professor Etienne Barnard

North-West University

Scientiae Magister in Computer Science

Classroom note taking is a fundamental task performed by learners on a daily basis. These notes provide learners with valuable offline study material, especially in the case of more difficult subjects. The use of class notes has been found to not only provide students with a better learning experience, but also leads to an overall higher academic performance. In a previous study, an increase of 10.5% in student grades was observed after these students had been provided with multimedia class notes. This is not surprising, as other studies have found that the rate of successful transfer of information to humans increases when provided with both visual and audio information.

Note taking might seem like an easy task; however, students with hearing impairments, visual impairments, physical impairments, learning disabilities or even non-native listeners find this task very difficult to impossible. It has also been reported that even non-disabled students find note taking time consuming and that it requires a great deal of mental effort while also trying to pay full attention to the lecturer. This is illustrated by a study where it was found that college students were only able to record ~40% of the data presented by the lecturer. It is thus reasonable to expect an automatic way of generating class notes to be beneficial to all learners.

Lecture transcription (LT) systems are used in educational environments to assist learners by providing them with real-time in-class transcriptions or recordings and transcriptions for offline use. Such systems have already been successfully implemented in the developed world where all required resources were easily obtained. These systems are typically trained on hundreds to thousands of hours of speech while their language models are trained on millions or even hundreds of millions of words. These amounts of data are generally not

available in the developing world.

In this dissertation, a number of approaches toward the development of LT systems in resource-scarce environments are investigated.

We focus on different approaches to obtaining sufficient amounts of well transcribed data for building acoustic models, using corpora with few transcriptions and of variable quality. One approach investigates the use of alignment using a dynamic programming phone string alignment procedure to harvest as much usable data as possible from approximately-transcribed speech data. We find that target-language acoustic models are optimal for this purpose, but encouraging results are also found when using models from another language for alignment.

Another approach entails using unsupervised training methods where an initial low-accuracy recognizer is used to transcribe a set of untranscribed data. Using this poorly transcribed data, correctly recognized portions are extracted based on a word confidence threshold. The initial system is retrained along with the newly recognized data in order to increase its overall accuracy. The initial acoustic models are trained using as little as 11 minutes of transcribed speech. After several iterations of unsupervised training, a noticeable increase in accuracy was observed (47.79% WER to 33.44% WER). Similar results were however found (35.97% WER) after using a large speaker-independent corpus to train the initial system.

Usable LMs were also created using as few as 17955 words from transcribed lectures; however, this resulted in large out-of-vocabulary rates. This problem was solved by means of LM interpolation. LM interpolation was found to be very beneficial in cases where subject-specific data (such as lecture slides and books) was available.

We also introduce our NWU LT system, which was developed for use in learning environments and was designed using a client/server based architecture.

Based on the results found in this study we are confident that usable models for use in LT systems can be developed in resource-scarce environments.

Keywords - acoustic modeling, automatic speech recognition, language modeling, lecture transcription, unsupervised training

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Contents

Learning disabilities are a worldwide problem that prevent many children from reaching their full academic potential. In the United Kingdom (UK) for example, 2.35 million children aged 6–21 were reported to have disabilities (Anon., 2011). This problem is not restricted to children, though: according to the *National Institute on Disability and Rehabilitation Research*, 15% to 20% of randomly selected people can have impairments considered as disabilities (Bain et al., 2002:193). In 1999/2000, around 4% of 677100 students that enrolled in 172 institutions in the UK for their first year were known to have disabilities (Bain et al., 2002:193). These figures include both deaf students as well as students with other disabilities who have difficulty generating their own class notes. These learners often need more time to process learning material a lecturer presents (Ranchal et al., 2013:2). If such students are provided with supplemental learning material, they will be able to review a particular lecture's content as often as required, and at a convenient time (Ranchal et al., 2013:7). Supplemental learning materials may include among others lecture transcripts, video recordings and class notes. These have all been found to enhance both learning and teaching processes (Ranchal et al., 2013:1).

Class notes have been identified as one of the most requested supplemental learning aids by students with disabilities (Ranchal et al., 2013:2). Students that acquire and utilize class

notes have been proven to have a better learning experience and an overall higher academic performance (Ranchal et al., 2013:9). In a study conducted by Ranchal et al. (2013:9), an increase of 10.5% in student grades was observed, after these students had been provided with multimedia class notes. This is not surprising, as other studies have found that the rate of successful transfer of information to humans increases when provided with both visual and audio information (Anusuya & Katti, 2009:195).

Class note taking is a fundamental task performed by students on a daily basis. Note taking is performed in various ways, ranging from hand written notes to note taking on PC's (Kawahara, 2010:4; Kawahara et al., 2010:626). The note takers are generally student volunteers, as professional stenographers are too costly for everyday deployment (Kawahara, 2010:4; Kawahara et al., 2010:626). This task can become quite challenging for the students though, as it is time consuming and requires significant mental effort while the students are also paying full attention to the lecturer (Ranchal et al., 2013:3). For example, in a study conducted by Ranchal et al. (2013:3), it was found that college students taking notes were only capable of capturing ~40% of the information presented in a lecture. Another study painted an even bleaker picture, with 2 volunteers capturing only 20–30% of a spoken lecture (Kawahara et al., 2010:626). For more difficult subjects, such as science, these students required assistance with note taking (Ranchal et al., 2013:3). This task is even more challenging for students with learning disabilities, students who are deaf, or students attending classes in a language other than their mother tongue. This is corroborated by studies which found non-disabled students to generate up to 70% more lecture notes than disabled students (Ranchal et al., 2013:2).

The availability of class notes is thus clearly beneficial, but generating them is time consuming, expensive and often an unacceptable burden on the student volunteers who have to generate them. Therefore, automatic means of generating such class notes is a potentially rewarding endeavour. A modern technology which has been shown to address this problem, is Automatic Lecture Transcription, from here on referred to simply as Lecture Transcription (LT).

1.1 LECTURE TRANSCRIPTION

Lecture transcription employs modern technologies to automate the process of transcribing a lecturer's speech. The transcriptions can be presented to students in real time (visual input), or as supplementary learning material (offline). A typical classroom equipped with LT, will consist of an automated system that takes the speech of the lecturer as input, and outputs the

transcription of the recognized speech on a dedicated screen in real time.

The benefit of LT in the developed world is well understood: (Bain et al., 2002:192; Kheir & Way, 2007:264) have found LT to be very rewarding for both students with disabilities (students having trouble generating their own class notes, such as deaf students), as well as students without any disabilities. During an experiment conducted by Kawahara et al. (2010:628), a hearing impaired student (used as their test subject) reported that LT provided significantly more content compared to that obtained from note-takers. In (Kheir & Way, 2007:264), after implementing a LT system, a hearing impaired student was found to participate in a class discussion for the very first time. We believe that the potential benefit of LT systems may even be greater in the developing world, where lower literacy and a larger degree of multilingualism are more prevalent than in developed countries.

Implementing a LT system however, is a non-trivial procedure, with the development of the underlying automatic speech recognition (ASR) system being the main challenge. State-of-the art ASR systems, which will be discussed in Chapter 2, typically require hundreds of hours of speech to train acoustic models and millions of words to estimate reliable language models. These resources are necessary to create accurate transcriptions, but are expensive to collect. The resources necessary to build such systems in many languages of the developing world are however, either non-existent, or insufficient to reach the useful accuracy levels of resource-rich language LT systems.

LT systems are clearly very beneficial to learners, especially those with learning disabilities. The potential impact in the developing world is tremendous, but these benefits have thus far been out of reach, mainly due to resource constraints. In this dissertation, we will take some steps towards this goal of building LT systems with significantly fewer resources than which is typically required. We will investigate approaches to building language models with as little as 18000 words in Chapter 4. In Chapters 5 & 6, we show that acoustic models can be trained using lectures that are either partially transcribed, or completely untranscribed, when starting with as little as 11 minutes of transcribed data. Using our best acoustic and language models, we show that word error rates (WERs) of 35% on real-world lectures are achievable.

1.2 OBJECTIVES, HYPOTHESES AND OUTLINE

The main objective of this dissertation is to investigate ways in which ASR acoustic and language models can be built with significantly less resources than state-of-the-art, successfully deployed systems, while still operating at useful accuracy levels. The following hypotheses

are investigated:

1. Usable language models can be trained from a combination of resources one may expect in the developing world.

2. Data harvesting can be employed to generate enough usable data for building acoustic models suited for LT.

3. Unsupervised training approaches can be employed to utilize untranscribed lectures towards training more accurate acoustic models.

The rest of this dissertation is organized as follows: in Chapter 2, we will review relevant literature, describe a few existing LT systems, and focus on a number of training methods found useful for training LT systems in the past. The corpora used in this dissertation are then introduced and discussed in Chapter 3. Chapter 4 discusses language modeling with limited resources, while Chapters 5 and 6 focus on two approaches (supervised and unsupervised) for training acoustic models with limited resources. We discuss combining these components into a live LT system in Chapter 7 and conclude and summarize the work presented in this dissertation in Chapter 8.

CHAPTER TWO

BACKGROUND

Contents

The use of LT systems was first introduced at Saint Mary's University in 1998 (Bain et al., 2002:192). This was primarily to study the concept of ASR systems in classrooms, in order to improve the learning experience for students with disabilities. The LT system was found to be beneficial for both disabled students, as well as non-disabled students; consequently a research project known as the "Liberated Learning Project" was launched. LT has since become a valuable addition in many lecture rooms.

A LT system consists of a back-end (ASR component) which decodes incoming speech, and a front-end which processes, views and stores the results. (This is an intentional over-simplification for the purposes of reviewing the most basic components; a LT system may

entail much more than these basic components, with for example, keyword spotting, online channel adaptation and on-the-fly language model interpolation to name a few.) In this chapter, we will first provide a brief history of ASR. A general review of ASR is then followed by a short overview of language modeling, acoustic modeling and pronunciation modeling. We will then provide an overview of existing LT systems and look at the different functions such systems may provide. The resources necessary for training LT systems are then discussed, followed by a discussion about associated costs, which is a significant stumbling block to widespread adoption of LT in the developing world. We will then conclude with a discussion on the importance and implications of recognition accuracy.

2.1 ASR HISTORY

Speech is the most common form of human communication, and significant time and effort has been invested to replicate this ability in machines (Anusuya & Katti, 2009:181), establishing the active field of research into speech recognition and processing.

The earliest example of actual speech recognition we could find, was that of the Radio Rex toy from the 1920's (Anusuya & Katti, 2009:189). Radio Rex was a dog which emerged from his house when called by his name. A spring which controlled his movement, was supposedly activated when recognizing the first formant of "eh" in Rex, which occurs at around 500Hz (Jurafsky & Martin, 2000).

ASR research has progressed steadily from the early Radio Rex days. While isolated-word recognizers were the main focus up until the 1960's, connected word recognition emerged subsequently. Researchers also started to address issues such as changing speaking rate (Anusuya & Katti, 2009:190). Large vocabulary speech recognition was pioneered by IBM in the 1970's, focussing on among others dictation and database queries. At the same time, AT&T Bell Labs started to work on speaker independent ASR, while the well-known CMU speech group focussed on among others speech understanding (Anusuya & Katti, 2009:190). Their Harpy system was also one of the first to incorporate graph searching (Anusuya & Katti, 2009:190).

One of the big breakthroughs in speech recognition occurred in the 1980's with the shift from template-based to statistical modeling approaches for acoustic modeling; the hidden Markov models (HMMs) (Anusuya & Katti, 2009:191) approach was widely adopted. (The HMM was developed by Lenny Baum of Princeton University in the early 1970's (Anusuya & Katti, 2009:191)). Neural networks, which had not been widely used since the 1950's due to practical problems, were reintroduced in the 1980's.

Since the 1980's, many diverse aspects of speech recognition have been investigated, ranging from robustness to noise to decreasing an ASR system footprint and confidence scoring. New features and feature processing techniques have also been developed, with the most prominent probably being the use of multi-layer perceptrons (MLPs) to generate improved features from, more traditional features (Morgan & Bourlard 1995). Today, a popular approach involves creating a bottle-neck in the MLP architecture, hence the name "bottle-neck features". Deep neural networks (DNNs) (Mohamed et al., 2012) have also recently received significant attention as an alternative acoustic modeling technique. Other noticeable shifts that occurred over the last 20 years include the ability to recognize conversational speech, the ability to handle vocabularies of up to millions of words (Schalkwyk et al., 2010) and the widespread adoption of speech recognition in everyday applications, for example, Google's Voice Search and Apple's Siri on smart phones.

After decades of research, and many breakthroughs in the field of ASR technologies, many issues still remain that need to be resolved related to the performance of ASR systems.

2.2 OVERVIEW: ASR

An ASR system consists of an acoustic model, a language model, a pronunciation model and a decoder which uses the other three models to transform incoming speech to text. Below, a holistic view of the ASR process is given, followed by a more in-depth discussion about acoustic, language and pronunciation models respectively.

Acoustic models are composed of a set of statistical models representing the various sounds of a language to be recognized (Gales & Young, 2008:197). One significant benefit related to the use of statistical models is that the required models can be trained automatically from a corpus of transcribed speech (Gales & Young, 2008:198). HMM's and DNN's are both popular statistical models. These models provide a simple and effective framework for modeling the time-varying nature of speech and are ideal for use in ASR systems (Gales & Young, 2008:195).

Spoken words are composed of units of sound, called phones (Gales & Young, 2008:201). The word "dogs" for example, is composed of the phones /d/ /Q/ /g/ /z/. In isolation, these phones are also known as monophones. Phones are heavily influenced by the preceding or succeeding phones, though. For example, the pronunciations for the words "cat" and "hang" may use the same vowel for "a", yet in practice they are quite different "a" sounds due to the influence of the preceding and succeeding consonants (Gales & Young, 2008:206). For this reason, more context is typically used when modeling phones; in the case where the

preceding and following phones are taken into account, this model is known as a triphone model. One such model is created for every phone together with all possible corresponding left and right neighbours (Gales & Young, 2008:207).

A typical acoustic modeling strategy is to model each phone with an HMM (Gales & Young 2008) (2008:203). An example of a simple 5 state HMM is shown in figure 2.1. An HMM is a finite state machine that may change its current state with every time step (typically 10 milliseconds); an HMM has transition probabilities, which describe the probability of transitioning from one state to another probabilistically. For each input speech vector (or speech frame), a state transition may thus take place to either the next state, or it may remain in the current state.

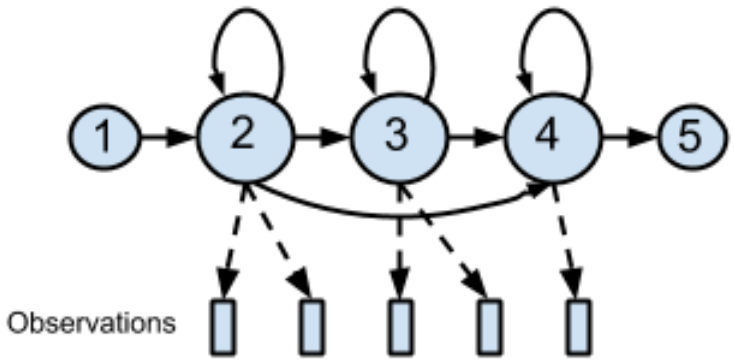

Figure 2.1: hidden Markov model

The vectors of numbers that represent the input speech are called *features* and the process of creating them is called feature extraction. These features are extracted from an input audio waveform, and feature extraction takes place prior to recognition. In the process, the audio waveform is converted into a sequence of fixed size acoustic vectors (Gales & Young, 2008:200). The size of the acoustic vectors, also know as the *window size* is typically 25 milliseconds (ms). It is also standard practise for these windows to overlap, with forward time steps typically 10ms (Gales & Young, 2008:202).

The recognition process entails finding the best or least cost path through a recognition graph, from the start state (or node) to the end state. The algorithm used to find the best path is known as the *Viterbi algorithm*. Here, each node will represent the log probability when observing a specific frame in a particular model's state, while each arc corresponds to the log transition probability in the HMM (Young et al., 2009:9). The best path through the matrix is determined by selecting the path resulting in the largest log probability value (Young et al., 2009:9-10). More details on the Viterbi algorithm can be found in the section "Recognition

and Viterbi Decoding" in Young et al. (2009:9-10).

In the next three sections, we discuss each of the three models that comprise an ASR system in more detail.

2.2.1 LANGUAGE MODELING

The language model (LM) is a representation of the possible word sequences that a system can recognize. Various types of LMs exist; we focus exclusively on statistical LMs, and in particular on backoff n-gram models with Kneser-Ney smoothing, where *n* refers to the length of the word sequence being modeled. Typical values of *n* range from 1 – 5. For an excellent overview of LMs and in particular a comparison of different smoothing techniques, the reader is referred to Chen & Goodman (1999).

The LM is trained on large text corpora and stores the probabilities associated with different possible word sequences. The use of language modeling in an ASR system is well known to significantly increasing the accuracy of the speech recognizer (Munteanu et al., 2007:2355).

For Japanese LT systems, it has been found that LM *adaptation* can also be beneficial. In an experiment conducted by Nanjo & Kawahara (2003), it was found that the WER could be decreased from 33.1% to 31% by adapting a baseline LM using well recognized utterances. By combining LM adaptation with pronunciation variation modeling (where the context of a word determines which variants are allowed), the WER was further reduced to 28.7%. LM adaptation is outside the scope of this work, but it could be interesting for future work if one has access to many untranscribed lectures.

The particular data, which is used to estimate LMs, is very important; having much target application data is ideal. In a typical LT scenario however, each lecture may contain highly technical terms which are very specific to that field. Using general text corpora may thus result in these important terms being out of vocabulary (OOV). Several approaches have been investigated to supplement a general text corpus with lecture specific text. According to Park et al. (2005:497), lecture presentations for example, make use of relatively small vocabularies, but these vocabularies contain highly specialized words related to the topic and field of the lecture. These topic-specific terms may also be obtained from other relevant sources such as textbooks, lecture notes, journal articles and of course, transcriptions of actual lectures. While subject-specific sources such as textbooks or presentation slides will contribute most of the subject-specific words, this type of material will lack many words or phrases commonly used in conversational or spontaneous speech (Park et al., 2005:498). Thus, the source material used for building a LM should be compounded from both spoken

and written text sources. A popular approach to achieve this, is via LM interpolation: a LM is trained on each of the text sources, and the resulting LMs are interpolated, with the interpolation weights optimized on an (ideally application specific) development set.

After combining such different types of text sources, Park et al. (2005:500) found the addition of spontaneous speech text to reduce error rates, even though they provided considerably higher perplexity values when evaluated separately on the test set. In this case, Park et al. (2005:500) concluded lower error rates to be the result of fewer errors on function words and conversational speech, and not on keywords or key phrases.

2.2.2 ACOUSTIC MODELING

The acoustic model (AM) stores statistical representations of all acoustic sounds that words are composed of. It is trained on spoken audio (typically with corresponding transcriptions) and is used, in conjunction with a LM and a pronunciation dictionary, to hypothesize different acoustic sounds during recognition.

Traditionally, large amounts of recorded data together with their corresponding manually generated transcriptions were used to train ASR systems (Wessel & Ney, 2001:307). However, manually generating transcriptions is both time consuming and expensive. Because of this, as well as the abundance of untranscribed data in multiple forms, unsupervised training has become an attractive alternative to manual transcription (Lööf et al., 2009; Wessel & Ney, 2001). Unsupervised training typically requires only a small amount of transcribed acoustic training data; the initial recognizer will thus be less accurate. This recognizer is then used to create transcriptions of any untranscribed acoustic data. Well recognized pieces are then identified and extracted based on a confidence threshold and used (in combination with the original training data) to either adapt or retrain the AM. This can also be done in an iterative process (Lööf et al., 2009; Wessel & Ney, 2001). In (Lööf et al., 2009; Wessel & Ney, 2001; Kemp & Waibel, 1999), well recognized segments of data were extracted based on a word confidence measure. It was believed that setting the confidence threshold to a higher value would increase the probability of extracting useful data, that is, data more likely to contribute to the acoustic modeling. According to Wessel & Ney (2001:308), even though these extracted segments may in some cases not be correct, they will however contain elements with similar acoustic properties as those of the actual words.

Kemp & Waibel (1999:2725) performed unsupervised training by making use of only 30 minutes of transcribed and 50 hours of untranscribed data. This resulted in a decrease in WER from 32.1% to 20.6%. Kemp & Waibel (1999:2727) found using a confidence threshold of 0.5 to select recognized word sequences for adaptation/retraining, to produce

the most accurate output based on their data.

Wessel & Ney (2001:310) made use of only 1.2 hours of transcribed data to train their initial recognizer. This was then used in an iterative process (7 iterations) with a confidence threshold of 0.7 to recognize 70.8 hours of untranscribed speech and retrain a new system. Using two evaluation sets (Broadcast News '96 and Broadcast News '98), they found a decrease in WER from 71.3% to 38.3% and from 65.5% to 29.3% respectively. They also reported a reduction in WER as the amount of initial transcribed data was increased.

In the absence of large target-language speech corpora, cross-language bootstrapping has been investigated as an alternative to obtain acoustic models (Schultz & Waibel, 2001). Even though target language acoustic models provide better results, the method of cross-language bootstrapping was found to provide comparable results to that of target-language acoustic models (van Heerden et al., 2011:141). In an experiment conducted by Lööf et al. (2009), the authors did not make use of any transcribed target-language (Polish) acoustic training data to train their initial recognizer. Instead, they made use of an existing Spanish recognizer, ported to Polish by means of manually constructed phone mappings. Using an iterative unsupervised training approach in combination with speaker adaptation methods, they found a decrease in WER from 63.4% to 20% on their evaluation set. There is thus ample evidence that unsupervised training has the advantage of requiring much less transcribed data than supervised training, making it faster and more cost effective to create acoustic models.

Trancoso et al. (2006:282) performed a number of experiments to determine the effect that acoustic model speaker adaptation methods can have on a LT ASR system when used in combination with LM topic adaptation. These experiments were performed using recordings from two courses; "economic theory I" (ETI) and "production of multimedia contents" (PMC). The WER for the baseline recognizers of these two courses were 56.4% and 63.6% respectively. Using a single lecture of each course, acoustic model speaker adaptation was performed in conjunction with and without LM topic adaptation. Using speaker adaptation with LM adaptation for both courses, ETI and PMC, resulted in WERs of 44.7% and 44.8% respectively. Using speaker adaptation without LM adaptation for both courses, ETI and PMC, resulted in WER's of 45.4% and 48% respectively.

Glass et al. (2007:2553) have also found speaker- and topic adaptation to significantly reduce WERs. For a specific physics lecturer, LM adaptation using physics textbooks and 40 related lectures (from other lecturers) resulted in a small WER reduction (32.9% to 30.7%). Supervised adaptation using 29 hours of previous lectures from this lecturer, however, resulted in a significant reduction in WER (30.7% to 17%).

Another approach, *phone-based dynamic programming* (PDP), was found to be some-

what successful for semi-supervised training. This approach identifies well recognized portions by comparing the result of a forced alignment with that of a free decode, using a variable cost matrix (Barnard et al., 2011). An alternative approach to identifying well transcribed portions from approximately transcribed lectures entails background modeling. This was also found to be effective in two experiments conducted by van Heerden et al. (2011:142) where approximate transcriptions were used to train ASR systems. In the first experiment, this method was applied to an English bootstrapped corpus, and in the second, to an Afrikaans lecturing corpus. Both these experiments showed a reduction in phone error rate (PER), as this background model was used to place optional garbage markers between words, in order to absorb disfluencies such as incorrect or untranscribed portions.

2.2.3 PRONUNCIATION MODELING

A pronunciation lexicon contains all words the system can recognize, together with their corresponding acoustic units.

These acoustic units may either be phoneme-based or grapheme-based. Much work has been done to address the different problems associated with each: using graphemes instead of phonemes as acoustic units has been shown to be a viable alternative, especially for languages with a regular grapheme-to-phoneme relationship. Specific advances in this field include work to automatically create viable "grapheme" pronunciations for words which do not follow a (otherwise) regular grapheme-to-phoneme relationship (Basson & Davel, 2012).

Large pronunciation lexicons, such as OALD (Mitten, 1992) (British English) and CMU-Dict (Anon., 1998) (American English), have been developed for resource-rich languages. For under-resourced languages, however, large pronunciation lexicons are typically not available, and creating one is both time consuming and may be prohibitively expensive.

Sophisticated tools such as *Dictionary Maker* (Meraka-Institute, 2009) have been developed to enable mother-tongue speakers to rapidly build pronunciation lexicons in resource-scarce languages (Davel & Martirosian, 2009). Tools such as *Dictionary Maker* assist the users by allowing them to listen to the words and select the appropriate acoustic units. Using a machine learning algorithm, *Dictionary Maker* has the ability to predict the acoustic units for new words, which can then be altered by the user if necessary.

Algorithms such as Joint Sequence models (Bisani & Ney, 2008) and Default & Refine (Davel & Barnard, 2008) have also been developed to learn grapheme-to-phoneme rules from a pronunciation lexicon, and to use these rules to predict pronunciations for new words.

In this work, we will make use of phonemes as our acoustic units, and use the Default & Refine algorithm to predict the pronunciations of any unknown words.

2.3 OVERVIEW OF EXISTING LT SYSTEMS

LT systems have already been implemented in a number of learning environments. In this section, we provide an overview of a couple of these systems.

- **Villanova Speech Transcriber and Dictionary Building Software**

  The *Villanova University Speech Transcriber (VUST)* makes use of multiple dictionaries. These include 1) a general language dictionary, and 2) domain-specific dictionaries, containing any technical or domain-specific words (Kheir & Way, 2007:263). VUST works together with *Dictionary Building Software (DiBS)* that monitors textual input and scans for domain-specific words. Once new words are found, they are added to the domain-specific dictionary. Therefore, the users of the system have the ability to manually add words to the dictionary once they realize some words are not being recognized (Kheir & Way, 2007:263).

  During recognition, whenever silences are detected for a certain period (based on a predetermined threshold), VUST interprets them as the end of a sentence and replaces them with full stops (Kheir & Way, 2007:263). This is a technique known as *end-pointing* (Shriberg, 2005). This technique is not always robust when it comes to end-of-sentence detection though; some lecturers tend to string their sentences together without pauses, except in cases of hesitations, or when the lecturer waits for feedback from the students. This makes it difficult to accurately identify sentences. In addition to end-of-sentence detection, VUST can also detect the end of paragraphs by interpreting longer pauses as such.

  VUST also enables students to follow a live stream of the lecture via the Internet, by means of a Java applet. This video streaming output can then be controlled by the lecturer from the classroom terminal.

- **MIT Lecture Browser**

  This web-based interface described by Glass et al. (2007:2555), allows users to search, browse and retrieve lectures, as well as view them live through video streaming from the server. Users have the ability to start playback from multiple areas by means of a collection of play buttons on a timeline. As the lectures are viewed, the words in the transcription are underlined to indicate which words are being said. This feature is made possible by time-aligned transcriptions (Glass et al. 2007). Whenever a user

skips through the video, it will thus be easy to keep track of the exact location in the corresponding transcriptions. Keywords that were used in the search query (for the particular lecture being viewed) are also highlighted in the transcriptions.

Glass et al. (2007:2556) also proposed using a "*Wikipedia-style*" online editing scheme, where users will be able to manually correct the transcriptions as needed.

- **Lecturer and ViaVoice**

  Bain et al. (2002:193) used a specially designed ASR system, *Lecturer*, together with IBMs *ViaVoice* technology. This system requires each lecturer to first adapt the ASR system to recognize his/her own voice. This adapted set of models is referred to as the lecturer's "voice profile", which is then used to convert speech to text. This profile continuously needs to be updated and expanded. After each session, the transcription has to be edited to eliminate and correct recognition errors.

- **Julius and IPTalk 1**

  Julius and *IPTalk* are both open-source software packages (Kawahara, 2010:4). Lecture speech is captured by a microphone and sent to the system on which the speech recognition engine resides. In this case *Julius* was proposed as the free ASR engine. *IPTalk*, which is a software captioning program used in Japan by hearing impaired people, is then used to combine the recognition results with the recorded lecture (Kawahara, 2010:4).

- **Julius and IPTalk 2**

  Kawahara et al. (2010:627) are developing a LT system mainly for hearing impaired students in university classrooms. Lecture speech is captured by a wireless pin microphone and sent to a computer system located in the same room for decoding, in their case the *Julius* speech recognizer. After recognition has been completed, the generated output is first sent to a post-editing screen where a user can correct the output. A second post editing user may also be connected to the system by means of another terminal. When corrected, the output text is sent to a LCD screen visible to the students. This final presentation of the results is performed using *IPTalk*.

  Although this post-editing function may seem time consuming, (Kawahara et al., 2010:628) found that it takes a user only 8.91 seconds on average to select and correct

an erroneous utterance. More time will necessarily be required for lower accuracy ASR systems.

The existing LT systems previously described, thus perform much more than one would typically associate with the term LT; here we briefly list a few to summarize this section:

- Storing and indexing of lectures and their corresponding transcriptions. This is useful as for example supplemental study material or future corpus training data.
- Live video and transcription feeds (across campus / Internet).
- Ability to search for and retrieve lectures by keywords.
- Ability to manually edit and correct recognized speech.

2.4 RESOURCES FOR LECTURE TRANSCRIPTION

In this section, we describe typical resources required for training LT systems, as well as the associated cost of creating these resources. This is one of the prohibiting factors for adoption of LT in developing countries.

LT systems are typically trained on hundreds of hours of data while their LMs are trained using text corpora containing millions to hundreds of millions of words.

Park et al. (2005:497) used 147 hours of transcribed data to develop their audio information retrieval system. An additional 20 hours of data were also required: 10 hours for acoustic model adaptation and 10 hours for testing purposes (Park et al., 2005:498).

Kawahara (2010:3) developed their ASR system, which is used for parliamentary meetings and classroom lectures, using about 320 hours of acoustic training data: 225 for training and 95 for adaptation purposes. Their LM was built on text collected from official meeting records amounting to 170 million words.

Glass et al. (2007) made use of roughly 121 hours of their 200 hours transcribed speech to train their American English system. Here, 1 to 30 hours of data was available per speaker for use during speaker adaptation. Their LM alone was trained on more than 6 million words.

From these examples, it is evident that large amounts of training data are used in resource rich LT systems. This of course has the benefit of very low WERs (such as 17% WER found by Glass et al. (2007:2555)). Data availability and the cost related to the collection and transcription of such large amounts of data are important factors to consider in resource-scarce environments.

2.4.1 COST OF DEVELOPMENT

Having hundreds of hours of speech and millions of words of text is evidently preferable when building LT systems, but collecting these resources may be prohibitively expensive. Transcribing speech is especially expensive, and may cost anything upwards of $US20 to $US150 per hour (these figures are considered to be significantly cheaper than previous transcription efforts) (Novotney & Callison-Burch, 2010). Audio collection approaches using smart-phones is an attractive alternative, as users are prompted to say specific utterances (De Vries et al., 2011; Hughes et al., 2010). This eliminates the need for audio transcription, although techniques to perform automatic quality control have to be employed (Davel et al., 2011).

Recently the natural language processing (NLP) community also started utilizing online labour markets such as *Mechanical Turk* to have their data transcribed by non-professional transcribers (Novotney & Callison-Burch, 2010). Mechanical Turk is a platform where thousands of online workers (called "Turkers") perform simple tasks that are difficult for computers, but easy for humans. These tasks are known as human intelligence tasks (HITs). Users of this system can have large amounts of transcriptions created by non-professional transcribers, at significantly cheaper rates than when using professional transcribers. In an experiment conducted by Novotney & Callison-Burch (2010:209), the authors investigated the willingness of turkers to complete tasks, based on the amount offered per task. Several experiments were performed where for each experiment, files were uploaded for transcription while the payment rate was reduced from the previous experiment. They found the turkers were willing to complete HITs (in this case the transcription of 10 utterances) for as little as $0.05 each, even though some complained about the low payment rate. Mechanical Turk is thus clearly a cost-efficient way of collecting large amounts of transcribed data (at variable quality of course). Novotney & Callison-Burch (2010:209) found the non-professional transcriptions to be only 6% worse than professionally-done transcriptions, for 0.03% of the normal price. However, this approach requires that workers fluent in the relevant languages should be readily available; this is typically not the case for under-resourced languages.

Much work has also been done in the field of unsupervised training, where very little to no transcriptions are available. By using this method of training, a small amount of transcribed data is used to train an initial system. This system can then be used to automatically create transcriptions of any untranscribed data, albeit at a much lower accuracy than manual transcriptions. Well recognized portions can then be extracted and used to retrain the system, and in turn improve on the automatic transcriptions.

In the rest of this dissertation, we explore different ways in which LT systems can be built

with minimal resources. There is always a trade-off between more resources and accuracy though, and understanding the consequences of a lower accuracy system is therefore important. We thus conclude this chapter with a short overview on the impact of LT recognition accuracy.

2.4.2 IMPACT OF RECOGNITION ACCURACY

The accuracy of the transcriptions provided by the system could easily affect other areas such as indexing and retrieval, segmentation, browsing of lectures (Glass et al., 2007:2556) and ultimately the end-user experience. Depending on the required level of accuracy, some of the transcriptions may need to be corrected, which is a very time consuming exercise. The required level of accuracy is a non-exact number though. Munteanu et al. (2007:2353) for example, found that users reported transcriptions from a system with a WER of 25% to be useful.

According to Bain et al. (2002:194), a lecturer's speaking rate can vary between 100 and 200 words per minute; this means that a lecturer with a speaking rate of 150 words per minute will produce approximately 9000 words in an hour-long lecture. At a WER of 20%, this means that there will be 1800 recognition errors in the transcription. Taking into account that it takes a person on average 4.07 seconds to select a recognition error, and another 4.84 seconds to correct it, it can easily take up to 3 hours to correct only 1 hour of lecture speech (Bain, Basson & Wald, 2002:194), even when such an accurate baseline recognizer is available.

State-of-the-art LT systems achieve WERs anywhere between 45% and 20% (Trancoso et al., 2006:281; Bain et al., 2002:194; Glass et al., 2007:2553; Kheir & Way, 2007:264). A large percentage of errors can be attributed to false starts, filled pauses, hesitations, mispronunciations, partial words, non-grammatical constructions and other artefacts that are common in everyday human communication (Bain et al., 2002:194; Glass et al., 2007:2553; Trancoso et al., 2006:281; Shriberg, 2005:1781). Domain-specific words are also a problematic category, as these words often do not occur in typical LM corpora (Glass et al., 2007:2553). This means that terms used in a Computer Engineering course will most likely contain very different technical words, as opposed to a statistical course, and vice versa. Speaker differences also influence accuracy, not only because of acoustic differences, but also because of the subtly different ways in which each speaker expresses and pronounces words (Nanjo & Kawahara, 2003). Non-lexical artefacts such as coughs, laughs and other environmental noises also influence the accuracy of the speech recognizer. Other factors that influence accuracy, are explained in more detail by (Anusuya & Katti, 2009:183). these

include:

- the environment (acoustic setting, noise conditions),
- transducer (microphone, telephone),
- speaker (age, gender, physical state),
- speaking style (tone, speed, spontaneous or isolated word),
- and vocabulary (available training data, specific or generic vocabulary).

If favourable conditions (read speech in a controlled environment with little to no background noise), for example, WERs as low as 2% have been observed (Bain, Basson & Wald, 2002:194). Conversational speech on the other hand, is a much more difficult task, with state-of-the-art English ASR systems trained on 2000+ hours of conversational speech with a LM trained on more than a billion words and a hand-crafted pronunciation dictionary, having WERs of ~15%.

CHAPTER THREE

CORPUS COLLECTION AND PROCESSING

Contents

In this chapter we will list and describe the different corpora used throughout the remainder of this dissertation. The collected text corpora (used for language modeling) as well as the collection of speech corpora are described.

3.1 TEXT CORPORA

A number of text corpora were collected for use as LM training material.

All collected text corpora were preprocessed by following these steps:

1. Convert to .txt format (if necessary).
2. Remove byte order marks.
3. Normalize apostrophes and remove diacritics.
4. Normalize both abbreviations and digits.
5. Remove all unwanted characters.

The different text corpora are described below.

3.1.1 LECTURER

This corpus consists of a number of transcriptions which were manually generated from the collected OS corpus (discussed in Section 3.2.6). This corpus contains spontaneous, subject-specific and speaker-specific transcriptions that will be useful for fine tuning a LM (either as a single LM, or via interpolation with a larger more generic LM).

Given the limited amount of data available in the collected OS corpus (discussed in Section 3.2.6), cross-validation was used to compute a generalized measurement of performance across the whole data set. Since a total of 6 folds of cross-validation were used, 6 LMs were required (the specific lectures used for each LM are shown in Table 3.3 and Table 3.4). The Lecturer corpus used for the creation of each LM consisted of 4 classes in each case (for example ~4.12 hours/17953 words in fold 1 of cross-validation).

3.1.2 OS BOOKS

The OS books corpus consists of multiple English online books related to *operating systems* subjects. These books were all collected in either html or pdf format and converted to plain text format. This corpus contained a large number of domain-specific words, but very little text representing spontaneous speech. This corpus contained a total of 1002827 words.

3.1.3 STUDY GUIDE

The study guide corpus is composed of all available 2012 English study guides (collected from the North-West University Vaal Triangle campus), related to any *Information Technology* course. This corpus is similar to the OS books corpus in that it contains many domain-specific words, but very little text representing spontaneous speech. This corpus contained a total of 157608 words.

3.1.4 YOUTUBE

The *Youtube* corpus consists of several transcriptions uploaded to, or automatically generated by for example Google (Liao et al., 2013). These include online tutorials on operating systems, as well as operating system related subjects provided by Google talks. Automatically generated transcriptions were manually checked and corrected. This corpus contains 277535 words, which includes many domain-specific words as well as words present in spontaneous speech.

3.2 SPEECH CORPORA

A number of speech corpora were collected for use as acoustic model training material. These corpora are described below.

3.2.1 *ALT* - AFRIKAANS LT CORPUS

The *ALT* corpus consists of 20 hours of Afrikaans lecture data from two broad subject areas; law and science/chemistry. Male lecturers account for 14 hours of speaker data and females for 6 hours. All audio data has been manually segmented into 5 minute segments, mainly to increase the speed of the alignment and decoding (van Heerden et al., 2011:141).

A single first-language Afrikaans speaker produced orthographic transcriptions of the *ALT* corpus; the transcriber was given the following instructions:

- Transcribe exactly what was said (do not correct for grammar, hesitations, etc).
- Use punctuation (,.?!) only to indicate sentence structure.
- Do not use quotation marks or brackets.
- Write out numbers in words instead of using digits 0-9.
- Mark foreign words with # (for example #inja).

All speakers are listed in Table 3.1 with their associated subjects, gender and amount of training and testing data in minutes.

Table 3.1: ALT speaker information with training and testing data in minutes. A speaker could only contribute to the test set if they had more than one lecture, as no single lecture was split between the train or the test set.

SPKR ID	Gender	Subject	Train	Test	Total
m001	male	sci	17	0	17
m002	male	sci	42	37	79
m003	male	sci	84	37.5	121.5
m004	male	sci	31	0	31
m005	male	sci	44	0	44
m006	male	sci	46	37	83
m007	male	sci	43	0	43
m008	male	sci	37	0	37
m009	male	law	26	23	49
m010	male	law	36	0	36
m011	male	law	35	35.5	70.5
m012	male	law	62.5	37.5	100
m013	male	law	57.5	0	57.5
m014	male	law	47	0	47
m015	male	law	27	0	27
f001	female	sci	39.5	23	62.5
f002	female	sci	46.5	43	89.5
f003	female	sci	25	0	25
f004	female	law	32.5	30.5	63
f005	female	law	61.5	36	97.5
f006	female	law	40.5	0	40.5

The pronunciation dictionary was created by

1. using a dictionary lookup (using the ANCHLT model's dictionary) for known Afrikaans words (443 words),

2. identifying English words with a dictionary lookup (840 words) and

3. using the Default & Refine (Davel & Barnard, 2008) algorithm to automatically generate pronunciations for the remaining 6735 words.

English words occur fairly frequently in the *ALT* corpus; they were automatically identified by a dictionary lookup and the pronunciation was then mapped to similar Afrikaans phones (van Heerden et al. 2011). These mappings are shown in Table 3.2. All names and foreign words (which were marked with # by the transcriber) were then manually verified

and corrected if necessary (these words are prone to automatic pronunciation prediction errors, as they often do not follow the same regular grapheme-to-phoneme structure as words from other languages).

Table 3.2: English to Afrikaans phone mappings - the conventions of the Lwazi phone set Anon. (2013*a*) are used.

Eng	Afr	Eng	Afr
3:	@	Q	O
e@	E	r\	r
ai	a i	tS	t S
au	a u	u:	u
d_0Z	d Z	U	u
i:	i	T	f
O:	O	D	v
Oi	O i		

3.2.2 *ANCHLT* - AFRIKAANS NCHLT CORPUS

This corpus consists of speech from 206 Afrikaans speakers (approximately equal numbers of males and females), with approximately 500 3-5 word read utterances per speaker. These utterances were mostly recorded in controlled environments. This amounts to approximately 100 hours of speech data. The vocabulary of this corpus consists of 9375 distinct words, predominantly from the government domain. This corpus closely resembles the *Baseline* corpus described in van Heerden et al. (2013).

3.2.3 *ASL* - AFRIKAANS SPOKEN LECTURES CORPUS

This corpus consists of 12 recorded lecturers (9 male and 3 female), from various domains. The lecture recordings varied in duration, ranging from less than 5 minutes to approximately 45 minutes.

Many of the recordings were found to have inaccurate and inconsistent transcriptions. One source of inconsistencies was the fact that the lectures were transcribed over a period of 4 years. This resulted in predictable inconsistencies with regard to transcription protocols for entities such as numbers and abbreviations. Disfluencies, repetitions and filled pauses were also not transcribed consistently (some transcribers would include them in detail, while others would transcribe as if the speech was fluent and grammatically correct). Spelling mistakes were also common, which can be detrimental to automatic pronunciation prediction

approaches. Another problem expected to be common in many resource-scarce environments is the frequent use of English words and informal speech during lectures.

These transcriptions were preprocessed using the process described below:

- The entire corpus was spell-checked using an Afrikaans spellchecker.
- Proper names were identified by inspecting all capitalized words. Once identified, pronunciations were manually generated.
- Abbreviations and acronyms were identified by considering all words with fewer than five letters and with at most one vowel. Pronunciations for both the spoken as well as the abbreviated form were then created and added to the dictionary.
- Numbers written as digits were normalized to their spoken form where there was no ambiguity. Where ambiguity exists (for example in the pronunciation of "100", where the "one" is often omitted), the number was replaced with a special token, with both corresponding pronunciations being allowed in the dictionary.
- Possible English words were identified from an in-house South African English pronunciation dictionary. Because there is non-negligible overlap between English and Afrikaans words (words present in both languages), both the English pronunciation and an Afrikaans pronunciation (generated by rules if necessary) were retained for such words.

Pronunciations for all remaining words not in a reference dictionary (Davel & De Wet, 2010), were automatically generated using the Default & Refine algorithm (Davel & Barnard, 2008).

3.2.4 *ENCHLT* - ENGLISH NCHLT CORPUS

The English NCHLT corpus consists of speech from 210 speakers (approximately equal amounts of males and females), with approximately 500 3-5 word utterances read per speaker. These utterances were mostly recorded in controlled environments. This amounts to approximately 100 hours of speech data. The vocabulary of this corpus consists of 9530 distinct words, from various domains. This corpus closely resembles the *Baseline* corpus described in (van Heerden et al., 2013).

3.2.5 *NCHLT* - AFRIKAANS NCHLT CORPUS

Similar to the ANCHLT corpus, this corpus [1] consists of speech from 185 Afrikaans speakers (approximately equal numbers of males and females), with approximately 500 3-5 word utterances read per speaker. These utterances were mostly recorded in controlled environments. This amounts to approximately 80 hours of speech data containing 4300 unique words. This corpus closely resembles the *Baseline* corpus described in (van Heerden et al., 2013).

3.2.6 *OS* - ENGLISH OPERATING SYSTEMS CORPUS

The English Operating Systems (OS) corpus was collected on the North-West University Vaal Triangle campus, using our NWU LT system (discussed in Chapter 7).

The OS corpus consists of a single male lecturer providing an OS course. While the lectures are presented in English, the lecturer's mother tongue is actually Afrikaans. He speaks English fluently, and with a typical South African English accent. This lecturer has been presenting this subject for the past few years and is thus able to arrive relatively "unprepared" as he is able to recall the subject matter from memory. The lectures therefore contain many false starts, corrections and hesitations. Furthermore, the lecture room is typical of a normal lecturing environment, where students are asking questions. There are also regular pauses (as the lecturer writes on the board) and a few Afrikaans utterances in between.

During the data collection phase, the lecturer was asked to wear a head mounted microphone as lecturers tend to move around in the class, turning their heads regularly while speaking (Trancoso et al., 2006:281). This audio data together with the video feed (from a connected webcam) was captured and stored in mp4 format by our NWU LT system (see Chapter 7).

The audio portions of the recorded mp4 files were extracted and converted to WAV format. The audio lectures were then split into much smaller audio segments, ranging from less than one second, to about 40 seconds in duration. Using smaller segments of data will result in faster alignment and decoding (van Heerden et al., 2011:141). The audio segmentation was performed using Sox (Anon., 2013*c*); recordings were segmented based on a leading silence of 0.5 seconds at an audio threshold of 1%, and a trailing silence of 0.8 seconds at an audio threshold of 1%.

[1] This corpus differs slightly from the ANCHLT corpus, as the experiments reported in this dissertation were performed over a period of 3 years, which coincided with the NCHLT corpus development and refinement. The NCHLT corpus will soon be released by the Language Resource Management Agency of South Africa (Anon., 2013*a*).

The entire OS corpus amounts to approximately 12 hours of data (12 classes ranging from 19 minutes to 84 minutes in duration). From this data, 6 classes were manually transcribed; 4 classes for use with the training of the LM, 1 class for the development or tuning set, and 1 class for the evaluation set. The remaining 6 classes were left untranscribed for use during unsupervised training. Given our small collection of OS data, all experiments were performed using 6-fold cross-validation.

Table 3.3 shows a summary of the collected OS data. It shows the ID of each recording (to which we will refer from here on), number of words (transcribed files), total duration, total duration after segmentation, whether or not it had been manually transcribed, and what each of them were used for during fold 1 of cross-validation. Note the recording "U6/T0" will be used as transcribed training data in some experiments, while used as untranscribed data for unsupervised training in other experiments.

To clarify the data distribution during the 6-fold cross-validation, Table 3.4 shows exactly which lectures were used for the LM, for the development set and for the evaluation set for each fold respectively.

Table 3.3: Description of all recordings in the OS corpus. Here we list the IDs assigned to each recording, total minutes in duration, total minutes in duration after segmentation, whether or not they were transcribed, and an example of how the data is to be used during the first fold of cross-validation

ID	#Words	Dur.(Total)	Dur.(Segmented)	Transcribed	Use(Fold 1)
U1	-	55	33	False	Unsupervised Training
U2	-	19	8	False	Unsupervised Training
U3	-	84	53	False	Unsupervised Training
U4	-	77	47	False	Unsupervised Training
U5	-	72	31	False	Unsupervised Training
U6/T0	1658	22	11	False/True	Unsupervised Training/Training
T1	5746	70	37	True	Language Model only
T2	5746	43	17	True	Language Model only
T3	4095	62	22	True	Language Model only
T4	5423	72	28	True	Language Model only
T5	7620	70	47	True	Development set
T6	7494	76	47	True	Evaluation set

3.2.7 *WSJ* - ENGLISH WALL STREET JOURNAL CORPUS

This corpus is used in an experiment to determine how closely one can approximate the language-specific results when using a well-trained model from a different language. In this experiment the target language was Afrikaans.

Table 3.4: OS data distribution for the different folds of cross-validation. IDs are listed in Table 3.3

Fold	Language Model	Development set	Evaluation set
1	T1, T2, T3, T4	T5	T6
2	T2, T3, T4, T5	T6	T1
3	T3, T4, T5, T6	T1	T2
4	T4, T5, T6, T1	T2	T3
5	T5, T6, T1, T2	T3	T4
6	T6, T1, T2, T3	T4	T5

This corpus contains American English spoken utterances with corresponding transcriptions. The CMU pronunciation dictionary was used; however, phone mappings had to be created for phones from both languages (Afrikaans and English) to come up with a common phone set. We employed linguistic knowledge to generate such a mapping. As the transcription conventions used in the CMU dictionary do not model the schwa (/ax/) separately (it is modeled as an unstressed variant of the other vowels that are marked explicitly in the dictionary), we first employed an interpolated phoneme mapping to identify likely occurrences of schwas. Specifically (using ARPABET notation) all the /eh r/, /uh r/, /uw r/, /ih r/, /iy r/ and /er/ samples were mapped to /eh ax r/, /uh ax r/, /uw ax r/, /ih ax r/, /iy ax r/ and /ax r/ respectively and the unstressed /ah/ mapped to /ax/ (retaining stressed /ah/ as /ah/). Once the dictionary was reformatted, each phoneme (or combination of phonemes) was mapped directly to their closest Afrikaans counterparts. 18 of the phonemes could be mapped directly, the remainder are listed in Table 3.5. Only two English phonemes - /dh/ and /th/ - were not used.

Table 3.5: Source models for Afrikaans where direct mappings were not available

English	Afrikaans	Example	
(ARPABET)	(SAMPA)	English	Afrikaans
iy ax; ih ax	i@	peer	geen
uw ax; uh ax	u@	poor	boom
ih; iy	i	kin, keen	sien
iy ax; ih ax	2:	peer	museum
ax	9	(SAE) this	put
ih; iy	y	kin, keen	vuur
hh	x	hand	gaan
aw	a u	allow	gauteng
ay	a i	abide	baie
ch	t S	choke	tjalie
oy	O i	ahoy	boikot
jh	d Z	joke	jazz

CHAPTER FOUR

LANGUAGE MODELING

Contents

The LM is a representation of the possible word sequences that a system can recognize. Various types of LMs exist; we focus exclusively on statistical LMs, and in particular on backoff n-gram models with Kneser-Ney smoothing, where *n* refers to the length of the word sequence being modeled. Typical values of *n* range from 1 – 5. For an excellent overview of LMs and in particular a comparison of different smoothing techniques, the reader is referred to Chen & Goodman (1999).

LMs incorporated into existing ASR systems of the developed world are typically trained on text corpora consisting of millions of words (Kawahara, 2010:3; Glass et al., 2007:2554). In resource-scarce environments where such large text corpora are often nonexistent, alternative sources have to be considered as training material.

In this chapter, we discuss LM results from previous work in collaboration with Petri Jooste, where the main focus was interpolation of different LMs in order to develop LMs for use in LT systems in resource-scarce environments. We will then discuss the development of LM generation in a similar fashion, by utilizing different (but relevant) sources of text data.

4.1 LANGUAGE MODEL INTERPOLATION FOR AFRIKAANS LT EXPERIMENTS

In a previous study (de Villiers et al., 2012) (conducted in collaboration with Mr. Petri Jooste, Dr. Charl J. van Heerden and Prof. Etienne Barnard), we conducted some initial experiments on the development of LMs for LT systems in resource-scarce environments.

Here, three different types of text corpora were collected (discussed in more detail in Chapter 3). This collection consisted of transcriptions of actual lectures, domain-specific university study guides, as well as a general text source. The transcriptions of lectures and study guides consisted of data from two broad subject areas: law and science/chemistry.

As might be expected, the in-domain transcriptions provided the best results with relatively low perplexity (PPL) and out-of-vocabulary (OOV) word counts. Low OOV rates were also observed when using the general text as well as the university study guide material; subsequent interpolation experiments confirmed that a better LM could be built using a combination of different text sources, via LM interpolation.

Based on these results, it was decided to use the transcriptions of lectures (LT) and University study guides (SG) to train LMs to be used for off-line lecture transcription. Our goal here is to investigate whether the observed LM improvements from interpolation could be translated to improved recognition performance. We decided to focus on the within-topic interpolation of the study guides and transcriptions, that is, we built two sets of LMs, one for the *law* domain and the other for the *sciences* domain. In each set, we investigate the effect of different interpolation weights; from 0.0 (at which value the study guides' contribution dominates) to 1.0 (where the model is dominated by the transcriptions). For consistency, we report all results at a LM weight of 14, which is a reasonable value for our configuration, but not optimized for a particular LM.

As Figures 4.1, 4.2 and 4.3 show, we find in all cases, and for both the *law* and *sciences* test sets, that optimal performance is achieved at an interpolation value somewhere between the extremes, thus showing that LM interpolation is indeed beneficial in all cases. Following the same trend as the LM PPL results, the in-domain LMs perform best on both test sets; these differences in WERs are quite large, confirming the importance of language modeling for this task.

Note that our approach to interpolation requires a fixed vocabulary for all settings of the interpolation weight. Therefore, all words from both training sets are included in all interpolated models, albeit with only the unigram back-off probabilities in some cases. In Figures 4.1, 4.2 and 4.3 we can see that even these unigrams make a useful contribution to

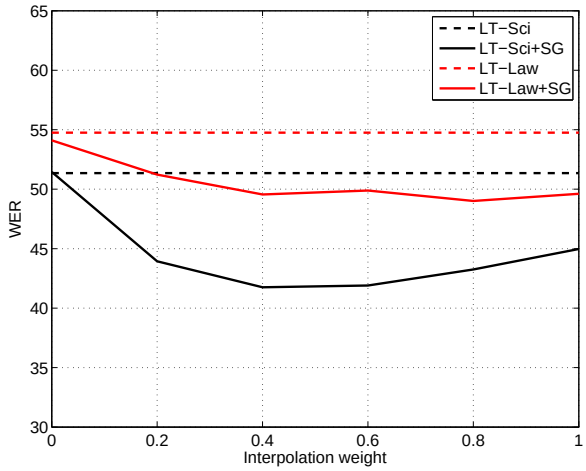

Figure 4.1: WER for off-line lecture transcription when trained on *sci* and *law* sources respectively and evaluated on the combined *sci* and *law* LT test set. The dotted lines correspond to LMs trained only on the LT training data, and the solid lines represent interpolated results, with the interpolation weight on the horizontal axis.

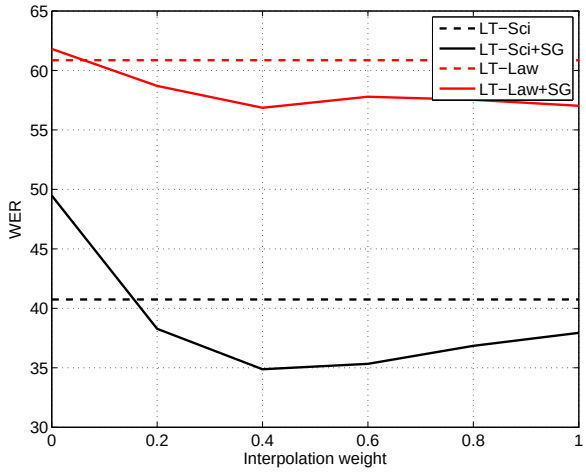

Figure 4.2: WER for off-line lecture transcription when trained on *sci* and *law* sources respectively and evaluated on the *sci* LT test set. The dotted lines correspond to LMs trained only on the LT training data, and the solid lines represent interpolated results, with the interpolation weight on the horizontal axis.

recognition accuracy, since the WERs with only the lecture-transcription LMs (dotted lines in Figs. 4.1, 4.2 and 4.3) are notably higher than the corresponding interpolated models (right-most points of the solid lines).

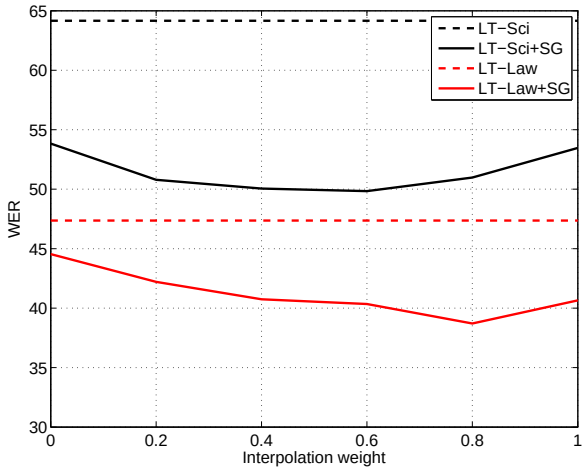

Figure 4.3: WER for off-line lecture transcription when trained on *sci* and *law* sources respectively and evaluated on the *law* LT test set. The dotted lines correspond to LMs trained only on the LT training data, and the solid lines represent interpolated results, with the interpolation weight on the horizontal axis.

4.2 LANGUAGE MODELS FOR ENGLISH LT EXPERIMENTS

Four corpora were used train "Operating systems" subject-specific LMs. These corpora include the "*Study guide*" corpus, the "*OS Books*" corpus, the "*Youtube*" corpus, and the "*Lecturer*" corpus (described in Section 3.1). (The LMs developed in this chapter will be integrated with the experiments in Section 6.2.)

For each LM, the corresponding vocabulary was created by selecting all words present in these four corpora with a word frequency higher than 2. This removed any rarely used (or misspelled), low-frequency words.

The LMs were created by following a similar process as described in Lööf et al. (2009:89) and van Heerden et al. (2011:141). Here, a LM was trained for each individual corpus using Kneser-Ney smoothing, calculating the best interpolation weight contribution for each corpus on a held out dev set. For this training procedure, relevant scripts used by Mr. Petri Jooste (van Heerden et al., 2011:141) were obtained and edited.

For the first fold of cross-validation, separate LMs were trained and evaluated using our English OS corpus dev set (T5) in order to determine corpus-specific performance. These results, together with the results of the interpolated model, are listed in Table 4.1. Here the amount of words contained by each LM, amount of unigrams and trigrams, as well as the perplexity (PPL) and % out-of-vocabulary (OOV) values are listed.

As seen in Table 4.1, the Lecturer corpus provided the lowest perplexity values, as was expected. This is due to the domain-specific and speaker-specific data present in this corpus.

Table 4.1: LM results found for fold 1 of cross-validation. Shows results of independent LMs as well as the interpolated model.

Corpus	#Words	Unigrams	#3-grams	PPL	%OOV
Lecturer	17955	1542	1675	169.48	15.1
Youtube	277535	8875	27569	239.76	4.85
Studyguide	157608	5988	18777	445.44	9.07
OS Books	1002827	20810	106702	259.06	3.52
Interpolated	-	11483	1675	453.12	1.28

Similar results were reported by de Villiers et al. (2012:141), however, contrary to what we observed, the study guide corpus performed much worse.

We created a total of 6 interpolated LMs (one for each fold when performing cross-validation) in the same manner as described above. The results of these interpolated LMs, tested on both the development (DEV) set, as well as the evaluation (EVAL) set are shown in Table 4.2.

Table 4.2: Interpolated LM results on development and evaluation sets. 6 LMs were created, one for each fold of cross-validation.

ID	Uni-grams	#3-grams	DEV PPL	#DEV words	DEV %OOV	EVAL PPL	#EVAL words	EVAL %OOV
Fold 1	11381	143064	174.82	6639	1.37	189.53	6766	1.37
Fold 2	11380	143128	169.10	6766	1.33	137.82	5018	1.20
Fold 3	11383	143354	132.88	5018	1.14	113.75	2446	1.51
Fold 4	11387	143413	119.99	2446	1.47	138.95	2999	1.00
Fold 5	11388	143201	136.73	2999	1.13	145.80	4358	0.99
Fold 6	11386	143044	138.88	4358	0.83	159.99	6639	1.36

Lastly we also created a Julius binary format LM, created in the same manner as described in Lee (2009), using the same optimal weight contributions calculated for each separate LM. For the remaining experiments reported in Section 6.2, only the optimal interpolated LMs were used.

CHAPTER FIVE

ACOUSTIC MODELING: SUPERVISED

Contents

State-of-the-art ASR systems used in the developed world are typically trained on hundreds or even thousands of hours of speech (Novotney & Callison-Burch, 2010:207; Kawahara, 2010:3). This amount of data is rarely, if ever, available in the developing world. For supervised training, well-transcribed audio data is typically required to train ASR systems. In this chapter, we will look at novel ways to obtain or create such corpora (from approximately transcribed corpora), and compare this to models trained on large, well transcribed corpora.

We will present 2 different studies performed over the course of this research (conducted in collaboration with Mr. Petri Jooste, Dr. Charl J. van Heerden, Prof. Etienne Barnard and Prof. Marelie H. Davel) (van Heerden et al., 2011; de Villiers et al., 2012). In the first study where only a limited set of approximately transcribed LT data was available, we investigated different approaches to alignment. The goal was to harvest as much well transcribed data as possible from these approximately transcribed lectures in order to train acoustic models. In the second study where a larger corpus of well-transcribed LT data was available, we investigated lecture (or speaker) adaptation on a per-lecturer basis. These two studies are described below.

5.1 APPROXIMATELY TRANSCRIBED LT DATA

In resource-scarce environments, one may typically expect to find audio recordings of variable quality and (potentially inaccurate) transcriptions of some of those recordings. For this reason, we wanted to investigate ways to employ speech-processing tools in order to utilize resources found in resource-scarce environments.

Towards achieving this goal, we investigated the alignment of the lectures under two conditions: one where we have a well-trained Afrikaans acoustic model available, and another where we try to align the corpus using a well-trained American English acoustic model. This represents the two possible extremes we expect to encounter in resource-scarce environments. On the one hand we have no target-language acoustic data available and have to resort to using an acoustic model trained on a close related language. On the other hand, we have a close to ideal scenario where a large target-language speech corpus is available. (The ideal scenario would of course be to have a large target-language, *domain & speaker specific* corpus available.)

5.1.1 CORPUS PREPARATION

Two corpora were prepared for this experiment namely the *NCHLT* corpus (discussed in Section 3.2.5), and the WSJ corpus (described in Section 3.2.7).

1) **NCHLT models**: For the first approach, we trained a model using the *NCHLT* corpus.

The acoustic model was trained on 39 dimensional Mel frequency cepstral coefficients (13 static with cepstral mean normalization, 13 deltas and 13 double deltas). The hidden Markov models (trained with HTK (Young et al., 2009)) were standard 3-state left to right tied-state triphone models, with 7 mixtures per state and semi tied transforms. The tied states were created using decision tree clustering. A 14 mixture garbage model was then trained

on the entire corpus and combined with this initial model.

This model was then used to perform initial alignment, inserting optional garbage markers between words to absorb disfluencies as well as inaccurately transcribed and untranscribed portions.

At this stage, we had initial alignments, with potentially untranscribed or poorly transcribed sections marked by the garbage model. The next step entailed salvaging as much of the good quality alignments as possible for further retraining. This was accomplished by following an approach described in (Davel et al., 2011), which is based on a dynamic-programming (DP) phone string alignment procedure. It compares the result of a forced alignment with that of a free decode, using a variable cost matrix, and subsequently identifies accurately transcribed sections of audio and corresponding text. These accurate portions were then in turn used to adapt the NCHLT model on a per-lecture basis, using maximum a posteriori (MAP) adaptation, followed by another round of alignment. MAP adaptation was only performed where a lecture was at least 15 minutes in duration (and in those cases we adapted on approximately half of the available speech, retaining the other half for evaluation).

2) **WSJ models**: In resource-scarce environments, large corpora such as the *NCHLT* corpus are generally not available. While we know that a model trained on data from the target language is likely to produce better alignments, it is interesting to determine how closely one can approximate the language-specific results when using a well-trained model from a different language.

An American English acoustic model was thus trained using the WSJ corpus and the CMU pronunciation dictionary, with phone mappings to Afrikaans (described in Section 3.2.7).

This model was then used to align the lectures (again inserting a garbage model between words), followed by the DP alignment procedure described above and corpus segmentation. MAP adaptation was then performed on a per-lecture basis (where enough data was available), and globally for use with those lectures with less than 15 minutes of audio - the same training and test segments as for the *NCHLT* corpus were employed. The process of alignment and corpus segmentation was repeated using the MAP adapted model. The segmented corpus resulting from this second iteration was then used to train a new Afrikaans model from scratch, using the original Afrikaans phone set. These models were again MAP adapted on a per lecture basis.

5.1.2 ACOUSTIC MODELING

From an acoustic modeling perspective, two different approaches were followed; a well trained Afrikaans model (corpus described in Section 3.2.5) was used to align Afrikaans lectures, and a bootstrapping approach was followed whereby a well trained American English model (corpus described in Section 3.2.7) was used for alignment of the same set of lectures. The reason for the two different approaches is to firstly test the feasibility of lecture transcription in resource-scarce environments where no data in the target language is available, and secondly to determine how detrimental the lack of target-language acoustic models is at various stages in the processing chain. In order to quantify the quality of our bootstrapped model, as well as alignment accuracy, a time aligned, accurate evaluation set is required. Since the audio in our corpus is only accompanied by approximate transcriptions, word or phone recognition - which would have been the standard way of evaluating our bootstrapped model - is infeasible. The transcriptions are also not time aligned, complicating the evaluation of our alignment accuracy. Davel et al. (2011) used 3 measures to quantify corpus and model improvement while processing Internet harvested corpora accompanied by approximate transcriptions:

(a) amount of audio absorbed by the garbage model,

(b) average frame log likelihood and

(c) average DP score.

These measures were found to correlate well with each other, as well as with phoneme accuracy as estimated on a carefully transcribed subset of data. We adopted these same measures to quantify performance on spoken lecture alignment.

Another measure, duration-independent overlap rate (Paulo & Oliveira, 2004) was employed to evaluate alignment accuracy. For this measure, 100 randomly selected word instances were manually time aligned across the corpus (50 within the segments that were used for MAP adaptation, and 50 in the segments that were used for evaluation only).

Eight systems were evaluated. The following notation is employed in Tables 5.1 and 5.2:

- **WSJ** refers to the WSJ model where phone mappings, as described in Table 3.5 were employed. A garbage model was also used in conjunction with this model.
- **WSJ no gm** is the same as the model mentioned above, except that no garbage model was trained. This is the only instance of the WSJ process where no garbage modeling was used.

- **WSJ + MAP/spk** refers to speaker adaptation using the training subset from the longer lectures, on a per lecture basis. A pooled model was also created by performing MAP adaptation using all training data (including the shorter lectures). Whenever a lecture had its own MAP adapted model, that model was then used to align and segment the lecture, using techniques described in Davel et al. (2011). If no such model exists, the pooled model was used for segmentation.
- The segmented "corpus" of training data was then used to retrain an Afrikaans spoken lecture (**ASL**) model from scratch. Since the segmented corpus contains Afrikaans lecture transcription data, the original Afrikaans phoneset was used. This model was again used to segment the corpus.
- **ASL + MAP/spk** refers to another round of speaker adaptation using the ASL model.
- For the assumption of the existence of a target-language acoustic model, we used the Afrikaans NCHLT corpus (**NCHLT**), together with a garbage model.
- **NCHLT no gm** is again the only instance where no garbage modeling was used.
- Speaker adaptation was again used to adapt this model on a per lecture basis (**NCHLT + MAP/spk**).

5.1.3 ALIGNMENT ACCURACY

Duration independent overlap was measured across two sets of manually time aligned words. The first set of 50 words (first column in Table 5.1) was selected from those lectures which are long enough to allow for a sizable amount of speech for MAP adaptation (after some held-out sections have been removed, typically the last half of the lecture). Another set of 50 words was selected from the remaining, much shorter lectures, only in models where these shorter lectures were not used as training data. DP scores, average frame log likelihood and percentage of audio absorbed by the garbage model were only measured on the held-out subset from the longer lectures.

From Tables 5.1 and 5.2, it is clear that having a target-language acoustic model available is the best case scenario. However, it is very encouraging to see that a bootstrapping approach can get very similar performance, at least as far as alignment accuracy is concerned.

Table 5.1: Duration independent overlap rate when using different models for alignment.

model	50 words	100 words
WSJ no gm	80.54	73.63
WSJ	66.94	61.62
WSJ + MAP/spk	77.79	-
(WSJ) LT	85.80	-
(WSJ) LT + MAP/spk	86.64	-
NCHLT no gm	83.65	76.84
NCHLT	88.53	82.76
NCHLT + MAP/spk	93.37	-

Table 5.2: Improvements observed during model refinement and alignment, reported on the evaluation set.

model	log P	non-speech (%)	Avg DPS
WSJ no gm	-91.79	36.37	-0.270
WSJ	-88.46	58.60	-0.130
WSJ + MAP/spk	-84.56	45.74	-0.063
ASL	-77.46	42.17	0.163
ASL + MAP/spk	-76.77	41.67	0.217
NCHLT no gm	-84.28	27.68	-0.076
NCHLT	-84.05	46.29	-0.016
NCHLT + MAP/spk	-81.42	42.53	0.236

5.1.4 THE EFFECT OF SPEAKER ADAPTATION

As expected, speaker adaptation seems to be beneficial to the process of alignment. This is most obvious from the difference between the system trained on segmented ASL data, compared to the ASL system which was MAP adapted to particular speakers.

5.1.5 THE EFFECT OF GARBAGE MODELING

The garbage model we employed was very effective when used with the *NCHLT* model, as can be seen from Table 5.2. All measurements (except the percentage of non-speech audio, which is not easy to interpret without knowledge of how much real speech is present) agree that garbage modeling is very beneficial to the process. On the *WSJ* model, the picture looks quite different, though. From the amount of non-speech, it seems that our garbage model is too greedy. The exact reason why it misbehaves with *WSJ* as opposed to *NCHLT*, is an interesting and important research question that needs to be answered for this technique to be completely robust. Our hypothesis is that language differences (between the model being employed and the audio being aligned) can lead to predictable failures of the garbage-model

approach. In particular, if the triphone models do not model the target language well, the more general garbage model (which has a large variance) becomes a better match than the closest matching phone. One easy remedy may be to use fewer mixtures with the triphone models, or to explicitly penalize the garbage model based on language distance measures, such as the Levenshtein distance.

The audio absorbed by the garbage model in Table 5.2 follows a different trend than that found by Davel et al. (2011:3155), where audio absorbed was found to be inversely correlated with the other measures. A negative correlation coefficient of -0.98213 was found by Davel et al. (2011:3155), while a value of 0.01696 was found by the data presented in Table 5.2. However, by ignoring the models containing no garbage models, a correlation coefficient of -0.8139 is found, similar to that found by Davel et al. (2011:3155). This indicates that the models "WSJ no gm" and "NCHLT no gm" trained without garbage models have difficulty identifying "garbage" in the speech data, therefore having lower non-speech values.

5.2 WELL-TRANSCRIBED LT DATA

The collection of the much larger (about 20 hours) *ALT* corpus (described in Section 3.2.1) enabled us to evaluate the approaches described in Section 5.1 on a larger and more diverse set of lectures.

In this experiment, we again made use of two corpora namely the *ANCHLT* corpus (described in Section 3.2.2) and the *ALT* corpus (described in Section 3.2.1).

The acoustic models for these corpora were trained on 39 dimensional Mel frequency cepstral coefficients (13 static, 13 deltas and 13 double deltas). Off-line cepstral mean and variance normalization was applied per speaker (that is, the same normalization constants were applied to all the speech from one speaker, and these constants were computed so that all speakers have the same cepstral means and variances after normalization). The hidden Markov models (HMMs), trained with HTK (Young et al., 2009), were standard 3-state left to right tied-state triphone models, with 8 mixtures per state and semi-tied transforms. A garbage model (Davel et al., 2011) was then trained and combined with the initial model.

The following acoustic models were trained:

- **ANCHLT baseline**. An acoustic model was trained on the *ANCHLT* corpus described in Section 3.2.2. This model was trained without a garbage model.
- **ALT (5-minute segments)**. We trained acoustic models using the entire manually segmented *ALT* corpus. Segments were approximately 5 minutes in duration. This

acoustic model resulted in a phone accuracy of 45.14%. Based on our earlier experience with this corpus, this was an acceptable accuracy for a baseline system; we nevertheless decided to make use of DP scoring to automatically further segment the ALT data into smaller – but more reliable – segments, that could be used for further training.

- **DP filtered ALT**. The *ALT* corpus was automatically segmented into ~10 second or smaller chunks, similar to the process followed by Glass et al.(2007). We employed the same dynamic-programming phone string alignment procedure described in Section 5.1.1 with a flat phone matrix. By using this technique, we segmented the 5 minute ALT data into small chunks of accurately transcribed data using the ANCHLT model. These well-aligned portions were then used to train another ALT model.
- **ANCHLT MAP**. The *ANCHLT* model was then also MAP adapted to all training speakers and used to automatically segment the ALT data using the dynamic-programming technique.

5.2.1 ALIGNMENT ACCURACY

The phone accuracies of these four baseline systems, tested on the same ALT data are shown in Table 5.3. Here, the reference phone strings were generated using the pronunciation dictionary since it was infeasible to obtain manual phone transcripts. The improvements in various measures of alignment accuracy (see (Davel et al., 2011) for motivations) after model refinement are shown in Table 5.4.

Table 5.3: Phone-recognition accuracies of baseline systems tested on ALT

ANCHLT LT(5 min)	ANCHLT	LT(DP scoring)	ANCHLT(MAP all)
19.28%	45.14%	49.70%	16.50%

Table 5.4: Measures of alignment accuracy achieved after model refinement on test set. Here the total hours and minutes extracted from the total duration is also shown

Model	Avg DP Score	Log P	Time extracted
ANCHLT	-0.176	-52.10	4:05/5:39
ANCHLT-MAP/all (ALT)	-0.217	-50.82	3:53/5:39
ANCHLT MAP/spk	-0.202	-51.03	3:35/5:39
ALT	0.114	-45.60	3:14/5:39

5.2.2 SPEAKER ADAPTATION

Speaker adaptation was performed on multiple speakers for which we had data from more than one lecture. One or more lectures were used for speaker adaptation (Train column, Table 3.1), and one lecture was held out for testing purposes (Test column, Table 3.1). The *ANCHLT* corpus acoustic model was also adapted to these speakers to see how important the use of speaker-specific data is to the overall system used for alignment. Table 5.5 summarizes the phone accuracies for different speakers on *ALT* without MAP adaptation, *ALT* with MAP adaptation, *ANCHLT* without MAP adaptation and the *ANCHLT* model with MAP adaptation. These results were obtained using the same techniques as in Trancoso et al. (2006:283) where 3 iterations of speaker adaptation is performed using the same adaptation data. We see that some speakers achieve only small gains in phone accuracy, and reduced accuracies after adaptation are even seen in many cases. These disappointing results are probably a consequence of the small amount of adaptation data available to us thus, the risk of overtraining is significant, and MAP adaptation is not able to compensate for the differences in recording conditions between the training and test lectures.

Table 5.5: Phone accuracies (%) achieved by performing MAP adaptation per lecturer on different models

SPKR ID	ALT	ALT + MAP	ANCHLT	ANCHLT + MAP
m002	59.69	62.12	27.48	29.01
m003	66.82	67.42	33.95	38.03
m006	48.84	49.59	12.38	12.20
m009	50.73	52.21	19.48	18.66
m011	59.39	61.92	19.97	18.72
m012	55.29	49.02	18.60	15.94
f001	39.24	39.43	16.99	15.45
f002	39.91	39.04	13.53	13.77
f004	40.93	34.97	17.84	15.68
f005	28.14	28.14	12.29	8.50

In this chapter, we found target language acoustic models to be beneficial for use in the development of LT systems. Making use of domain-specific training data makes an even larger contribution to the accuracy of LT systems, even in cases where only a limited amount of data is available. Speaker adaptation (MAP) was unexpectedly unsuccessful for about half of our speakers. There may be several reasons, among others, the significant differences in acoustic conditions during the lectures. In future work we hope to determine the exact cause of the degradation in accuracy for these speakers, as well as apply other speaker adaptation techniques, such as Constrained Maximum Likelihood Linear Regression (CMLLR).

Given the results found using supervised training methods, it would be interesting to determine how unsupervised training methods can be used for the development of LT systems.

CHAPTER SIX

ACOUSTIC MODELING: UNSUPERVISED

Contents

Unsupervised training entails machine learning on data with no labels. When applied to ASR, it means training on audio with no transcriptions. A baseline ASR system is then used to automatically "transcribe" the untranscribed data. The baseline system can be either one that is ported from another language (for example a state-of-the art English decoder with phone mappings to recognize Afrikaans), or a decoder trained on a small set of target-language data. After recognition, well recognized segments are extracted using some form of confidence scoring and then used for further retraining of the system. This approach can also be implemented as a bootstrapping process, allowing a rapid increase in ASR accuracy as more data is extracted.

For this reason, unsupervised training is understood to be potentially very rewarding when applied in resource-scarce environments, where only a limited amount of transcribed target-language training data is typically available.

In this chapter, we compare two different approaches to creating an initial acoustic model for unsupervised training. We investigate how beneficial a large target-language corpus is for this purpose, as well as using a small amount of transcribed, target-language (English [1]), *application and speaker specific* data. It would be interesting to compare the results of the two initial approaches, and to establish how important a large target-language corpus really is. Towards this end, 3 different experiments are conducted where the initial decoders are trained on different data sets, ranging from small to large.

The initial systems were trained on the following corpus selections:

1. A large topic-independent corpus with little to no domain specific data, and no speaker-specific data.

2. A large topic-independent corpus, together with a small set of transcribed domain and speaker-specific data.

3. Only a small set of transcribed domain and speaker-specific data.

We will refer to these experiments as ENCHLT, ENCHLT + OS(11 min) and OS(11 min) respectively, with the experiment name also indicating which corpora were used (see Section 3.2.4 and Section 3.2.6 for a description of the *ENCHLT* and *OS* corpora respectively). Before we commence with a detailed description of these experiments, we first report on work performed to determine which decoder to use, and how to optimize the decoder parameters.

6.1 COMPARING AND OPTIMIZING A DECODER

Before investigating the method of unsupervised training, we performed several experiments to identify the best available decoder. Using the best available decoder will result in more accurate transcriptions, which can then be utilized to harvest as much valuable data as possible. We will then have a look at the process used to estimate confidence threshold scores, which are necessary to identify and extract well recognized portions of data.

[1] Even though English is not a resource-scarce language, the OS corpus was considered suitable for this study as (1) the Afrikaans accented English used is significantly different from US or UK English (for which resources are abundant), and can thus be considered resource-scarce from an acoustic point of view (2) the specific lecturer was very willing to assist with this project in every way possible, which simplified the data collection process considerably, and (3) the techniques applied to this corpus are the same as that applied to the ALT corpus; the results hence corroborate the trends observed on the ALT corpus.

6.1.1 IDENTIFYING THE BEST DECODER

Two decoders were considered and evaluated for use with unsupervised training; *HDecode* and *Julius*. Each of these decoders was tested by evaluating the first-fold development set (T5), using the initial *ENCHLT* system (which we will discuss in Section 6.2.1). The evaluation entailed a grid search over language model weights (LMW) and insertion penalty values (INSP). The optimal decoder was then selected as the one obtaining the lowest possible WER. The WERs of the two decoders, *HDecode* and *Julius*, are shown in Table 6.1 and Table 6.2 respectively.

From the 431 audio segments used for decoding, the *Julius* decoder returned a number of failed audio segments, which it was unable to decode completely (shown in Table 6.3). *HDecode* also displayed some form of failed searches; however, *HDecode* has an option to return partial output, without discarding the complete file.

Table 6.1: %WER for different values of LMW and INSP when decoding with HDecode.

LMW/INSP	-10.0	-8.0	-6.0	-4.0	-2.0
8.0	54.29	54.54	54.92	55.19	55.73
10.0	51.39	50.99	50.76	50.64	50.73
12.0	50.19	49.92	49.44	49.71	49.63
14.0	51.29	50.76	50.43	50.34	50.08

Table 6.2: %WER for different values of LMW and INSP when decoding with Julius.

LMW/INSP	-12.0	-10.0	-8.0	-6.0	-4.0	-2.0
8.0	75.81	76.49	76.83	77.72	78.28	78.60
10.0	73.39	73.69	74.44	74.34	74.51	75.10
12.0	72.02	71.72	72.13	72.20	72.70	73.15
14.0	72.82	72.75	72.72	72.53	72.32	72.78
16.0	74.44	74.82	74.15	74.60	73.74	75.12
18.0	76.04	76.14	77.02	77.79	77.59	77.41

From the results shown in Table 6.1 and Table 6.2, it is clear that *HDecode* achieved the lowest WER (49.44%) on this data set. We therefore use *HDecode* for the remainder of the experiments.

Julius was selected as the decoder for the NWU LT platform (described in chapter 7) for several reasons, among others because it is free and open-source, as well as having received favourable reviews by some researchers. The higher WERs obtained when using *Julius* was thus disappointing, especially since it was found to be comparable to HDecode in several

Table 6.3: Total unsuccessful decodes for different values of LMW and INSP, using Julius.

LMW/INSP	-12.0	-10.0	-8.0	-6.0	-4.0	-2.0
8.0	0	0	0	0	1	1
10.0	2	1	1	1	1	0
12.0	10	8	7	3	3	3
14.0	17	18	15	12	12	8
16.0	23	21	21	21	22	20
18.0	61	57	49	33	36	38

other studies (Silva et al., 2010:131). One possible explanation may be that some parameter settings in Julius are more sensitive to the big mismatch between the *ENCHLT* and *OS* corpora. Another explanation may be that our specific model building process is more favourable for HDecode.

6.1.2 CONFIDENCE SCORE ESTIMATION

When decoding untranscribed data during unsupervised training with a less than optimal ASR system, many recognized words will, in fact, be recognition errors. In this section, we describe the confidence scores we used to select those words that are likely to be correct, for use during retraining.

During our experiments we make use of word lattices which are generated during the recognition phase. These word lattices show all the possible word sequences that were not pruned. From the lattices, word posteriors can be computed, which is considered to be a state-of-the-art confidence measure. An appropriate confidence measure threshold also has to be determined for use during unsupervised training.

According to Kemp & Waibel (1999:2727), when the threshold value is chosen too high, only well understood words will be retrieved. In this case, the decoder will only learn what it already knows. When the threshold value is set too low, poorly modeled words will be retrieved, adding more erroneous data to the training set and lowering recognition performance. The confidence score threshold should ideally be chosen such that most correct words are retrieved while the least correct are discarded.

To illustrate this, the reader is referred to Fig. 6.1. For this illustration our aim was to determine the best confidence value, using the initial *ENCHLT* model (discussed in Section 6.2.1). All thresholds from 0.0 to 1.0 with increments of 0.05 were tested on the dev set (T5). From these results, we determined the amount of correct/incorrect words identified by each threshold. This amount of correct/incorrect words are shown Figure 6.1.

The optimal threshold can be seen to be ~0.5 where the difference between the amount

of correct words and the amount of incorrect words is largest. This result is very similar to the 0.5 found by Kemp & Waibel (1999:2727).

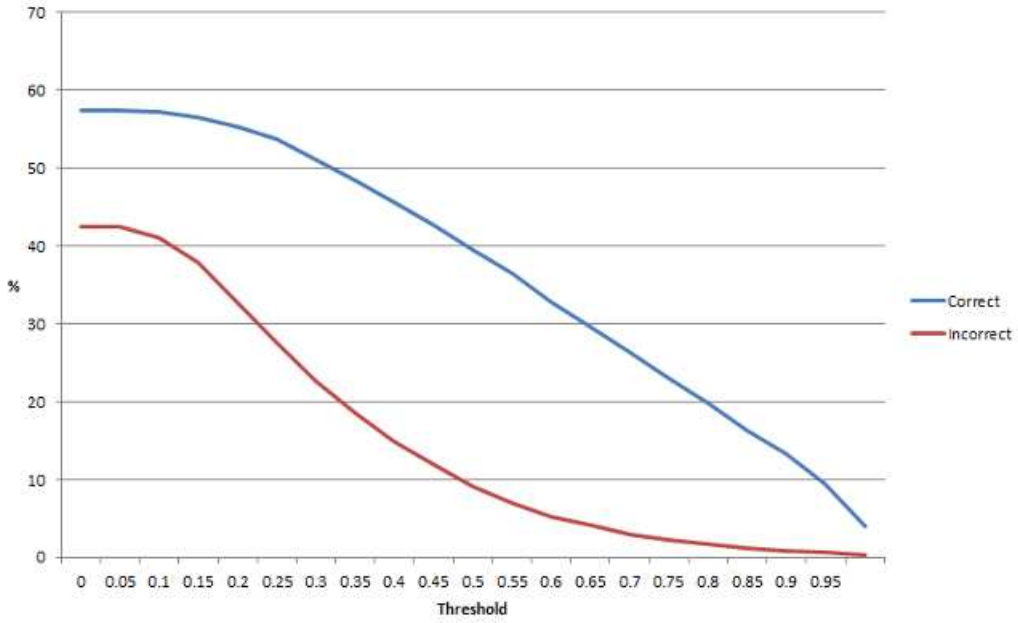

Figure 6.1: Accuracies achieved on different word confidence thresholds

Note this is simply an illustration of the process we use to determine the optimal thresholds. This process will be repeated for each iteration during the unsupervised training process (described in Section 6.2).

6.2 UNSUPERVISED TRAINING

In this section, 3 initial acoustic models will be trained (ENCHLT, ENCHLT + OS(11 min) and OS(11 min)), based on an iterative approach. A total of 7 iterations are performed for each experiment below. Each iteration also entails a confidence threshold re-estimation step where the best confidence value was again determined on the dev set (as done in Section 6.1.2). These recursive training steps can be described as follows:

1. Decode any untranscribed OS data.
2. Extract word based lattices.
3. Determine best confidence threshold on dev set.
4. Segment original MFCC files based on word confidences.

5. Train new models or MAP adapt the original ENCHLT models using this new data.

6. Repeat 1–5

Each system was trained on 39 dimensional Mel frequency cepstral coefficients (13 static, 13 deltas and 13 double deltas). Offline cepstral mean and variance normalization was applied per lecture (that is, the same normalization constants were applied to all the speech from a single class). The hidden Markov models (HMMs) were trained with HTK (Young et al., 2009) as done by de Villiers et al. (2012:140) and were standard 3-state left-to-right tied state triphone models containing 8 mixtures per state.

6.2.1 ENCHLT

It would be interesting to establish what accuracies are achievable in the case where large, target-language speech corpora are indeed available.

For this experiment, we focused on the accuracies achievable when training the initial acoustic models on a large, target-language, non domain-specific corpus (*ENCHLT*). Each iteration was decoded using the *HDecode* decoder, set with a language model weight of 10.0 and an insertion penalty value of -8.0. After each iteration, MAP adaptation was performed on the original ENCHLT models, using the newly recognized OS data.

Table 6.4 displays the results from the first fold of cross-validation, after being evaluated on its dev set (T5). Here, the iteration number (where 0 represents the initial system), WER, amount of data extracted (from the untranscribed OS data), as well as the optimal confidence threshold (Thres.) for each iteration is shown.

Table 6.4: Fold 1 Iterative Unsupervised training results using ENCHLT model

Iter	Dev WER	Eval WER	#Words	Time Extracted/Total Time	Thres.
0	50.99	54.37	13181	1:32/3:03	0.45
1	38.05	41.40	19850	2:12/3:03	0.55
2	35.47	37.05	21415	2:22/3:03	0.55
3	34.39	35.90	21930	2:25/3:03	0.55
4	33.45	35.44	22170	2:27/3:03	0.55
5	33.03	35.35	22275	2:27/3:03	0.55
6	32.72	35.28	22399	2:28/3:03	0.55
7	32.66	35.31	-	-/3:03	-

Table 6.5 shows the average WERs reported on both the dev and eval set, calculated across all 6 folds of cross-validation.

Table 6.5: Average WERs achieved across all 6 folds cross-validation, for all 7 iterations of iterative unsupervised training using ENCHLT model

Iter	Dev	Eval
0	54.28	54.13
1	41.06	41.13
2	38.36	38.02
3	37.05	37.05
4	36.58	36.44
5	36.23	35.93
6	35.94	35.88
7	35.51	35.97

6.2.2 ENCHLT + OS(11MIN)

When a large target-language speech corpus, as well as a small set of transcribed domain-specific, speaker-specific data is available, the smaller data set can be utilized to perform acoustic model adaptation (such as MAP adaptation) on the larger corpus.

For this experiment, the *ENCHLT* corpus was used, together with a small transcribed dataset, extracted from the *OS* corpus (U6/T0). This smaller data set consisted of a 22 minute class (11 minutes after audio segmentation and silence removal). The initial system was created by training the ENCHLT data and performing MAP adaptation (with 3 iterations of adaptation as done in Trancoso et al. (2006:283)) on the corresponding models, using the 11 minutes of transcribed OS data. Each iteration was decoded with *HDecode*, using a language model weight of 10.0 and an insertion penalty of -8.0. After each iteration of training, MAP adaptation was performed on the initial ENCHLT models using the newly recognized OS data, as well as the 11 minutes of transcribed OS data.

The pronunciation dictionary was composed of all words from the *ENCHLT* corpus, the LM as well as the 11 minute transcribed data set. Pronunciations for unknown words were predicted by means of the Default & Refine algorithm (Davel & Barnard, 2008), which were then also manually checked.

Table 6.6 displays the results from the first fold of cross-validation, after being evaluated on its dev set (T5). Here, the iteration number (where 0 represents the initial system), WER, amount of data extracted (from the untranscribed OS data), as well as the optimal confidence threshold (Thres.) for each iteration is shown.

Table 6.7 shows the average WERs reported on both the dev and eval set, calculated across all 6 folds of cross-validation.

Table 6.6: Fold 1 Iterative Unsupervised training results using ENCHLT + OS(11 min) model

Iter	Dev WER	Eval WER	#Words	Time Extracted/Total Time	Thres.
0	39.70	40.23	17974	1:56/2:51	0.45
1	33.85	34.97	20033	2:10/2:51	0.55
2	32.70	34.91	20658	2:16/2:51	0.55
3	32.57	34.10	21736	2:21/2:51	0.50
4	32.13	34.29	21235	2:19/2:51	0.55
5	31.90	34.30	21993	2:23/2:51	0.50
6	31.66	33.51	21420	2:21/2:51	0.55
7	31.44	34.13	-	-/2:51	-

Table 6.7: Average WERs achieved across all 6 folds cross-validation, for all 7 iterations of iterative unsupervised training using ENCHLT + OS(11 min) model

Iter	Dev	Eval
0	42.30	42.06
1	36.38	36.25
2	35.27	34.99
3	34.62	34.27
4	34.48	34.27
5	34.44	34.36
6	34.27	34.15
7	34.37	34.06

6.2.3 OS(11 MIN)

In the absence of large target-language corpora, one may have to resort to manually transcribing small amounts of data. Kemp & Waibel (1999:2728) made use of only as little as 30 minutes of transcribed data to build their baseline acoustic models and achieved accuracies very close to transcription performance by means of unsupervised training (21.4% WER); however, in their case they made use of a large set of untranscribed data (51 hours). In this section, we attempt to replicate this feat, but with much less untranscribed data available.

For this experiment, the initial system was trained on a small transcribed dataset, extracted from the *OS* corpus (U6/T0). This smaller data set consisted of a 22 minute class (11 minutes after audio segmentation and silence removal). Each iteration was decoded using *HDecode*, with a language model weight of 12.0 and an insertion penalty of -4.0. After each iteration, a new system was trained using the initial 11 minutes of transcribed OS data, together with the newly extracted data.

Table 6.8 displays the results from the first fold of cross-validation, after being evaluated

on its dev set (T5). Here, the iteration number (where 0 represents the initial system), WER, amount of data extracted (from the untranscribed OS data), as well as the optimal confidence threshold (Thres.) for each iteration is shown.

Table 6.8: Fold 1 Iterative Unsupervised training results using OS(11 min) model

Iter	Dev WER	Eval WER	#Words	Time Extracted/Total Time	Thres.
0	44.54	43.94	14371	1:34/2:51	0.50
1	34.51	35.47	19131	2:04/2:51	0.55
2	31.72	33.25	20426	2:12/2:51	0.55
3	31.28	31.42	20915	2:16/2:51	0.55
4	31.45	31.61	21153	2:18/2:51	0.55
5	30.91	31.29	22180	2:24/2:51	0.50
6	30.28	30.85	22491	2:25/2:51	0.50
7	30.44	31.10	-	-/2:51	-

Table 6.9 shows the average WERs reported on both the dev and eval set, calculated across all 6 folds of cross-validation.

Table 6.9: Average WERs achieved across all 6 folds cross-validation, for all 7 iterations of iterative unsupervised training using OS(11 min) model

Iter	Dev	Eval
0	47.76	47.79
1	37.42	37.67
2	35.22	34.90
3	34.29	33.98
4	34.18	33.98
5	33.64	33.08
6	33.40	33.24
7	33.71	33.44

From these experiments, an average confidence threshold of 0.65 is observed. This is very similar to 0.7 found useful by Wessel & Ney (2001:310).

From Tables 6.4, Table 6.6 and Table 6.8 the increase in WER is clearly visible, also when considering the amount of data extracted. As expected, better results were found in 'ENCHLT + OS(11min)'; however, very similar results were observed in 'ENCHLT' and 'OS(11min)'. When considering Table 6.5, Table 6.7 and Table 6.9, it becomes clear that similar WERs were achieved across 6 folds cross-validation. These results indicate that even a very small transcribed set of speaker-specific, domain-specific data can be used to achieve similar results than those achieved by using large topic-independent, speaker-independent corpus.

Clearly the method of unsupervised training proved useful when implemented on LT data.

CHAPTER SEVEN

NWU LT SYSTEM

Contents

ASR plays a fundamental role in LT systems: without an ASR capability, LT systems would not be of much use. Nevertheless, after an appropriate ASR system is trained, a usable system is still required to capture and present the transcripts to the learners. In this chapter we will present an overview on the *NWU Lecture Transcription system* we developed for use in classrooms.

The NWU LT system was designed in two parts: the server-side and client-side system. The server-side system handles the communication to and from multiple connected client LT systems, as well as the operation of all active ASR decoders running on the server. The client-side system is to be used within the classrooms. It displays the transcribed text to the students and allows the recording and storing of both video files as well as the generated transcriptions.

This client/server architecture was chosen to allow generalization of the decoder, so it can be accessed and used both locally and remotely. This will create a centralized storage space for the lecture recordings and transcriptions, will not require an expensive processing system in every classroom, and will not require the latest acoustic models to be available on every machine (Bain et al., 2002:195).

After starting the client system (in the classroom), a username and password is requested (we expect the lecturer to start the system). This authentication is mainly to identify the users of the system to prevent them from choosing an incorrect ASR profile. After authentication, the client system queries a remote database to acquire user-stored settings and available profiles. The user can then select a profile based on the subject and language in which they are to lecture. Each profile contains a certain set of acoustic models, language models and other ASR related settings to be used during recognition. This means each lecturer can have many profiles (e.g. one for each subject or language). After the required profile is selected, the ASR service may be started. When started, the client application communicates with the server-side application through an asynchronous network port. All clients are registered on the server by storing relevant data such as their IP (Internet Protocol) address and the selected profile to be used during recognition. After the new client has been registered on the server application, it invokes a new Julius instance specifically for this client which then waits for incoming speech on any available port found. The server then notifies the client that its ASR engine is ready to receive speech. The client then starts the Julius "adintool" (Lee, 2010:54) application which takes input from the selected client microphone and sends it to the corresponding Julius speech recognizer running on the server. The decoded text returned by Julius is read by the server application and sent to the corresponding client.

The overview of the NWU LT system is presented in figure 7.1, followed by more detail on each subsystem.

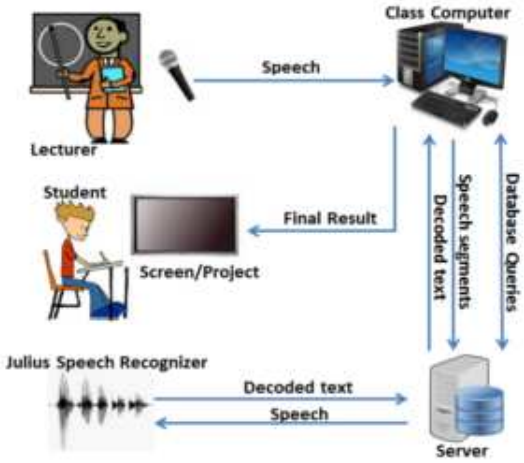

Figure 7.1: NWU LT system overview

7.1 THE SERVER-SIDE SYSTEM

The server-side system is a console application written in the C++ programming language, mainly to enable cross-platform server support.

The server-side system starts a new instance of the Julius ASR engine for each connected client application. When audio segments arrive through a specified port on which the Julius decoder is listening, the audio is decoded, and the decoded text is sent back to the client.

Many of these server applications may be run on a single server, as long as they are configured to communicate through different ports. This might provide the lecturing system with alternatives in case an unrecoverable error has occurred in one of its primary LT server applications.

A sample of the server-side console is shown in figure 7.2

Figure 7.2: NWU LT system server-side view

7.2 THE CLASSROOM SYSTEM

The classroom system was written in the Visual Basic programming language, giving a more aesthetically pleasing environment for the user to work in.

This system has the ability to display and record a live video feed from a connected webcam or even the computer desktop, by using the *Microsoft Expression Encoder 4* api (Anon., 2013*b*). This allows better communication to students in larger classes and makes it possible to distribute recorded lectures with transcriptions.

It also has a built-in function which allows the user to choose and make use of control phrases. This means the user can utter chosen phrases to control basic functions of the system, once the ASR service has been started. Currently available control phrases include the function to start recordings, end recordings and switch the view of the system.

The system has three visual output settings; video only, transcription only or both video and transcription. Samples of the 3 different views of the client system are shown in figures 7.3 to 7.5.

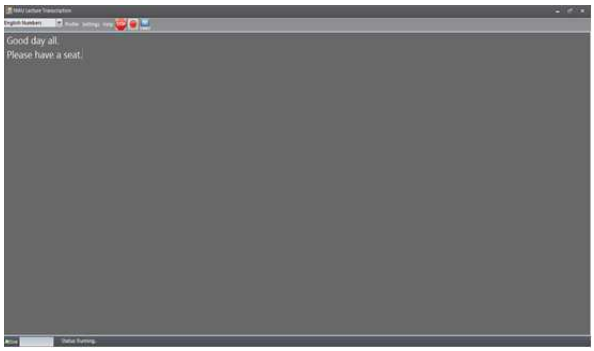

Figure 7.3: NWU LT system transcription view

Figure 7.4: NWU LT system video/transcription view

Figure 7.5: NWU LT system video/transcription view

CHAPTER EIGHT

CONCLUSION

Contents

Lecture transcription(LT) systems in classrooms has long been understood to be a potentially rewarding endeavour. These systems make use of ASR systems in order to provide students with real-time in-class transcriptions or recordings and transcriptions for offline use. These systems will provide clear benefits to students with hearing impairments, visual impairments, physical impairments, learning disabilities or even non-native listeners. Apart from these students experiencing such learning obstacles, LT systems were however also found useful by everyday students, as they were found to better understand the learning material and to productively augment their own class notes.

LT systems have successfully been implemented in a number of learning environments found in the developed world where all necessary resources required were easily obtainable. These ASR system acoustic models are typically trained on hundreds or even thousands of hours of speech (Novotney & Callison-Burch, 2010:207; Kawahara, 2010:3), while their language models are typically trained on text sets consisting of millions of words (Kawahara, 2010:3; Glass et al., 2007:2554). These amounts of resources might typically not be available in the developing world and might provide only a limited set, or low accuracy training data. For this reason, new methods of ASR training should be considered in order to utilize this limited training sources and train usable ASR systems for use in lecturing environments.

In this study, a number of approaches toward the development of usable LT systems in resource-scarce environments have been investigated.

We focus on different approaches to alignment using the Dynamic-programming phone string alignment procedure (Davel et al., 2011), to determine how approximately-transcribed speech data can be utilized in order to harvest as much usable data for the development of acoustic models. We find that target-language acoustic models are optimal for this purpose, but encouraging results may even be found using models from another language.

We also make use of unsupervised training methods where an initial low-accuracy recognizer is used to transcribe a set of untranscribed data. Using this poorly transcribed data, well understood portions are extracted based on a word confidence threshold. The initial system is retrained along with the newly recognized data in order to increase its overall accuracy.

Using this method of unsupervised training we trained our initial recognizer using as little as 11 minutes of transcribed speech. After this method was applied in an iterative fashion, a noticeable increase in accuracy was observed (47.79% WER to 33.44% WER). Similar results were however found (35.97% WER) after using the large speaker-independent corpus to train the initial system.

Usable LMs were also created using as few as 17955 words; however, this resulted in large out-of-vocabulary rates. This problem was solved by means of LM interpolation. LM interpolation was found to be very beneficial in cases where only subject-specific data (such as lecture slides and books) was available. Even though these subject-specific data sources made the largest contribution related to domain-specific words, common words typically found in spontaneous speech were poorly modeled. Thus, lower perplexity values were found after LM interpolation (as opposed to individual LMs) using different types of text corpora. We also introduce our NWU LT system, developed for use in learning environments, designed on a client/server based architecture.

Based on the results found in this study we are confident that usable models for use in LT systems can be developed in resource-scarce environments. As LT systems are incorporated into classrooms, much more training data, whether untranscribed or partially transcribed, will rapidly become obtainable, thus greatly enhancing the potential of this technology.

8.1 FUTURE WORK

The most important matter for future work is the refinement of our approaches to unsupervised training, using initial models from another language. Since this application is likely to be the most common use case in the developing world, the promising LM and AM results

that we have obtained should be extended.

It would also be interesting to determine the effect of larger sets of untranscribed data on recognition accuracy, in order to determine whether it is worthwhile recording large numbers of lectures even if transcription is infeasible.

Further investigations might also experiment with different speech recognition technologies such as Kaldi, which enable state-of-the-art ASR systems to be trained and tested. The benefits of using the improved capabilities of Kaldi (compared to the HTK toolkit used in this work) have been quite variable, and it is important to evaluate their potential in the LT domain.

REFERENCES

Anon. (1998), 'The cmu pronouncing dictionary', http://www.speech.cs.cmu.edu/cgi-bin/cmudict. Accessed: 2013-10-26.

Anon. (2011), 'Data accountability center: individuals with disabilities education act(idea) data', www.IDEAdata.org. Accessed: 2013-10-25.

Anon. (2013*a*), 'The language resource management agency', http://rma.nwu.ac.za/. Accessed: 2013-10-31.

Anon. (2013*b*), 'Microsoft download center: microsoft expression encoder 4', http://www.microsoft.com/en-za/download/details.aspx?id=18974. Accessed: 2013-10-27.

Anon. (2013*c*), 'Sox - sound exchange', http://sox.sourceforge.net/. Accessed: 2013-10-31.

Anusuya, M. & Katti, S. K. (2009), 'Speech recognition by machine, a review', *International Journal of Computer Science and Information Security* **6**(3), 181–205.

Bain, K., Basson, S. H. & Wald, M. (2002), Speech Recognition in University Classrooms : Liberated Learning Project, *in* 'Proceedings of the fifth international ACM conference on Assistive technologies - Assets '02', New York, USA, pp. 192–196.

Barnard, E., Davel, M., van Heeren, C., Kleynhans, N. & Bali, K. (2011), Phone recognition for spoken web search, *in* 'Working Notes Proceedings of the MediaEval 2011 Workshop', Pisa, Italy.

Basson, W. D. & Davel, M. H. (2012), Comparing grapheme-based and phoneme-based speech recognition for afrikaans, *in* 'Proceedings of the 23rd Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)', Pretoria, South Africa, pp. 5–9.

Bisani, M. & Ney, H. (2008), 'Joint-sequence models for grapheme-to-phoneme conversion', *Speech Communication* **50**(5), 434–451.

Chen, S. F. & Goodman, J. (1999), 'An empirical study of smoothing techniques for language modeling', *Computer Speech & Language* **13**(4), 359–393.

Davel, M. & Barnard, E. (2008), 'Pronunciation predication with Default&Refine', *Computer Speech and Language* **22**, 374–393.

Davel, M. H. & De Wet, F. (2010), Verifying pronunciation dictionaries using conflict analysis, *in* 'Proceedings of the 11th Annual Conference of the International Speech Communication Association (Interspeech)', Makuhari, Japan, pp. 1898–1901.

Davel, M. H., Van Heerden, C., Kleynhans, N. & Barnard, E. (2011), Efficient harvesting of internet audio for resource-scarce asr, *in* 'Proceedings of the 12th Annual Conference of the International Speech Communication Association (Interspeech)', Florence, Italy, pp. 3153–3156.

Davel, M. & Martirosian, O. (2009), Pronunciation dictionary development in resource-scarce environments, *in* 'Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech)', Brighton, United Kingdom, pp. 2851–2854.

de Villiers, P., Jooste, P., van Heerden, C. J. & Barnard, E. (2012), Towards lecture transcription in resource-scarce environments, *in* 'Proceedings of the 23rd Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)', Pretoria, South Africa, pp. 138–143.

De Vries, N. J., Badenhorst, J., Davel, M. H., Barnard, E. & De Waal, A. (2011), Woefzela-an open-source platform for asr data collection in the developing world, *in* 'Proceedings of the 12th Annual Conference of the International Speech Communication Association (Interspeech)', Florence, Italy, pp. 3177–3180.

Gales, M. & Young, S. (2008), 'The application of hidden markov models in speech recognition', *Foundations and Trends in Signal Processing* **1**(3), 195–304.

Glass, J. R., Hazen, T. J., Cyphers, D. S., Malioutov, I., Huynh, D. & Barzilay, R. (2007), Recent progress in the mit spoken lecture processing project., *in* 'Proceedings of the 8th Annual Conference of the International Speech Communication Association (Interspeech)', Antwerp, Belgium, pp. 2553–2556.

Hughes, T., Nakajima, K., Ha, L., Vasu, A., Moreno, P. J. & LeBeau, M. (2010), Building transcribed speech corpora quickly and cheaply for many languages., *in* 'Proceedings of the 11th Annual Conference of the International Speech Communication Association (Interspeech)', Makuhari, Japan, pp. 1914–1917.

Jurafsky, D. & Martin, J. H. (2000), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edn, Prentice Hall, Upper Saddle River, New Jersey, USA.

Kawahara, T. (2010), Automatic transcription of parliamentary meetings and classroom lectures-a sustainable approach and real system evaluations, *in* 'Proceedings of the 7th International Symposium on Chinese Spoken Language Processing (ISCSLP)', Tainan, Taiwan, pp. 1–6.

Kawahara, T., Katsumaru, N., Akita, Y. & Mori, S. (2010), Classroom note-taking system for hearing impaired students using automatic speech recognition adapted to lectures., *in* 'Proceedings of the 11th Annual Conference of the International Speech Communication Association (Interspeech)', Makuhari, Japan, pp. 626–629.

Kemp, T. & Waibel, A. (1999), Unsupervised training of a speech recognizer: recent experiments., *in* 'Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech)', Budapest, Hungary, pp. 2725–2728.

Kheir, R. & Way, T. (2007), Inclusion of deaf students in computer science classes using real-time speech transcription, *in* 'Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education', Dundee, Scotland, United Kingdom, pp. 261–265.

Lee, A. (2009), 'Julius: using a srilm n-gram on julius', `http://web.archive.org/web/20080207010024/`. Accessed: 2013-10-23.

Lee, A. (2010), 'The julius book'.

Liao, H., McDermott, E. & Senior, A. (2013), Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription, *in* 'IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)', Olomouc, Czech Republic, pp. 368–373.

Lööf, J., Gollan, C. & Ney, H. (2009), Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a polish speech recognition system., *in* 'Proceed-

ings of the 10th Annual Conference of the International Speech Communication Association (Interspeech)', Brighton, United Kingdom, pp. 88–91.

Meraka-Institute (2009), 'Dictionarymaker', `http://dictionarymaker.sourceforge.net/`. Accessed: 2013-10-31.

Mitten, R. (1992), 'Computer-usable version of oxford advanced learners dictionary of current english. technical report, oxford text archive.', `http://ota.ahds.ac.uk/texts/0154.html`. Accessed: 2013-10-31.

Mohamed, A. R., Dahl, G. E. & Hinton, G. (2012), 'Acoustic modeling using deep belief networks', *IEEE Transactions on Audio, Speech, and Language Processing* **20**(1), 14–22.

Morgan, N. & Bourlard, H. (1995), 'Continuous speech recognition', *IEEE Signal Processing Magazine* **12**(3), 24–42.

Munteanu, C., Penn, G. & Baecker, R. (2007), Web-based language modelling for automatic lecture transcription., *in* 'Proceedings of the 8th Annual Conference of the International Speech Communication Association (Interspeech)', Antwerp, Belgium, pp. 2353–2356.

Nanjo, H. & Kawahara, T. (2003), Unsupervised language model adaptation for lecture speech recognition, *in* 'ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR)', Tokyo, Japan.

Novotney, S. & Callison-Burch, C. (2010), Cheap, fast and good enough: Automatic speech recognition with non-expert transcription, *in* 'Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)', Los Angeles, USA, pp. 207–215.

Park, A., Hazen, T. J. & Glass, J. (2005), Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling, *in* 'Proceedings of the 30th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)', Philadelphia, USA, pp. 497–500.

Paulo, S. & Oliveira, L. (2004), *Advances in Natural Language Processing*, Springer, Heidelberg, Berlin.

Ranchal, R., Taber-Doughty, T., Guo, Y., Bain, K., Martin, H., Robinson, J. P. & Duerstock, B. S. (2013), 'Using speech recognition for real-time captioning and lecture transcription in the classroom', *IEEE Transactions on Learning Technologies* **99**, 1–14.

Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., Kamvar, M. & Strope, B. (2010), "your word is my command": Google search by voice: A case study, *in* 'Advances in Speech Recognition', Springer, pp. 61–90.

Schultz, T. & Waibel, A. (2001), Experiments on cross-language acoustic modeling., *in* 'Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)', Aalborg, Denmark, pp. 2721–2724.

Shriberg, E. (2005), Spontaneous speech: how people really talk and why engineers should care., *in* 'Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech)', Lisbon, Portugal, pp. 1781–1784.

Silva, P., Batista, P., Neto, N. & Klautau, A. (2010), An open-source speech recognizer for brazilian portuguese with a windows programming interface, *in* 'Computational Processing of the Portuguese Language', Brazil, pp. 128–131.

Trancoso, I., Nunes, R., Neves, L., Viana, C., Moniz, H., Caseiro, D. & Mata, A. I. (2006), Recognition of classroom lectures in european portuguese., *in* 'Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech)', Pittsburgh, Pennsylvania, USA, pp. 281–284.

van Heerden, C., Davel, M. H. & Barnard, E. (2013), The semi-automated creation of stratified speech corpora, *in* 'Proceedings of the 24th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)', Johannesbrug, South Africa, pp. 115–119.

van Heerden, C. J., de Villiers, P., Barnard, E. & Davel, M. H. (2011), Processing spoken lectures in resource-scarce environments, *in* 'Proceedings of the 22nd Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)', Vanderbijlpark, South Africa, pp. 138–143.

Wessel, F. & Ney, H. (2001), Unsupervised training of acoustic models for large vocabulary continuous speech recognition, *in* 'IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)', Madonna di Campiglio, Italy, pp. 307–310.

Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. & Woodland, P. C. (2009), *The HTK Book*, University of Cambridge.

APPENDIX A

LIST OF ACRONYMS

ALT	Afrikaans lecture transcription
AM	Acoustic model
ANCHLT	Afrikaans National Centre for Human Language Technologies
ASL	Afrikaans spoken lectures
ASR	automatic speech recognition
CV	cross-validation
CMU	Carnegie Mellon University
CMLLR	Constrained Maximum Likelihood Linear Regression
DNN	Deep neural network
DP	Dynamic programming
ENCHLT	English National Centre for Human Language Technologies
HMM	Hidden Markov model
IBM	International Business Machines
INSP	Insertion penalty
LMW	Language model weight
LT	lecture transcription
MAP	Maximum a posteriori
MIT	Massachusetts Institute of Technology
MLP	Multi-layer perceptron
NCHLT	National Centre for Human Language Technologies
OALD	Oxford Advanced Learner's Dictionary
OOV	Out of vocabulary

OS	Operating systems
PDP	Phone-based dynamic programming
PER	Phone error rate
PPL	Perplexity
UK	United Kingdom
US	United States
VUST	Villanova University Speech Transcriber
WER	Word error rate
WSJ	Wall street journal