Small-Vocabulary Speech Recognition for Resource-Scarce Languages

Fang Qiao
School of Computer Science
Carnegie Mellon University
fqiao@andrew.cmu.edu

Jahanzeb Sherwani
iTeleport LLC
j@iteleportmobile.com

Roni Rosenfeld
School of Computer Science
Carnegie Mellon University
roni@cs.cmu.edu

ABSTRACT

We describe a technique for attaining high-accuracy, small-vocabulary speech recognition capability in resource-scarce languages that requires minimal audio data collection and no speech technology expertise. We start with an off-the-shelf commercial speech recognizer that has been trained extensively on a resource-rich language such as English. We then derive phonemic representations for any desired word in any target language, by a process of cross-language phonemic mapping. We show that this results in high accuracy recognition of vocabularies of up to several dozen words – enough for many development-related applications such as information access, data collection, and simple transactions.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Voice I/O*.

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Speech recognition and synthesis I/O*.

General Terms

Algorithms, Human Factors, Languages.

Keywords

ICT4D, SLT4D, Small Vocabulary, Resource-Scarce Languages.

1. INTRODUCTION

Recent studies have pointed to potential benefits of developing speech technologies for developing regions [7, 9, 15, 16]. In particular, high-quality automatic speech recognition (ASR) is an essential part of spoken dialog systems (SDS), which have particularly high potential in telephone-based applications. Such applications are particularly relevant for the ICTD community as they leverage the high penetration rates of mobile phones, require only the ability to make a phone call, and perhaps most importantly, can be used by both literate as well as non-literate users. However, among the approximately 7000 living languages spoken in the world today, only a tiny fraction have been incorporated into speech recognizers, primarily due to market forces, as well as the limited availability of experts in speech recognition technology. Commercial packages like the Microsoft Speech Server (MSS) provide high-quality recognition for a few dozen languages and dialects most commonly used in the developed world. Open source recognition engines like Carnegie Mellon University's Sphinx and open-platform tools like HTK allow in principle the creation of speech recognizers in any language, but require very significant amounts of recordings in the target language to be collected and processed. To achieve adequate accuracy, they also require significant speech technology expertise for training and tuning the system. Thus the process of creating ASR capability in a new language requires significant time, money and expertise – daunting requirements in developing regions with limited financial resources and overstretched workers.

Recognizing this technological impediment to the otherwise large potential of spoken dialog systems in the developing world, we set out to develop a technique that will allow a low-cost, accurate speech recognizer to be built for any language. Specifically, we sought a technique that would:

- work for any language
- require very minimal data collection effort (on the order of 3-5 repetitions of each word), which could be done over the phone
- require no linguistic or speech technology expertise
- result in a speech recognizer suitable for use by low-literate users
- provide high-accuracy (>95%) recognition over vocabularies of up to a few dozen words

2. BACKGROUND

2.1 Speech Technologies for the Developing World

Speech recognition technology is a few decades old. However, serious studies of speech technology for development-related applications began only recently. The notion that speech technology can play a positive role in development is suggested by the observation that illiteracy and low-literacy are major roadblocks to the wider dissemination of information services in the developing world. Despite the inability of many major technologies to take hold, the cell phone has been a widespread success, readily absorbed by virtually all developing communities [7]. Thus telephone-based spoken dialog systems appear promising for bridging the gap between low-literate populations and the information society.

ACM DEV'10, December 17-18, 2010, London, United Kingdom.
Copyright 2010 ACM 978-1-4503-0473-3-10/15/10... $10

Experimentation with speech interfaces in developing countries began with UC Berkley's TIER group's Tamil Market project [9], and was soon followed by several other pilot experiments and case studies [6, 8, 15, 16]. Some of these studies (e.g. [15]) demonstrated that speech interfaces can be effective for low-literate users, while others (e.g. [9]) pointed to the need for high recognition accuracy. These findings motivated the work we report here. Both [4] and [15] discuss challenges to speaker recruiting, audio collection and user testing caused by illiteracy.

2.2 Related Methods

We seek a technique for obtaining high accuracy speech recognition in any language without relying on much data collection or technological expertise. Experiments conducted at Meraka Institute [1, 3] suggest that developing competent general-purpose SR systems from scratch will require tens of speakers and up to hundreds of training samples per speaker. For a resource-strapped developing world NGO, this may be unachievable. Instead, we seek methods that have fewer requirements, even if they can only support very small vocabularies.

In the past two decades there have been many efforts to construct multilingual phoneme databases. One line of work by Schultz et al. is the GlobalPhone project [10, 11, 12, 13], where large amounts of speech data were collected from various source languages, so that only a limited amount of training data in the target language would be required to create acoustic models for that language. This approach still requires a moderate amount of data recording and a fair amount of expertise, and is geared towards creating unrestricted, large-vocabulary, moderate-accuracy speech recognition capability. As such, it is not optimal for the small-vocabulary, high-accuracy recognition capability we believe is needed for development-oriented applications.

An earlier attempt employing both a cross-language pronunciation transcription and a data-driven approach to automatically process speech was reported by Constantine and Chollet [5]. Specifically, they employ a relatively simple variation of Genetic Algorithms to generate phoneme transcriptions based on a multilingual speech database.

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 derive pronunciations. However, modification of the decoding algorithm for audio has to be done at a low level in the speech engine, which both requires technical expertise and excludes the use of commercial recognizers that employ highly-trained acoustic models.

2.3 The Salaam Method

One promising approach to our problem is the Speech-based Automated Learning of Accent and Articulation Mapping (Salaam) method [14], which is a refinement of the "Poor Man's Speech Recognizer" (PMSR) method described in [15, 16]. In the PMSR method, a speech expert builds small-vocabulary recognizers by transcribing the pronunciation of a word from the target language into phonemes in the source language. Specifically, by employing cross-language phoneme mapping using existing acoustic models, one can avoid training new acoustic models, often the costliest and most complex part of training a speech recognizer. While PMSR required a speech expert to manually define word pronunciations, in the Salaam approach the speech recognizer was used to semi-automatically decode a few recorded samples of each target word to obtain more accurate pronunciations, improving upon those provided by a human expert (and diminishing the need for such an expert).

The idea of representing foreign words by automatically derived cross-language pronunciations is not new to Salaam. It has been tried before by many researchers using so called "all-phone decoding" in open speech recognition platforms such as Sphinx or HTK. But anecdotal reports suggested that the accuracy of such an approach is insufficient even for a vocabulary of as few as 10 words, which is the smallest vocabulary needed for all but the most trivial applications. The gist of the Salaam idea is to use the same approach but to also take advantage of the superior quality and robustness of commercial recognition systems, which are trained on hundreds of hours of speech recordings and are carefully tuned by expert speech engineers. Since commercial systems do not usually provide the rich interface needed to run all-phone decoding, the Salaam method effectively achieves the same result by heuristically querying the commercial recognition engine through whatever interface it supports. Thus the Salaam method is not a new modeling technique but rather a practical method for enabling highly accurate spoken language interfaces in new languages with very minimal training data and no technological expertise.

The Salaam method was first tested anecdotally as part of a live demonstration during the ICTD 2009 conference in Doha, and yielded less than 10% word error rate (WER) on ten diverse languages, with vocabulary sizes ranging from 3 to 10 words [14]. Using a similar technique, a comparative study on voice interfaces using a prototype system by IBM Research in rural India [8] has attained less than 6% WER with sentences/phrases of the target language mapped to English phonemes, although the effective vocabulary size was only 2--3. These studies suggest that the Salaam method can yield good performance (though it still falls short compared to recognizers trained directly using significant resources from the target language).

Our proposed solution builds upon the Salaam method. We review key details of that method in the next section.

3. INCORPORATING SALAAM'S COMPONENTS

To take advantage of the potential shown by the Salaam method, we pick up on two of its most important components: the cross-language phoneme mapping and the data-driven optimization.

3.1 Cross-Language Phoneme Mapping

Using an existing, highly-trained speech recognition system in a *source language*, cross-language phoneme mapping is done by defining each word or phrase in the *target language* using a sequence of source-language phonemes. An obvious problem with this approach is that the phonemes of the source language and the target language are different, sometimes dramatically so. For instance, the Hebrew word for "one" has an uvular fricative phoneme that sounds like a mix between the "H" and "K" phonemes in English. In such cases, we pick the phoneme that most closely matches 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 multiple pronunciations per word are allowed.

3.2 Data-Driven Approach in Salaam

In the original Salaam method, a data-driven approach is leveraged to aid the human expert with the task of generating a pronunciation for a new word – the aforementioned cross-language transcription. The idea is largely reliant on the scoring of recognition results returned by the baseline recognizer which is run in an “all-phone-decoding” mode, namely allowing it to return any sequence of phonemes, rather than regular vocabulary words. Since most commercial recognizers do not expose their “all-phone-decoding” capabilities, we simulate this mode by defining artificial words that consists of one, two or three phonemes. If the recognizer is given an exhaustive set of these “words”, it would pick out the ones that best match the audio samples, and provide acoustic and/or confidence scores that we can then use to select target pronunciations. However, with a typical phoneme set of, say, 37 phonemes, trying to match a sequence of even only 5 phonemes creates a search space of 37^5 distinct sequences, making the task computationally impractical.

The design described by the Salaam method 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 artificial word boundaries inside the word. The former action reduces the search space by relying on human expert knowledge, and the latter effectively partitions the problem into a set of smaller, separable and more tractable search problems. For example, if a word has 2 phonemes that the expert is uncertain of (e.g. S ? L ? M), one can place the artificial word boundary somewhere between the two unknown phonemes (e.g. S ? / L ? M), and the Salaam method will match each separate word with a set of pronunciation possibilities, whose size is equal to or less than the total number of phonemes in the baseline recognizer. In general, if there are N phonemes in the language and n uncertain phonemes in the target word or phrase, the complexity of the search can be reduced to $O(nN)$.

3.3 Means for Automated Learning

The original Salaam method for cross-language phoneme mapping required a language expert with deep knowledge of both the source and the target language, as well as a certain level of understanding of how phonology is used in speech technologies. But in the developing world setting, finding or training such an expert can be difficult.

To eliminate the need for human linguistic experts, Salaam introduced a further improvement: heuristic letter-to-sound rules are used to generate initial candidate pronunciations, starting from a written transliteration of the target word as typed by a native speaker of the target language, using a source language (e.g. English) alphabet (e.g. Indian cell phone users often Romanize Hindi in SMS text messages). This moved much of the burden in pronunciation generation away from reliance on human expertise.

4. OUR IMPROVED METHOD

The improved method we present here adopts cross-language phoneme mapping directly from Salaam. But we go further in relying only on minimal amounts of recorded data, and nothing else. Specifically, we attempt to overcome the limitations of Salaam in the following areas:

1. Salaam’s reliance on the phonemes fixed by the expert or letter-to-sound rules, and on a pre-determined fixed number of phonemes in the target pronunciation.
2. Salaam’s reliance on artificial word boundaries to reduce computational complexity. These boundaries are undesirable because modern speech recognizers use approximate acoustic matching at word boundaries, which degrades the acoustic match and results in suboptimal pronunciations.

Eliminating the reliance on hints provides by human experts or heuristic letter-to-sound rules means that the baseline recognizer must be used to generate the phoneme sequences from scratch, **without any prior knowledge of the word to be recognized**. To do this, we must look at some subsets of all possible phoneme sequences, and take the ones that the recognizer matches best given the audio samples of the target word. But as pointed out before, the set of potential phoneme sequences grows exponentially with the number of phonemes in the sequence. So due to computing limitations, we still leverage artificial word boundaries to cut down on the size of the search space, albeit in a different manner.

4.1 Details of the Improved Method

We designed an iterative algorithm that, for each desired word in the target language, uses a small number (between one and five) of recorded samples, and progressively generates phonemes resulting in a decoded phoneme sequence that has been given a relatively high score by the underlying recognizer. The speech recognition grammar used in this method hinges on one critical grammar element, which we call the super-wildcard. This super-wildcard can be described in the following shorthand:

$$\underbrace{\left\{X\right\}_1^3 / \left\{X\right\}_1^3 / \ldots / \left\{X\right\}_1^3}_{10}$$

$\{X\}$ represents a phoneme wildcard – namely, it can represent any phoneme in the speech recognizer’s phonetic vocabulary. The subscript and superscript denote that all permutations of between 1 and 3 phonemes are being represented, while the / represents an artificial word boundary. This super-wildcard consists of 10 such “words”, with each word consisting of all permutations of between 1 and 3 phonemes. It should be kept in mind that this super-wildcard is used to represent the pronunciation for a *single word*, and we use these artificial word boundaries only to reduce the computational complexity of the search task, and not to imply that the word itself is composed of multiple subwords.

We will describe the algorithm with reference to a concrete example. Specifically, we demonstrate here how our technique generates pronunciations for the Hebrew word for “one”, roughly pronounced “EH-HUD”, using the English recognizer from the Microsoft Speech Server.

In the first pass, the super-wildcard grammar is used on its own, and recognition is performed on a word's audio using this grammar. The recognition results from this pass are then parsed to determine what phonemes to consider for the final pronunciation. For the *i*th pass, we accept up to *i* phonemes, and so for the first pass, we accept only the first phoneme as the potential first phoneme in the final pronunciation. We keep a list of "competing" first phonemes, and we do not just take the sequence with the highest score, as the nature of artificial word boundaries makes the intermediate step a heuristic recognition result; so a phoneme from a recognition result with low score may in fact be a part of a high-score pronunciation once it is tried without word boundaries.

In the first iteration, the super-wildcard is used on its own, with each "word" unit comprising all the sequences of length 1 through 3 of MSS's English recognizer's phonemes, repeated from 0 up to 10 times across each sample. Concretely, each "word" unit consists of the following sequences:

AA
AE
AH
...
Z
ZH
AA AA
AA AE
...
ZH ZH
AA AA AA
AA AA AE
...
ZH ZH ZH

We allow the recognizer to treat each audio sample as consisting of from 0 up to 10 words, and match each word to one of the above sequences. Thus, the upper bound on the number of phonemes in a word that our system can recognize is 30 phonemes – large enough to adequately capture any word or short phrase.

Continuing with this particular example, the recognition results pooled from all samples from the first run consist of the following:

K AA D
T AA D
H AA D
K AO D
T AO D
H AO D

As this is the first iteration, we accept the very first phoneme from each result as the potential first phoneme in our final sequence. In this case, we record **K**, **H**, and **T**, and move to the next iteration.

In the second iteration, we again build a grammar that leverages the super-wildcard construct; however, we prepend the phonemes under consideration to the grammar. Thus, the complete form of the grammar may be represented as:

$$\{P\}\underbrace{\left\{X\right\}_1^3/\left\{X\right\}_1^3/\ldots/\left\{X\right\}_1^3}_{10}$$

Here, $\{P\}$ represents the set of phonemes under consideration till the current iteration – namely, K, H and T. Thus, the grammar for the first "word" in the second iteration consists of the following phoneme sequences:

K
K AA
K AE
...
K ZH ZH ZH
T
T AA
...
T ZH ZH ZH
H
H AA
...
H ZH ZH ZH

Based on the top scoring results of the second iteration of recognition, we now *fix the first two phonemes*.

The algorithm then repeats as in the previous iteration, Thus, we iteratively fix one more phoneme in every iteration, and then append the super-wildcard construct to help identify the next best phoneme. We continue this until we arrive at iteration four, and obtain K AA D as the best recognition result, which consists of only 3 phonemes. The stopping condition for the algorithm is to check if there are fewer than *i* phonemes discovered on iteration *i*, or if there are no *i*-length phoneme sequences with as high a score as the best pronunciation from the previous pass ("K AA D" in this case). In our example, this is exactly what has happened, and so we output the best single-word recognition results from the current pass as the pronunciation for "ehad" to the lexicon of our new Hebrew recognizer. The top three results consist of:

K AA D
K AA AA D
K O AA D

Using this technique, we are able to create pronunciation definitions for words or phrases without any *a priori* knowledge of the words' phonetics or length. In the next section, we describe evaluation of our method.

5. EVALUATION

5.1 Data Collection

To evaluate our method, a list of 50 words/short phrases in English was compiled, consisting of numbers, commands to a typical information-access applications, and disease names. Each entry was selected because it is either a single word or a short phrase, and it pertains to the topic of a service that could be provided by a Spoken Dialog System (SDS). Given our goal of high accuracy, small-vocabulary speech recognition, the vocabulary size was kept to a maximum of 50 words. Three target languages were chosen: Yoruba, Hindi, and Hebrew. The first recorded speaker for each target language provided the translation

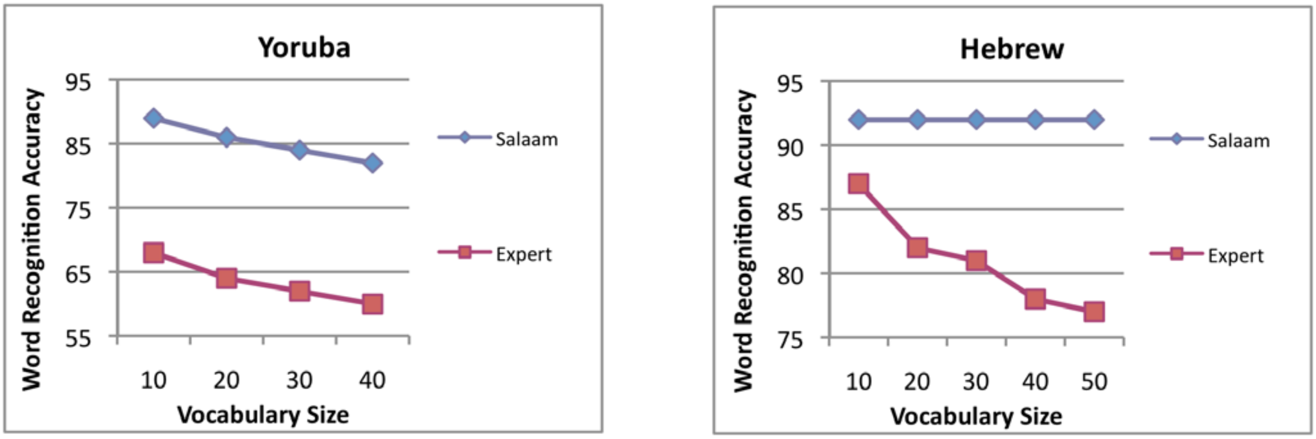

Figure 1. Same-speaker leave-one-out recognition accuracy for Yoruba and Hebrew for both manual and automatically generated pronunciations with varying vocabulary size.

of the 50 words into that language (written in that language's native writing system), and we adhered to that translation for all subsequent recordings in that language.

The source language we used was US English, using the Microsoft Speech Server bundled with Microsoft Unified Communications Managed API 2.0 SDK.

We recorded sample audio using both analog and digital landlines, as well as cellular telephones, since these are prevalent in developing regions and are what we expect the SDS applications to be used with. All recordings were done at 8kHz sampling rate. We have not addressed general dissimilarities between the sets of recordings we collected, such as possible differences in speech coding and compression used by different cellular carriers, or any difference in quality between digital and analog landline telephones.

We built an SDS for collecting audio data, using VoiceXML and hosted on Voxeo[1]. During each recording session, participants were prompted to read each of the 50 words one at a time. To obtain more than one sample per word, we had participants iterate over the entire set multiple times, collecting one sample of each word per iteration, rather than recording all samples of each word all at once, to minimize the effect of repeating the same word multiple times in quick succession, as this can drastically change the way a particular word is pronounced.

For the results presented below, we have used data from two speakers each for Yoruba and Hindi, and from three speakers for Hebrew[2]. Each speaker provided five samples for each word.

5.2 Results

5.2.1 Expert-Produced vs. Automatically-Generated Pronunciations (same-speaker)

The first set of results for the method described here is a same-speaker five-fold cross-validation test on pronunciations generated from four samples/words of single speakers, for Yoruba and Hebrew (See Figure 1). Alongside the results from our improved Salaam method, we have also shown recognition results based on expert-supplied pronunciations, from the older PMSR method.

As expected, word recognition accuracy generally degrades as vocabulary size increases. Most importantly, pronunciations generated automatically by our method result in recognition accuracy that is consistently, substantially, and statistically significantly better than that achieved with pronunciations generated by linguistic experts. The automatically generated pronunciation result for Hebrew is especially noteworthy, in that the few recognition failures were all due to failure of our method to produce any pronunciations (this happens when no vocabulary choice provides reasonable match to the recording, as might happen if there is excessive noise during the recording or particularly unusual pronunciation). In other words, for those words for which our method did produce a pronunciation, subsequent recognition accuracy was 100%. This is significant because a failure to produce a pronunciation can be detected at training time and corrective action can be taken: collecting more samples, using expert-selected pronunciations, or suggesting to the developer that they use alternative wording.

[1] www.voxeo.com.

[2] Although Hebrew is not a developing world language, we chose it out of convenience and to demonstrate that our technique works across very different language families.

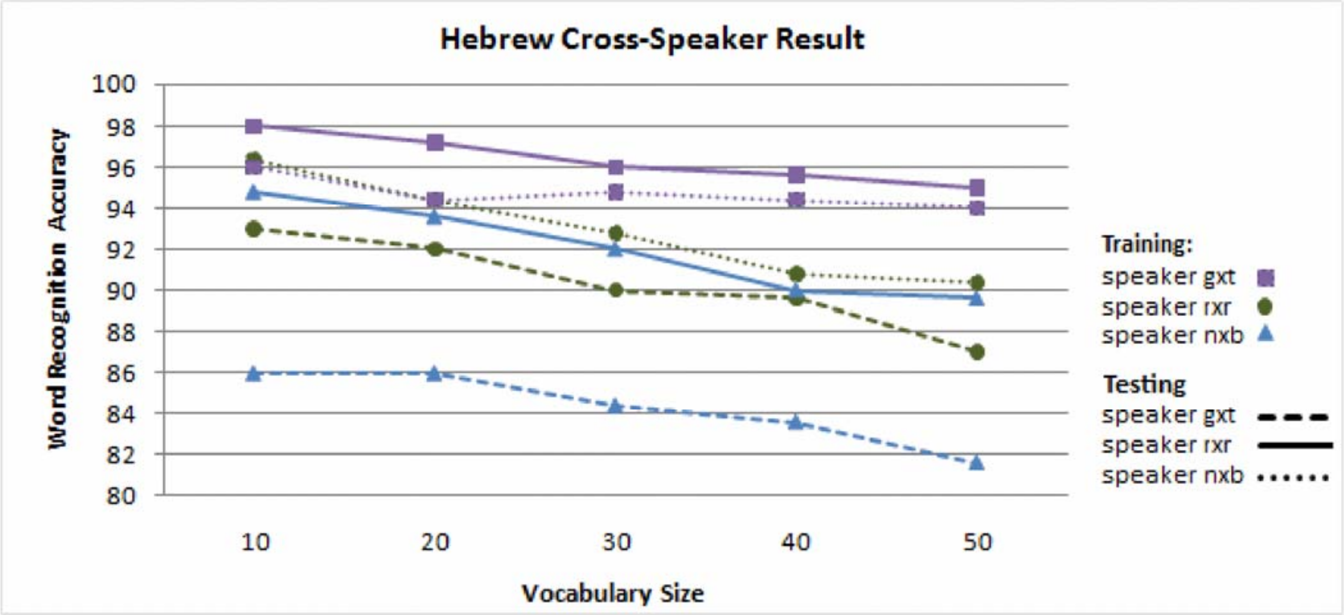

Figure 2. Cross-Speaker recognition accuracy for pronunciations trained on single speakers.

5.2.2 *Cross-Speaker Accuracy (single-speaker training)*

Next, we tested cross-speaker recognition accuracy: pronunciations trained on each speaker were tested on the two other speakers (Figure 2). Recognition accuracy varies noticeably based on the specific speakers used. While pronunciations trained on speaker gxt worked extremely well, and those trained with data from speaker rxr also performed satisfactorily, those from speaker nxb did not always do very well. Similarly, recognition accuracy on test speaker gxt's voice was consistently lower than that on the other two speakers. Speaker variations are a known phenomenon in speech recognition, and highlight the need to create robust pronunciations based on multiple speakers.

5.2.3 *Multiple Pronunciations per Word (cross-speaker, single-speaker training)*

Next, we probed the potential benefit of providing the recognizer with more than one pronunciation for each target word (Figure 3). Our pronunciation-generation method routinely generates a ranked list of pronunciations for each target word. In the experiments reported above we used only the top-ranked pronunciation in each such list. In this experiment, we compared this with giving the recognizer the top three alternatives for each target word. Even though this is an extremely simple method for selecting the number of pronunciations, Figure 3 shows that it does result in some further improvement in recognition accuracy when the vocabulary size is relatively large. This suggests that further improvement may be possible if we choose the number of

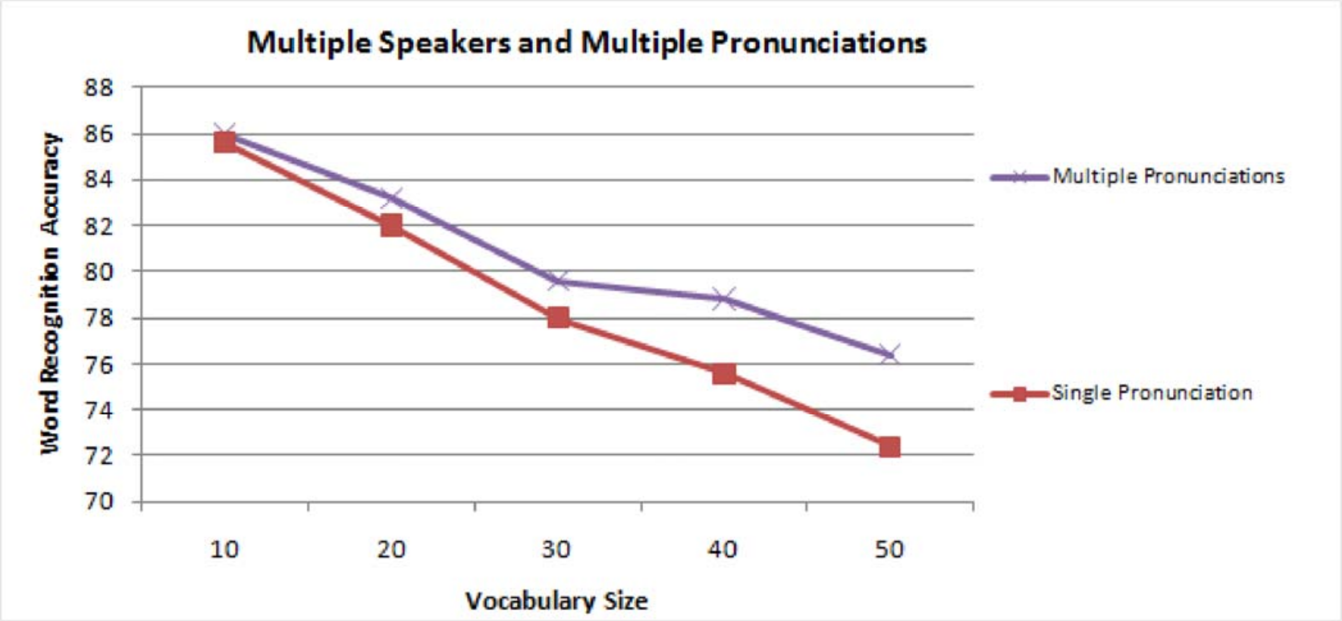

Figure 3. Recognition results for Hindi of a recognizer with a single pronunciation for each word, vs. one with multiple pronunciations per word.

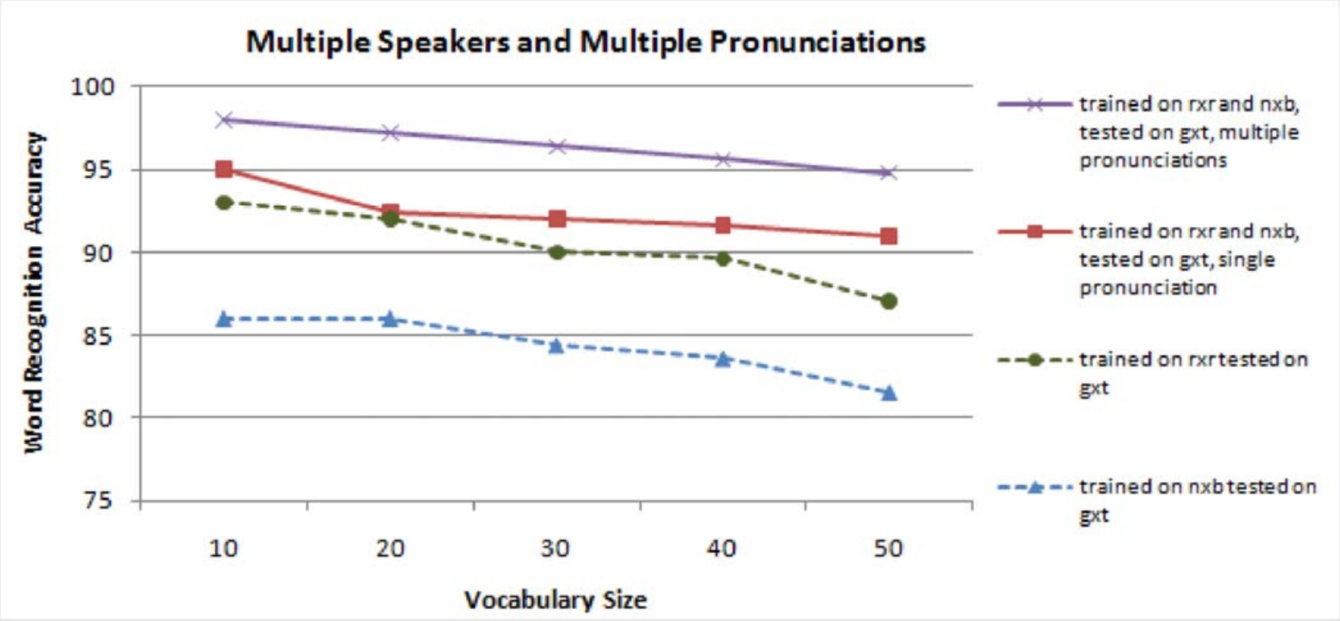

Figure 4. Hebrew recognition results, with single speakers generating single pronunciations, multiple speakers generating single pronunciations and multiple speakers generating multiple pronunciations.

pronunciations intelligently and individually for each target word. This has indeed shown to be the case in subsequent work (in preparation).

5.2.4 Multiple Pronunciations per Word (cross-speaker, multi-speaker training)

In this final experiment, we generated multiple pronunciations for each word by training on audio samples from two speakers, and tested their accuracy on the third speaker. We compared the results to those of pronunciations trained on single speakers, and also to recognition runs restricted to a single pronunciation per target word (Figure 4). These comparisons reveal that training on multiple speakers' voices result in more robust pronunciations, and re-confirm that allowing multiple pronunciations further improves accuracy (this time, across all vocabulary sizes).

6. CONCLUSION

The results from the last section present empirical confirmation that our method achieves high recognition accuracy for small vocabulary sizes without the involvement of any human experts, and with extremely meager language resource requirements. Modern, general-purpose speech recognition systems require hundreds of hours of net speech data – while our method requires only 10 minutes worth (~1 second per word, with 50 words, 5 repetitions per word, and 2 speakers per word, which gives 500 seconds). The clock time required to record the two speakers was an hour each. We know of no other techniques that yield that level of accuracy in speech recognition for resource-scarce languages. Moreover, our method yields pronunciations that consistently outperform those provided by linguistic experts. While other methods exist to create small vocabulary recognition capability, ours is the only one we know of that can achieve greater than 90% accuracy with such trivial resource requirements – and our experience in working with developing world NGOs shows that there are real limits on the amount of resources that can be allocated for such initiatives. Many spoken dialog applications become usable when the error rate drops below 5% -- this is already the case with our method when the number of input choices at any point in the application is limited to about 10 – typical of many useful information access applications. Furthermore, we have also shown that one can improve upon the quality of recognition achieved with our technique by expanding the training set size and the number of speakers for training, or mapping multiple pronunciations to a single word. Further 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 Afro-asiatic languages (Hebrew), the Niger-Congo languages (Yoruba), and the Indo-Aryan languages (Hindi); and the method yielded satisfactory results for all. There is a greater implication for the Yoruba and the Hindi test sets – these languages are used in developing regions of the world, and little 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 low literacy levels. We also plan to field-test recognizers built with our method in developing regions.

As per our description of the method's design in section 4, implementation of our method should not entail low-level modifications to a speech recognition engine of the source language - our design could be used with any recognition engine, including commercial, proprietary ones. An interesting future direction would be to test this method's effectiveness on different recognition engines.

We hope that other groups build on our work to improve recognition accuracy, and we welcome collaboration to create toolkits that could enable a completely turnkey solution for organizations in the developing world to create and use speech recognition capabilities for languages of their interest. We envision that this would enable the creation of speech-based applications that can target the needs of those with the least amount of resources available to them – low literate individuals for whom such technology may be their only option to interact with the digital world.

7. CORPORA STANDARDIZATION AND DATA AVAILABILITY

As part of our ongoing research we continue to collect small vocabulary, isolated-phrase, telephone-bandwidth multiple-speaker speech samples in a variety of languages. As of November 2010 we have collected recordings of 50-100 phrase standardized vocabularies in Mandarin, Yoruba, Hebrew, Hindi and Urdu, with 2-3 speakers per language and 5 samples per phrase per speaker. We plan to increase the breath and depth of this collection, and to record more South Asian and African languages in the near future. To encourage standardization of speech corpora for developing-world languages, we will make all our data available upon request to interested parties for research and development.

8. ACKNOWLEDGEMENTS

Partial support for the project was provided by the U.S. Agency for International Development under the Pakistan-U.S. Science and Technology Cooperation Program.

9. REFERENCES

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

[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.

[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.

[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.

[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.

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

[7] ITU. Measuring the information society: The ict development index. Accessed May, 2009.

[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.

[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.

[10] T. Schultz and A. Waibel. *Fast Bootstrapping of LVCSR Systems With Multilingual Phoneme Sets*. Proc. Eurospeech, pp. 371-374, Rhodes, 1997.

[11] T. Schultz and A. Waibel. *Adaptation of Pronunciation Dictionaries for Recognition of Unseen Languages*. Proc. SPIIRAS International Workshop on Speech Communication, St. Petersburg, Russia, October 1998.

[12] T. Schultz and A. Waibel. *Language Independent and Language Adaptive Large Vocabulary Speech Recognition*. Proc. Int. Conf. Spoken Language Processing (ICSLP), Sydney, 1998.

[13] T. Schultz, M. Westphal, and A. Waibel. The globalphone project: Multilingual lvcsr with janus-3. In *Proc. SQEL*, pages 20–27, 1997.

[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.

[15] J. Sherwani, N. Ali, S. Mirza, A. Fatma, Y. Memon, M. Karim, R. Tongia, and R. Rosenfeld. *HealthLine: Speech-based Access to Health Information by Low-literate Users*. In *Proceedings of ICTD 2007*, Bangalore, India, 2007.

[16] J. Sherwani and R. Rosenfeld. *Speech vs. Touch-tone: Telephony Interfaces for Information Access by Low-Literate Users*. In *Proceedings of ICTD 2009*, Doha, Qatar, 2009.

[17] 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.