Layperson-Trained Speech Recognition for Resource Scarce Languages

Fang Qiao

May, 2010

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Keywords: Speech recognition, developing contries

For mentors and friends who came,
then stayed or went.
To those who gave me strength and wisdom for each step that I take.

Abstract

As speech technology continues to advance, the languages for which they are studied, and for which they have been developed also expands to cover more and more of those used by the first world countries. It is not until the recent years have efforts been made to apply our growing knowledge to languages of the developing world. However, building speech technologies for these languages has revealed new challenges such as the limited economic, human and speech resources, as well as the low-literacy of target population.

In my thesis, I will describe an approach to develop speech recognition for languages of the developing word. The resulting technique will help build speech recognizers that retain the high recognition accuracy of a high-quality commercial speech recognizer over a small vocabulary.

Our approach addresses many issues faced by similar efforts today. It is designed to utilize an existing, well-trained speech recognizer as the baseline to circumvent the reliance on large amount of audio data and human resource for developing acoustic models for a target language. Then, by using cross-language phoneme mapping, we can use the baseline recognizer to build new recognizers for any target language. Furthermore, we design the approach to minimize the need of human expertise through the incorporation of a data-driven approach in generating the pronunciation rules for the target language.

Finally, I will present test results of our technique on both first world and resource-poor languages, and discuss both the potentials of our approach and possible future extensions.

Acknowledgements

It is the third hour since I first attempted to write this section. I purposefully left this for the last because I thought it was going to be the easiest, since I had a pretty clear idea of all the people I was thankful of, from the bottom of my heart, for this wonderful research experience in Carnegie Mellon. But each time I began typing words, I found my self at an utter lost words to express my gratitude to the fullest extend. Hence I have decided to keep my thanks from here on as direct as possible, but note that not even the most embellished words in the world could convey the depth of my gratefulness.

I would first like to thank my advisor Professor Roni Rosenfeld for everything he has taught me during my time working with him. Much of my growth from a computer science student who knows only of theories and little of real world problems, to someone who now has acquired experience in scientific research, is due to him. Professor Rosenfeld led me step by step from the beginning to gain all the knowledge I needed to not only conduct research, but to work with one of the most challenging problems in the field of speech technology today, and has not once lost his patience no matter how busy he was. I thank him for his painstaking guidance.

At the same time, I want to thank my other advisor, Dr. Jahanzeb Sherwani, who is a master mind behind much of the overall idea as well as many of the instrumental concepts of this research. In the course of our project Dr. Sherwani has taken time away from his recent highly successful startup company to work with us, contributing important ideas and giving me invaluable lessons in theory, software development, as well as team work. Being a recent PhD Graduate of Carnegie Mellon University's School of Computer Science himself, Dr. Sherwani is a true inspiration for me. Thank you.

Also, much of this research would not have happened for me without my undergraduate advisor Dean Mark Stehlik, whose email first led me to Professor Rosenfeld. Furthermore, Dean Stehlik was in charge of coordinating all of us undergraduate thesis students, providing us with enormous aid and advice. There is no better undergraduate advisor I could ask for.

And last but not least, I must thank my parents for all their love and support over the years, despite the excessive over-protectiveness for a single-child that is no small source of my exasperation. I thank both of them for their hard work to put me through my undergraduate years, and also thank my father, who was once a university professor, and still is a researcher, for all the useful advice.

Finally, I want to take back what I said at the beginning of this section, about having a clear idea of whom I want to thank. The truth is that there is no end to the people I can thank here. Even as I am set on finishing this acknowledgement, faces accumulate in my mind ever the faster. These include various people that helped out during this research, and others that simply put warmth in my heart everyday. They are (not in any particular order): Dan Kilgalin, Daegun Won, Toby Zhang, everyone who contributed voice data for our experiments, Sam Tetruashvili, Linda Cai, Mohammad Haque, David Chen, Linus Li ...and many many others, thank you!

Contents

List of Figures

Chapter 1

Introduction

Technologies in speech recognition has readily perpetrated daily lives of many advanced nations. From voice commands to electronics, to speech interfaces in the video games of recent years, the applications of accurate speech recognizers for a language continues to create more convenience for the people speaking that language. However, among the approximately 7000 living languages used in the world today, only a tiny fraction has been incorporated into commercial and open source speech engines, due to factors such as small potential market of the products and low availability of resources to develop them. Commercial packages like the Microsoft Speech Server (MSS) provides high quality recognizers for the worlds most commonly used languages and dialects. Open source recognition engines like the CMU Sphinx allow the training language models in any languages, but with the premise that the trainer has much knowledge about how speech technology works.

Studies in recent years have pointed to both the efficacy and practical benefits of developing speech technologies for areas of the developing world [7, 9, 15, 16, 17]. It is highly certain at this point that once high-quality and low-cost speech recognizer emerges, the low/semi-literate population will be provided many more productive means of communication, information access, data collection, etc. Therefore, speech technology is seen as a possible link between the developing areas and the information age.

At the same time, however, a growing understanding of the work has led to the identification of major road blocks in our efforts. In particular, high-quality automatic speech recognition (ASR) is an essential part of voice user interfaces (VUI) such as ones in various spoken dialogue systems (SDS). But a creating a good ASR for any language requires resources such as audio data, linguists, and technical experts; and the process of building the ASR takes time and money – all of which are unreasonable requirements in rural areas where many of the populations are low-literate.

To summarize many of the more immediate problems we face, we propose the following question: How do we develop a technique that will allow low-cost, accurate speech recognizers to be built, for any language in the world? We seek to provide one answer to this question in this research, by developing practical methods for creating speech recognizers for the resource-

scarce languages where many of the speakers are illiterate. Specifically, our design will focus on achieving following goals in building high-quality speech recognizers:

- The technique should work for any language.
- The technologies utilized must be cost-efficient.
- The resulting speech recognizers should be fit for use by the low-literate people.
- The speech recognizers should have high accuracy over small vocabulary.
- A speech recognizer built with an implementation of the technique should not require either linguistic or technical expertise.

Chapter 2

Background

Speech technology is a few decade old. Serious studies regarding speech technology for developing countries began only very recently, but yielded sanguine outlook on the potential applications. However, few good solutions have been proposed to solve the many problems we face in general.

2.1 Applications and Efficacy of Speech Technologies in the Developing World

The notion that speech technology can be a major factor in the advancement of the developing world is sustained by many observations. Particularly, relevant studies have often identified illiteracy/low-literacy as a major road-block in establishing technologies of the fist world countries in a underdeveloped region with very specific cultures and customs. Yet, despite the inability of many major technologies to take hold, cell phone has been a widespread success, readily absorbed by communities of developing areas [7]. It thus appears reasonable to expect SDS and ASR to be the bridge between the low-literate population and information technology.

Major efforts to experiment with speech and/or touch-tone systems in developing countries began with UC Berkley's TIER groups Tamil market project [9], and we have since seen a number of case studies and experiment conducted in the rural areas of the developing world [6, 8, 9, 15, 18]. Some of these studies, including The Tamil Market, CMU's Healthline, etc. yielded optimistic results on the effectiveness of speech recognition for the low-literate people. These findings, in turn, drove us to find general solutions that collectively solves the various common problems in effectively providing speech recognition technologies to a developing area.

And the issues involved in creating SR for the developing regions are understandably multi-faceted. The article by Brewer et al [4] included a overview of real challenges caused by illiteracy when it comes to speaker recruiting, audio collection, as well as user testing – all of which call for novel ways to achieve desired knowledge/result. Moreover, experiments conducted at Meraka Institute [1, 3] suggests that developing competent SR systems will require tens of speakers and up to hundreds of training samples per speaker. It is clear at this point that the traditional approach to building speech recognizers from scratch maybe to costly and also impractical in the

settings in which they are to be constructed.

2.2 Related Methods in Speech Recognition

Looking at techniques for speech recognition in the past, we take particular note on radical ones that could potentially eliminate expert involvement and may not require training data with vast coverage of a target language.

In the past two decades there has been many efforts to construct multilingual phoneme databases. One series of work done by Schultz and Waibel was the GlobalPhone project [10, 11, 12, 13], where large amount of data was collected from source languages, so that only a limited amount of training data will be required to create acoustic models in a new language. Their models generated by a data-driven approach could not beat those obtained from a heuristic approach, however, so there still was no satisfactory solutions to eliminate human involvement in building recognizers for a target language.

An earlier approach employing both a cross-language pronunciation transcription and a data-drive approach to automatically process speech was done by Constantine and Chollet [5]. Specifically, they employ relatively simple approach using GA to generate phoneme transcriptions based on a multilingual speech database.

A more recent work by Bansal, Nair, Singh and Raj [2] introduced a joint decoding algorithm on the training audio of a target language to automatically determine the pronunciations. However, modification of the decoding algorithm for audio has to be done at a low level in speech engines, and that excludes the prospects of using off-the shelf recognizers as a base-line in which training with the source language(s) has already been done.

2.3 The Salaam Approach

To date, one assuring solution proposed to comprehensively address many common issues in related works of the field is the Speech-based Automated Learning of Accent and Articulation Mapping (Salaam) by Sherwani [14], which is in turn first introduced as the ”Poor Man’s Speech Recognizer” in previous technical research publications by Sherwani et al on speech technology for low-literate users [15, 16, 17].

The published Salaam approach is aimed at building small-vocabulary recognizers by transcribing the pronunciation of a word from the target language into phonemes in the source language, using a well-trained speech recognizer as an underlying recognizer. Specifically, cross-language phoneme mapping using existing recognizers is employed as the work around to avoid training acoustic models relying on bountiful data. Moreover, the baseline recognizer was also used to semi-automatically decode training data of the target language to help with obtaining

more accurate pronunciations, improving upon those provided by a human expert.

Several existing research has tested all or parts of the Salaam approach. Sherwani's own test of the method at an international conference yielded less than 10% word error rate (WER) on various languages, with vocabulary sizes from ranging from 3 to 10 words[14]. A comparative study on voice interfaces in Rural India [8] has attained less than 6% WER over a small domains over languages with a system using recognizers trained on other languages. Another project conducted by the Meraka Institute in African test a few recognizers build atop recognizers trained on other languages [19], and received varied results from as high as over 90% to as low as just over 50% recognition rate. Both studies showed that the Salaam approach can reach promising performance but understandably falls short compared to recognizers trained directly using resources from the target language.

We base many of our ideas and designs on Salaam. Some of the method's details are described in the next chapter.

Chapter 3

Incorporating Salaam's Components

Due to the promise showed by the Salaam approach under various tests, we pick up on two of its most important components. The cross-language phoneme mapping and the data-driven approach.

3.1 Cross-Language Phoneme Mapping

Using an existing speech recognition system, cross-language phoneme mapping can be done by defining the word or phrase using a sequence of phonemes that are defined in the system, which in turn represents an entry in a lexicon file. For example, the standard North American pronunciation of the word "long" in phonemes of MSS U.S. English recognizer would be similar to "L AO NG."

A clear drawback of this approach is of course that the set of phonemes in the source language and the target language are most likely not the same. For instance, the Hebrew word for one has a uvular fricative phoneme that sounds like a mix between an "H" sound and the "K" sound in English. In such cases, we pick the one that our baseline recognizer agrees with the most given the training samples. So with the MSS U.S. English recognizer, the resulting pronunciation would be similar to "E H AA D" or "E K AA D", or both if the implementation allows multiple pronunciations per word.

3.2 Data-Driven Approach in Salaam

Incorporating a data-driven approach aims to help humans with the task generate a pronunciation for new words – i.e. the aforementioned cross-language transcription. The idea is largely reliant on the scoring of recognition results returned by the baseline recognizer. It follows that, if the recognizer is given a large set of potential phoneme sequences, it would pick out the ones that matches the audio input, and provide acoustic score and/or confidence score we can then use them to pick out the best pronunciations for the training data. But, trying to match even a 5 phoneme sequence creates the search space of 37^5 distinct sequences on a recognizer with 37

phonemes, making the task computationally impractical.

The design described by the published Salaam methods is a semi-automatic pronunciation generation technique that also addresses the computational complexity issue by having a linguistic expert fix down a number of phonemes that humans are more certain of (e.g. the consonants) and then create arbitrary word boundaries in the word. The former part of this design endeavors to relieve the recognizer of problems human experts can solve; and the latter effectively make the resulting words induced by the boundaries "separate problems." In practice, if we have a word where there are 2 phonemes the expert is uncertain of, one can place the word boundary somewhere between the two phonemes, the Salaam method will match each separate word with a set of sequences whose size is equal to or less than the total number of phonemes in the baseline recognizer. So in general, if there are N phonemes in the recognizer and there are n uncertain phonemes, the complexity of the search can be reduced to $O(nN)$.

3.3 Means for Automated Learning

The most direct approach to cross-language phoneme mapping is to involve a language expert who has knowledge of both the source and the target language, as well as a certain level of understanding for how phonology works in speech technologies. But in the setting of a developing area, obtaining or training one such personnel can be difficult and costly.

To eliminate the need for human linguistic experts, the published Salaam method suggested using a existing speech engine's letter-to-sound rules to generate the initial pronunciations with the help of a foreign word expressed in the English alphabets provided by a native speaker (much like typing on instant messaging or SMS text messages). As such, Salaam moves much of the burdens in pronunciation generation away from human.

Chapter 4

Method Design

Our method adopt cross-language phoneme mapping directly from Salaam.

As for the data-driven method, our work aims for a design that overcomes the limitations of the old method in the following areas:

1. The published methods reliance/assumptions on the phonemes fixed by the expert or text-to-sound rules, and on the total number of phonemes in the target pronunciation.
2. The reliance on word boundaries to reduce computational complexity. We want the resulting pronunciation for each word to consist of a single, continuous phoneme sequence.

Removing the hints provide by human/test-to-sound rules implies that the baseline recognizer must be used to generate the entire phoneme sequence. To do this, we must look at some subsets of all possible phoneme sequences/decoding, and take the ones that the recognizer agrees with the most given the audio data of the target word. But as pointed out before, the set of potential phoneme sequences cannot be too large. So due to computing limitation we still need to use word boundaries, albeit in a different manner, to cut down on the size of the search space.

4.1 Method Design

We designed an iterative algorithm that progressively generates phonemes resulting in a decoded sequence that has been given a relatively high score by the underlying recognizer. To start, we explicitly enumerate all possible phoneme sequences for the first 1 through 3 phonemes for the recognizer to match on the training audio, and then try to match a number of phonemes after the initial 1 through 3, with word boundaries separating them. A graphical representation of this would be:

" $*$ / $*$ / $*$ / $*$ ..." or " $**$/ $*$ / $*$ / $*$ ..." or " $***$ / $*$ / $*$ / $*$ ..."

Where the " $*$ " denotes any phoneme, and "/" denotes the word boundaries. For the sake of easy representation, we write this setup as "$Phx[1-3]/Ph[n]$", where the "$Phx[1-3]$" is the explicitly enumerated phoneme sequences of length 1 through 3, and the "$Ph[n]$" represents n word boundary separated phoneme sequences we match after the initial explicitly enumerated

ones [1]. By this setup, when the recognizer decodes the audio data it evaluates one of the three cases: the first phoneme by itself (separated by a word boundary from the rest), or along with another phoneme, or along with two more phonemes. We accept the very first phonemes from recognition results as the potential first phoneme in the final pronunciation. Note we do not just take the sequence with the highest score because there are still boundaries in that sequence; so a phoneme from a recognition result with low score may in fact be a part of a high-score pronunciation that is without word boundaries.

In the second iteration, we concatenate each phoneme obtained from iteration one to the beginning of our "$Phx[1-3]/Ph[n]$" rule. So if the first iteration gave produced the set P of m phonemes, we will have m "$XPhx[1-3]/Ph[n]$" sequences sets to match in this step for each phoneme "X" in P. With the same logic as the first iteration, we can now accept the very first 2 phonemes from each recognition result as the potential first and second phonemes in the final pronunciation.

Likewise, at iteration i, we should have a set P_{i-1} of phoneme sequences each consisting of $i-1$ phonemes from iteration $i-1$. For each phoneme sequence "$X[i-1]$" in P_{i-1}, we will match "$X[i-1]Phx[1-3]/Ph[n]$" in the current iteration. The algorithm stops when on an iteration j, we no longer produce any more phonemes from "$Phx[1-3]/Ph[n]$", i.e. the recognition results are no longer than $j-1$ phonemes. Then we can accept the "$X[j-1]$" phoneme sequences that have the highest scores given by the recognizer as the final pronunciations.

The size in the search space for each sequence in the set P_{i-1}, using a recognizer with N built-in phonemes, is $N+N^2+N^3+nN$. We can further limit size of P_{i-1} to be M, and upper-bound the length of the final pronunciation of any word to be L, then the overall complexity in the search space of pronunciation generation for that word is $O(N^3ML+nNML)$.

4.2 Generating a Pronunciation

As an example, I demonstrate here how our technique generates pronunciations for the Hebrew word for "one", which we represent here as "ehad", using the English recognizer from the microsoft speech server (see Appendix B).

We begin with a English recognizer, and a set of audio samples files of "ehad". Then we enter the algorithm to successively generate phonemes.

In the first iteration, we build a grammar that allows the recognizer to match the audio sample with all sequences of length 1 through 3 of MSS's English recognizer's phonemes, repeated from 0 up to 10 times. Conceptually, this is to matching the audio samples to the following sequences:

[1] "$Ph[n]$" can have any number of word boundaries up to n. The greater the number of word boundaries, the less complex the search. When the number of word bundaries is n, the size of the search space becomes $O(nN)$, for a baseline recognizer with N phonemes.

None
AA
AE
AH
...
Z
ZH
AA AA
AA AE
...
ZH ZH
AA AA AA
AA AA AE
...
ZH ZH ZH

and we allow the recognizer to treat each audio sample as mulitple words, and match each word to one of the above sequences.

The recognition results pooled from all samples from our run consists of “K AA D”, “T AA D”, “H AA D”, “K AO D”, “T AO D”, and “H AO D”. As this is the first iteration, we accept the very first phoneme from each result as the pontential first phoneme in our final sequence. In this case, we record “K”, “H”, and “T” for the next round.

In the second iteration, we again build a grammar that consists of all the sequences in the grammar in the first iteration, only after each of the three first phonemes:

None
K
K AA
K AE
...
K ZH ZH ZH
T
T AA
...
T ZH ZH ZH
H
H AA
...

we again allow the recognizer to treat each audio sample as 0-10 words, only the first word must be matched with one of the above sequences. The words that follow must be matched with one of the sequences from the very same grammar from the first iteration.

The recognition results pool from all samples of our run. Because this is the second iteration,

we store the first two phonemes of each result for our next iteration.

The algorithm then repeats as the second iteration, until we arrive at iteration four, and obtain "K AA D" as the best recognition result, which is consists of only 3 phonemes. This means we have not generated anymore phoneme from this iteration (or, no length 4 phoneme sequences are as good as "K AA D"). Then, instead of storing the best sequences of length upto four for another pass, we output the best single-word recognition results from the current pass as entries for "ehad" to the lexicon of our new hebrew recognizer. And the top three entries consists of:

K AA D
K AA AA D
K O AA D

Appendix C shows an example of a complete set of generated pronunciations for fifty Hebrew words.

Chapter 5

Experimentation and Results

5.1 Data Collection

For experimenting with our technique, we have compiled a list of 50 words/short phrases (see Appendix A) in English, consisting of numbers, commands to computer systems, and disease names. Each entry was selected because it consists of one word or a short phrase, and it pertains to the topic of a service provided by an SDS system (one that may use a speech recognizer built with our technique). Because we aim for small-vocabulary speech recognizers, vocabulary sizes of 50 or less is a good baseline for us to conduct our experiments. The first speaker for each language provides the translation of the words in to that language in that language's writing, and we adhere to that translation for all subsequent recordings in that language.

At first we have recorded audio data using desktop microphones. But during the earlier stages of the research we decided that the only recording medium we use would be either traditional/digital landline or cellular telephones, for they are prevalent in developing regions and are what we expect the recognizers to work on. This also eliminates some of the problems we may face, because different inputs can provide different audio data. Currently, the data we have has been tested to all have 8kHz sample rate. However, we have not addressed the possible effects from encryptions used by cellphone services and desktop audio input, nor the potential difference in quality between digital and traditional landline telephones.

We have built an SDS using VoiceXML for collecting audio data, hosted by Voxeo [1]. During a recording session, a participant is prompted to read each of the 50 words one at a time. To obtain more than one samples we iterate over the words to minimizes the effect of repetition on the way a particular word is pronounced.

For the result I will present, we have used data from from two speakers each for Yoruba and Hindi, and from three speakers for Hebrew. Each speaker provided five samples for each word.

[1] www.voxeo.com.

5.2 Experiment Results

In this section we discuss results from four experiments on our technique.

Pronunciations generated from Single Speakers vs. Expert Pronunciations

The earliest set of promising results for the design described in this thesis consists of five-fold cross-validation tests on the voice data of single speakers.

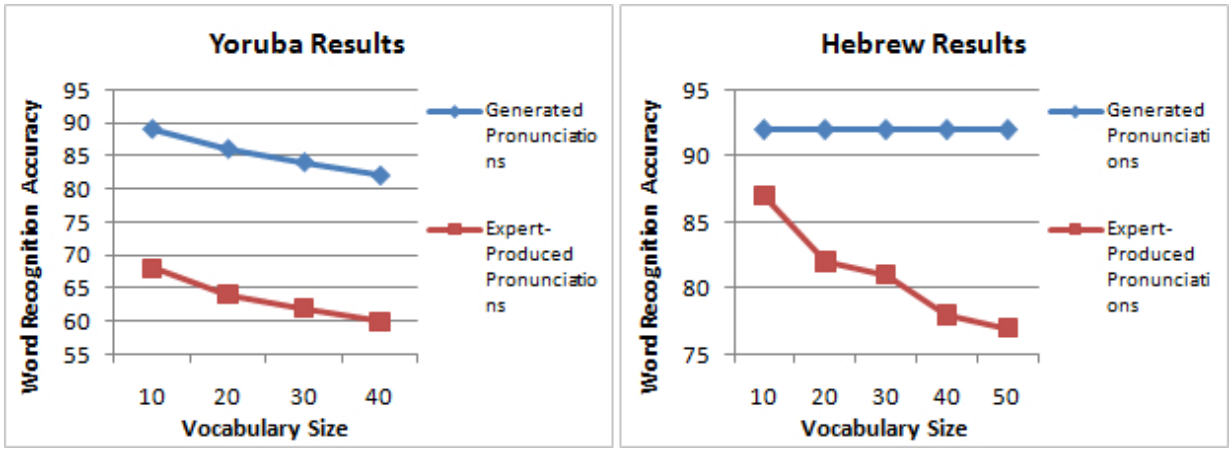

Figure 5.1: Recognition results for Yoruba and Hebrew for both manual and automatically generated pronunciations with varying vocabulary size.

As expected, word recognition accuracy goes down, generally, as vocabulary size goes up [2]. The automatically generated pronunciation result for Hebrew was especially interesting because every test audio was correctly recognized, except for a few times when the four samples were not enough to generate a pronunciation in Hebrew, so there was no recognition, which explain the consistent high, but not perfect, recognition accuracy .

Furthermore, for these two languages, we have obtained phoneme sequences of each word written down by one language expert of Hebrew, and one of Yoruba. The word recognition accuracy on all five samples using expert pronunciations are then pitted against the word recognition accuracy average across the 5-folds of our own generated pronunciations from four training samples on one test sample. And the results from from the experiments on both Yoruba and Hebrew demonstrate that the automatically generated pronunciations readily beats pronunciations provided by experts.

[2]Word recognition accuracy is none increasing as vocabulary sizes increase. This can be seen in all of our experiments, and is generally true for almost all cases in speech recognition.

Cross-Speaker Results Pronunciations Generated from Single Speakers

The experiments that immediately followed was a test on cross-speaker recognition accuracy. The pronunciations trained from each speaker are tested on the two others.

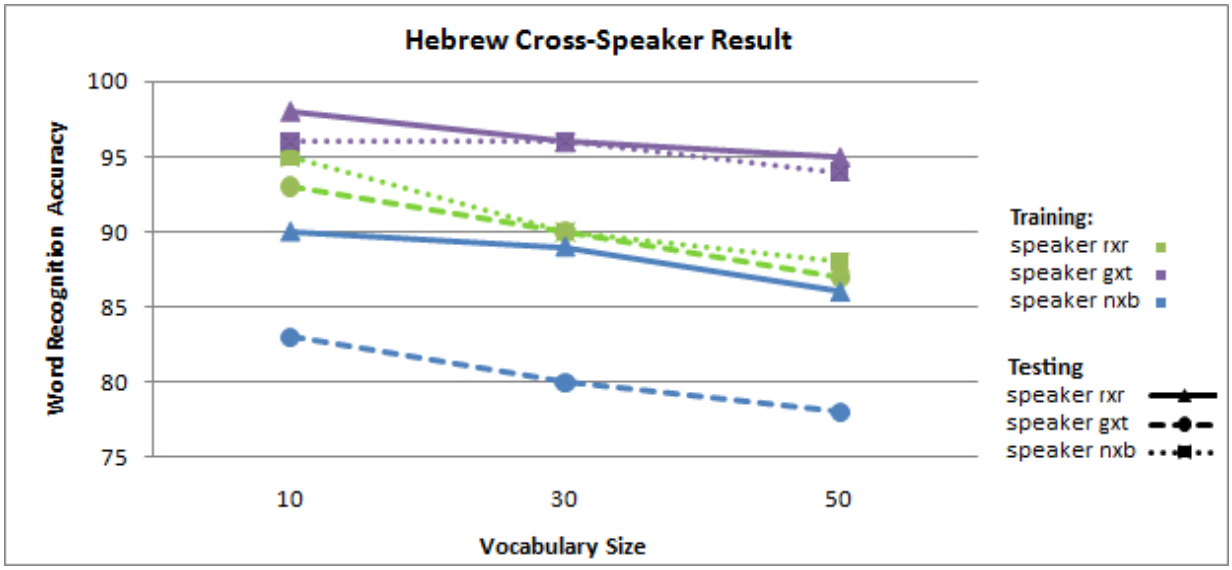

Figure 5.2: Cross-Speaker results for pronunciations trained on single speakers.

The immediate observation from the results was a surprise: there is a clear correlation between the speaker of the training data and recognition accuracy. While pronunciations trained on speaker gxt worked fantastically, and those trained with data from speaker rxr also performed satisfactorily, those from speaker nxb did not always do very well.

Next, there also seems to be are relationship in the effectiveness of the training data on the testing data between the speakers. Pronunciations trained on rxr yielded a subtly higher word recognition accuracy when tested on nxb, and at the same time, those trained on nxb clearly favor rxr's voice much more.

Besides the all the questions this set or results poses on the causes of the correlations, it also provided the important implication on the potential benefits of training on multiple speakers.

Multiple Pronunciations Per Word Generated from a Single Speaker

In this experiment was designed to study the benefits of mapping multiple pronunciations to a single word in a recognizer.

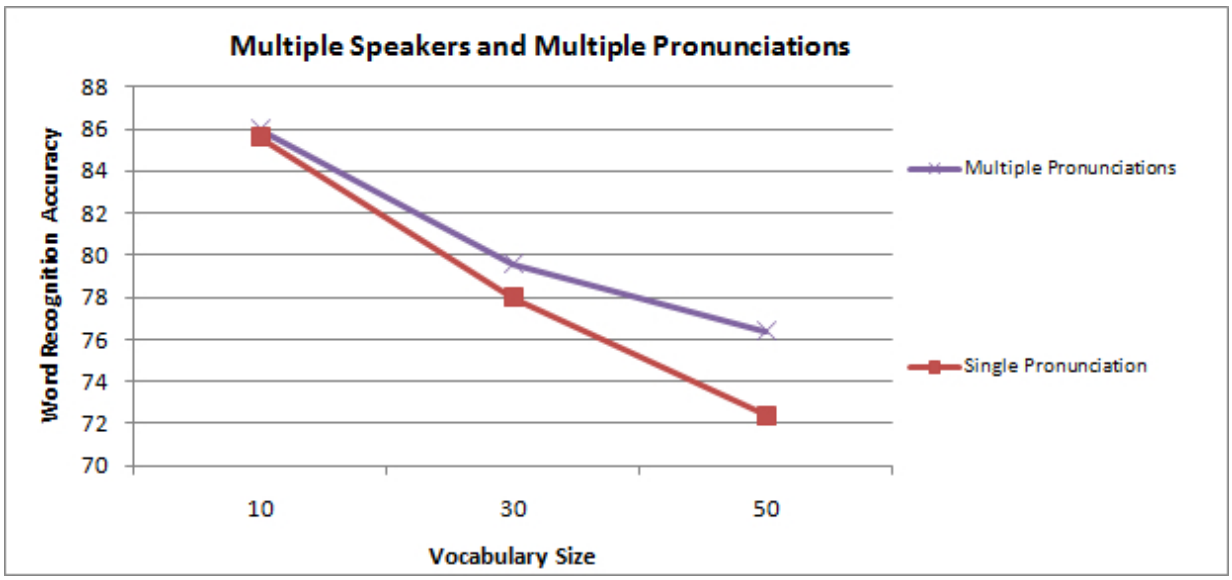

Figure 5.3: Recognition results for Hindi of a recognizer with multiple pronunciations for a word, and a recognizer with with a single pronunciation for each word.

This time, we accept three top unique results during pronunciation generation when training on one speaker. For the sake of comparison, we take the same exact set of pronunciations and remove the second and third best pronunciations. The two set of pronunciations are then pitted against each other when testing on the data of the other speaker.

The result shows that the benefits of mapping multiple pronunciation to a word is not apparent when the vocabulary size is very small. Although as the vocabulary size gets bigger, improvements in recognition accuracy from having alternative pronunciations can be significant. Since our method and looks at many alternative pronunciations during generation, outputting alternatives does not cost more computing power. This is a trick worth incorporating.

Multiple Pronunciations Per Word Generated from Multiple Speakers

In this final experiment I present, we combined the two ideas obtained from the last two experiments. We generate multiple pronunciations for each word by training on multiple speakers, and pit the results against an instance of the results from single speaker cross-speaker experiments.

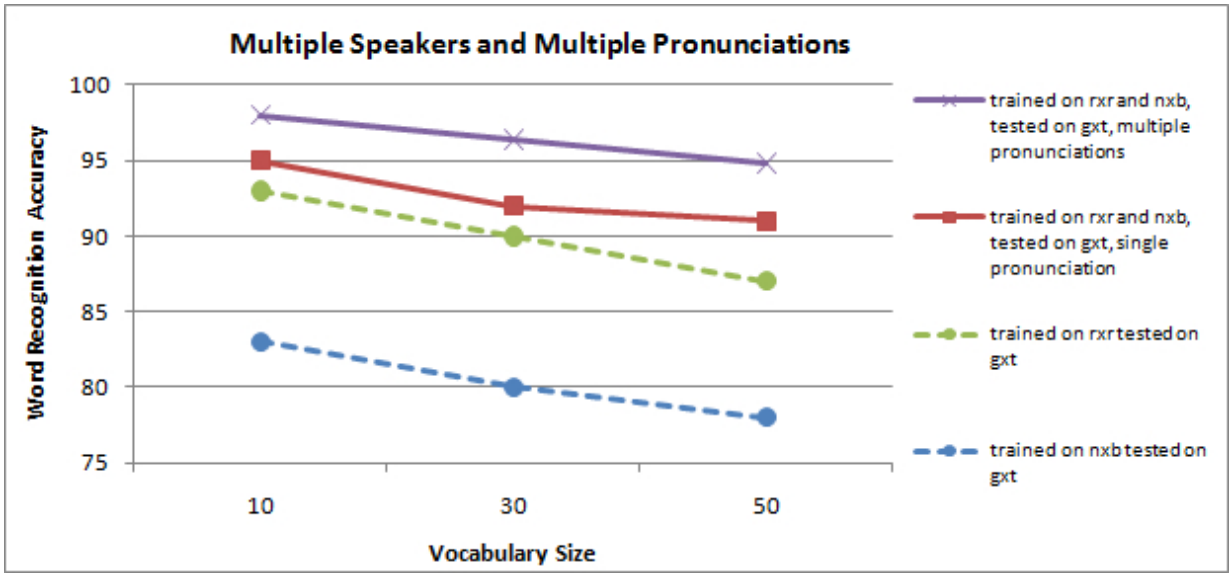

Figure 5.4: Cross-Speaker results for pronunciations trained on single speakers.

As we can see, with training data from the same two speakers rxr and nxb, and testing on the same test data from speaker gxt, generation from the combined training data set produced results that out performs pronunciations generated by single speakers from either rxr or nxb. Moreover, when we accepting multiple pronunciations during generation for each word, the resulting recognitiong accuracy gains a significant boost. This outcome presents a very good promise for our technique.

Chapter 6

Conclusion

Results from the last chapter present empirical confirmation that our method can achieve high recognition accuracy over a small vocabulary for a language without any involvement of human experts or reliance on sizable language resources. Pronunciations in the source language generated by our algorithm consistently outperform those provided by linguistic experts, hence proving our method to be a potent way to apply cross-language phoneme mapping when training data in a target language is lacking. Furthermore, we have also shown that one can improve upon the quality of a recognizer built with our technique by expanding the training set size and the number of speakers for training, or mapping multiple pronunciations to a single word. Futher studies can help discover other strategies to use in junction with this technique.

Although we only have results from three different languages, these languages come from three different areas and belong to distinct language families: the Afroasiatic languages (Hebrew), the Niger-Congo languages (Yoruba), the Indo-Aryan languages (Hindi); and the method yielded satisfactory results for all. There is a slightly greater implication for the Yoruba and the Hindi test sets - these languages are used by some developing areas of the world, and no deployable speech technology has been developed for them so far. It would be very useful to study this technique using other languages, especially ones from regions with significant low-literacy. We also look forward to field-testings in developing regions with recognizers built with our method.

As per our description of the method's design in 4.1, implementation of our method should not entail low-level modifications to a speech recognition engine of the source language - our design could be implemented using any recognizer, even closed-sourced ones. It could be an interesting research in the future to test the method's effectiveness/performance when implemented with different base-line recognizers.

Appendix A

List of 50 Words/Phrases used for Data Collection

One	Hello
Two	Goodbye
Three	Faster
Four	Slower
Five	Select
Six	Start
Seven	Stop
Eight	Delete
Nine	Add
Ten	Open
Eleven	Close
Twenty	Sleep
Thirty	Fever
Forty	Smallpox
One Hundred	Chickenpox
Two Hundred	Cancer
One Thousand	AIDS
One Million	Hepatitis
Repeat	Malaria
Next	Diarrhea
Previous	Diabetes
Go back /Scratch that	Infection
Reverse	Symptoms
Yes, thats correct	First
No, thats wrong	Second

Appendix B

List of Phonemes for the English Recognizer of MSS

Vowels	Consonants
AA	AX L
AE	AX M
AH	AX N
AU	B
AO	CH
AX	D
AX RA	DH
EH	F
EH RA	G
EI	H
ER	J
I	JH
IH	K
O + UH	L
OI	M
U	N
UH	NG
	P
	RA
	S
	SH
	T
	TH
	V
	W
	Z
	ZH

* Taken from http://msdn.microsoft.com/en-us/library/bb813894.aspx, retrieved in May, 2010.

Appendix C

Example Pronunciations Generated (Hebrew)

Pronunciation	Word/Phrase	Pronunciation	Word/Phrase
K AA D	Ehad	SH AX L O M	Shalom
SH N AI EI M	Shnaim	L I D AO L O T	Lehitraot
SH AX L O L SH	Shalosh	M EH H EH EI AX T EI L	Maher yoter
AA L B AA AX	Arba	M EH AH CH U T EI L	Leat yoter
H AH M EI SH SH	Hamesh	DH AX H AA R L	Bhar
SH EH SH	Shesh	AX K S EI AX N L D	Hathel
SH EH V RA AX	Sheva	AX K S IH L L	Atsor
SH M AO RA L I	Shmoneh	DH AX H AA K K	Mhaq
G UH SH EH AX	Tesha	H O S EI AX S F	Hoseff
EH S EH L	Eser	T K D AA H O NG T	Ptah
H AO T AX F L I	Ahat essreh	S M G O AX L L	Sgor
Z S L I M	Esreem	N AH S U SH O N	Leh lishon
S L SH I M	Shloshim	H O M	Homme
H O B R EI I M	Arba im	V IH L EH K AX P L IH L T	Daleket reot
M EI EH AX	Meah	AX B AA B L O K U L H O NG	Abaabuot ruah
M AX T D AI EI M	Matayim	F O P D AH N	Sartann
EH N IH V	Ellef	EI D S	Aids
M I O N	Milion	S EH H EH V IH T	Tsahevet
EH M O SH U U V	Emor shoov	M AX L O I AX	Malarya
AX B AA	Haba	SH U SH U U N F	Shilshool
AA K W IH D EI AX M	Hakodem	S AX K IH EH D EI S T	Sakeret
S K AO S M I Z EI	Shkah mizeh	Z I H O M	Zeehoom
P AX F UH L S S	Hafoh	F IH NG S T O M I M	Simptomim
DH AX H O L AX N	Nahon	L I SH O AO N	Rishon
L O DH AX H O L N	Lo nahon	SH IH N M I	Sheni

Bibliography

[1] Jacob A. C. Badenhorst and Marelie H. Davel. *Data requirements for speaker independent acoustic models*. Cape Town, South Africa, November 2008. 2.1

[2] D. Bansal, N. Nair, R. Singh, and B. Raj. A joint decoding algorithm for multiple-example-based addition of words to a pronunciation lexicon. In *Proc. ICASSP*, 2009. 2.2

[3] E. Barnard, M. Davel, and van Heerden C. Asr corpus design for resource-scarce languages. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech)*, September 2009. 2.1

[4] E. Brewer, M. Demmer, M. Ho, R.J. Honicky, M. Plauchè J. Pal, and S. Surana. The challenges of technology research for developing regions. *IEEE Pervasive Computing*, 5 (2):15–23, April–June 2006. 2.1

[5] A. Constantinescu and G. Chollet. On cross-language experiments and data-driven units for automatic language independent speech processing. In *Proceedings Automatic Speech Recognition and Understanding Workshop*, pages 606–613, St. Barbara, CA, 1997. 2.2

[6] A. Grover, M. Plauchè, and C. Kuun. *HIV health information access using spoken dialogue systems: Touchtone vs. Speech*. Doha, Qatar, April 2009. 2.1

[7] ITU. Measuring the information society: The ict development index. URL `http://www.itu.int/ITU-D/ict/publications/idi/2009/index.html`. Accessed May, 2009. 1, 2.1

[8] N. Patel, S. Agarwal, N. Rajput, A. Nanavati, P. Dave, and T. S. Parikh. A comparative study of speech and dialed input voice interfaces in rural india. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, 2009. 2.1, 2.3

[9] M. Plauchè, U. Nallasamy, J. Pal, C. Wooters, and D. Ramachandran. Speech recognition for illiterate access to information and technology. In *Proc. International Conference on Information and Communciations Technologies and Development*, 2006. 1, 2.1

[10] T. Schultz and A. Waibel. *Fast Bootstrapping of LVCSR Systems With Multilingual Phoneme Sets*. Rhodes, 1997. 2.2

[11] T. Schultz and A. Waibel. *Adaptation of Pronunciation Dictionaries for Recognition of Unseen Languages*. St. Petersburg, Russia, October 1998. 2.2

[12] T. Schultz and A. Waibel. *Language Independent and Language Adaptive Large Vocabulary Speech Recognition*. Sydney, 1998. 2.2

[13] T. Schultz, M. Westphal, and A. Waibel. The globalphone project: Multilingual lvcsr with

janus-3. In *Proc. SQEL*, pages 20–27, 1997. 2.2

[14] J. Sherwani. *Speech Interface for Information Access by Low-Literate Users in the Developing World*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA, May 2009. Also published as technical report CMU-CS-09-131. 2.3

[15] J. Sherwani and R. Rosenfeld. *Speech vs. Touch-tone: Telephony Interfaces for Information Access by Low-Literate Users*. Doha, Qatar, 2009. 1, 2.1, 2.3

[16] J. Sherwani, N. Ali, S. Miraza, A. Fatma, Y. Memon, M. Karim, R. Tongia, and R. Rosenfeld. *HealthLine: Speech-based Access to Health Information by Low-literate Users*. Bangalore, India, 2007. 1, 2.3

[17] J. Sherwani, N. Ali, R. Tongia, R. Rosenfeld, Y. Memon, M. Karim, and G. Pappas. *HealthLine: Towards Speech-based Access to Health Information by Semi-literate Users*. Singapore, 2007. 1, 2.3

[18] J. Sherwani, R. Tongia, R. Rosenfeld, Y. Memon, M. Karim, and G. Pappas. *Towards Speech Interface for Health Information by Semi-literate Users*. Hyderabad, India, 2007. 2.1

[19] C. Van Heerden, E. Barnard, and M. Davel. Basic speech recognition for spoken dialogues. In *Proceedings of the 10th Annual conference of the International Speech Communication ASsociation (Interspeech 2009)*, pages 3003–3006, September 2009. 2.3