Discriminative Pronunciation Learning for Speech Recognition for Resource Scarce Languages

Student
Hao Yee Chan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
haoyeec@andrew.cmu.edu

Advisor
Roni Rosenfeld
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
roni@cs.cmu.edu

0. ABSTRACT

We develop pragmatic solutions that create small vocabulary speech recognizers at a fraction of the cost and time that current packages require. Little to no expertise in speech recognition is needed in training a recognizer for any language, and only a few audio samples per word are required. Hence, these solutions are ideal for targeting languages that have a small or economically disadvantaged user base which are typically ignored by the commercial world. In particular, building on previous work, we design algorithms that utilize current speech recognition engines to generate pronunciations which map phonemes across languages. Using discriminative training techniques, we further improve the pronunciation generated, resulting in higher recognition accuracy.

1. INTRODUCTION

In developed countries today, speech recognition systems are ubiquitous. From basic spoken dialog systems (SDS) to complex Google voice search, speech technology can be found everywhere. However, of all the languages in the world, only languages spoken by economically developed nations are available in such systems. Commercial systems such as the Microsoft Speech Server produce high quality recognizers but suffer from the same lack of diversity. Open-source options such as the CMU Sphinx system offer an alternative but copious amounts of training data and deep knowledge of speech technology are basic prerequisites for such techniques.

Ironically, speech technology offers far more benefits to the developing world than to a mostly literate developed world [1, 2]. Moreover, compared to traditional touch-tone interfaces, speech-based interaction has been found to be more effective [3]. However, such interaction must be based on high quality automatic speech recognition (ASR) and well-designed user interface [4]. The problem lies with the development of such an ASR given the constraints of technical knowledge and training data. Such constraints are exacerbated by the conditions found in the populations of target languages, where literacy rates are low.

The aim of this study is to develop methods to build speech recognizers for languages where the speakers are mostly illiterate. Given these constraints, the speech recognizer should be the following:

- Language-independent
- Cost-efficient
- Easy to use without need for a technical background
- Accurate
- Relatively fast to run

2. RELATED WORK

Our work is based upon previous work, Speech-based Automated Learning of Accent and Articulation Mapping (SALAAM) [Sherwani, Fang] [5].

The Salaam approach aims to build small-vocabulary recognizers by using cross-language phoneme mapping using existing recognizers as a baseline. The approach removes the need for linguistic experts to provide transliteration of word types by fully automating the population of phonemes. Word types refer to a word or phrase in the target language, which is taken to be a single token for identification. Via a "divide-and-conquer" technique, the approach reduces the search space exponentially, resulting in an accurate yet fast speech recognizer.

The SALAAM approach achieved less than 15% word error rate for a vocabulary of 10 to 50 word types on Hebrew, which is promising.

3. METHOD DESIGN

3.1 Cross-Language Phoneme Mapping

Cross-language phoneme mapping uses acoustics models in the source language to represent corresponding phonemes in target language. A word type in the target language is defined by a sequence of phonemes in the source language. This sequence of phonemes is taken to be the correct pronunciation of the target word type. For example, the word "three" in Mandarin will be recorded as "S AA N" in the MSS U.S. English recognizer.

It is immediately clear that for target languages that have pronunciations which do not have corresponding phonemes in the source language would not be represented phonetically as compared to a linguistic expert. For example, the French word "Trois" has a voiced uvular fricative, which sounds like a mix between "K" and "H" in English. When encountering such difficulties, the sequence of phonemes that best matches the audio samples will be selected.

3.2 Filtering Approach

Given a new word type and all possible combinations of phonemes, the problem is to identify the best combination of phonemes as the correct pronunciation for the word type. A naïve approach would be to brute force all possible combinations. Even if the word type is known to be 5 phonemes long and we have only 37 different phonemes in the MSS U.S. English recognizer, we have an exponential search space of 37^5 combinations. It is clear that such an approach would be impractical. We would like to filter out poor phoneme sequences at every stage.

3.2.1 Previous SALAAM Method by Fang

Fang enumerated all combinations of 1 to 3 phonemes to seed his initial pronunciation. The number of word boundaries is fixed at 10. This means that we allow the recognizer to assume that each audio sample may consist of up to 10 words.

*/ */*/… **OR** **/*/*/… **OR** ***/*/*/…

where * denotes any phoneme and / denotes word boundaries.

The speech recognizer will return a set **A** of phoneme sequences. The top N results are then taken to be a set **B**. For each phoneme sequence in **B**, the first phoneme **ph1** is taken to be the potential first

phoneme in the final result. A new set of phonemes sequences is built for the next iteration by appending **ph1** to the combinations above.

ph1 */*/… **OR ph1** **/*/… **OR ph1*****/*/…

where * denotes any phoneme and / denotes word boundaries.

On this second iteration, the top K results from **ph1** */*/…, top K results from **ph1** **/*/… and top K results from **ph1****/*/*/… are taken. This is what Fang calls the divide-and-conquer method. The first two phonemes are then appended to the combinations and the cycle repeats.

We terminate the iterations when the best phoneme sequence returned remains the same for three consecutive iterations. This implies that the speech recognizer has decided that there are no additional phonemes that are present apart from what it has already detected. This final phoneme sequence is then taken to be the result.

3.2.2 Current SALAAM Method

Improving Previous Method

The initial seeding of phoneme sequence is similar. However, we take all results for the next iteration instead of only the top K. We are concerned that we might prematurely discard potential good phonemes sequences. For example, the K + i^{th} result from **ph1****/*/… might be better than all the results from **ph1** */*/… If we discard that result immediately, we might have already removed the best phoneme sequence.

In the second iteration, we build the phoneme sequences as follow:

ph1 */*/… **OR ph1** **/*/…

We removed the last phoneme sequence of 3 phonemes in order to accommodate the additional time taken by considering all results instead of just the top K, reducing the time taken by at least three times. Subsequent iterations are similar and the termination criterion remains unchanged.

Discriminative Training

There are two approaches to ASR systems. The goal is to identify the word type given an audio sample. One of them is the typical maximum-likelihood training used for ASR systems. Given an audio sample, the system tries to identify the phoneme sequence that best describes the sample, without regard for the vocabulary. The other way is to consider the other possible word types and identify the phoneme sequence that is closest to a word type in the vocabulary.

Previously, the SALAAM method attempts to find the best phoneme sequence for each word type. These sequences are independent from each other. Given a word, SALAAM will always produce the same phoneme sequence regardless of the vocabulary. Thus, this follows the maximum-likelihood training methodology.

We now improve the SALAAM method by getting a list of possible phoneme sequence per word. Our goal is to select a subset of phoneme sequences to minimize any conflicts between word types with similar pronunciations. Our concern is that while a given phoneme sequence is best for a given word type, the sequence might be recognized for a different word which sounds the same, resulting in unnecessary errors. If we had a less “correct” phoneme sequence which was still recognized for the

given word type but avoided being recognized for any other word type in the vocabulary, the word error rate would decrease.

We categorize errors into two groups:

Eager errors: an audio sample is recognized by a phoneme sequence which is not the correct word type. This phoneme sequence is then known to be "eager" for recognition.

Shy errors: an audio sample is not recognized by a phoneme sequence which is the correct word type. This phoneme sequence is then known to be "shy" or avoids recognition.

Based on varying the weights of the errors, ranking and confidence scores, we hope to achieve a good heuristic for filtering out badly performing phoneme sequences. Ranking refers to the order in which Microsoft Speech Server returns a list of recognized word types, while confidence scores refer to the confidence score of each recognized word type. While we expect the confidence score of a word type would correspond directly to its rank, this was not the case. It is possible that Microsoft Speech Server uses other criteria to rank their list of results apart from the confidence scores.

Our attempt was to remove all phoneme sequences with eager errors regardless if they were correctly recognized. We hoped that the alternative phoneme sequences would be recognized in turn when the errors were removed.

Another approach was to change the recognized phoneme sequence by sorting by confidence score or ranking and returning the top phoneme sequence.

4. METHODOLOGY AND RESULTS

4.1 Data Collection

A list of 100 word types was created, consisting of numbers, short commands, diseases, time and other common phrases used in the agricultural domain. A bilingual person in both English and the target language selects the most appropriate or commonly used translation of the word types. The translated list is then used for the recording.

The recording is done in a quiet location, through a landline or a cell phone, to replicate the various audio transformations done by telecommunication companies. For this purpose, we have created a SDS on the Tropo platform. Contributors can call in at a specified number or we can trigger the SDS to make an outbound call. The outbound call can be done at

http://salaamserver01.speech.cs.cmu.edu/salaam/set_values.php

Each contributor will read out the word types in order for 5 iterations because consecutive repetition of identical words may affect the pronunciation.

We currently run SALAAM on 5 samples from 5 contributors.

4.2 Evaluation and Results

Evaluation is carried out in this manner. The improved SALAAM method is applied to get a list of alternative pronunciations for the vocabulary. Discriminative training is then carried out, using audio samples from 4 speakers and leaving 1 out as the test speaker. The resulting list of pronunciations from discriminative training are then used to test the test speaker's audio samples. The word error rate

is then determined by the percentage of audio samples that were not identified correctly by the MSS speech recognition engine.

4.2.1 Baseline

With the improved SALAAM, we have a baseline where we select a speaker as the test subject. We train SALAAM on the data from the other 4 speakers and test it on the data from the test subject. By varying the vocabulary size and the number of alternative pronunciations, we get the following result.

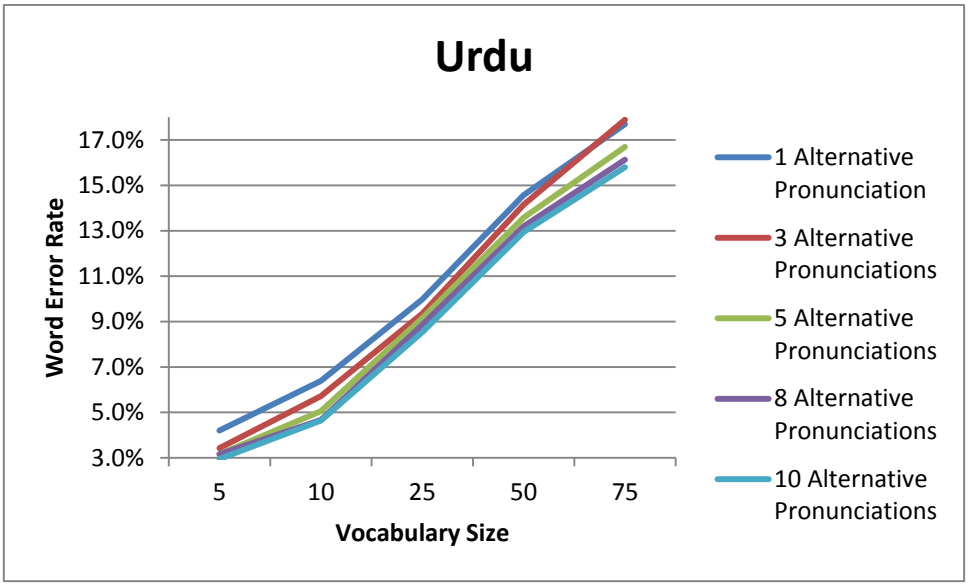

Figure 4.1 Recognition result on Urdu for improved SALAAM method with varying vocabulary sizes and alternative pronunciations

We see that increasing the number of alternative pronunciations will increase the accuracy of recognition. However, with 8 and 10 alternative pronunciations, the difference is no longer apparent. This is possibly due to eager errors which increase with the number of possible conflicts.

4.2.2 Basic Discriminative Training

Preliminary Training on 50 Word Types Picked Randomly

50 random words were chosen from the vocabulary of 100 words and discriminative training was then applied. This was repeated 10 times and the results are as shown.

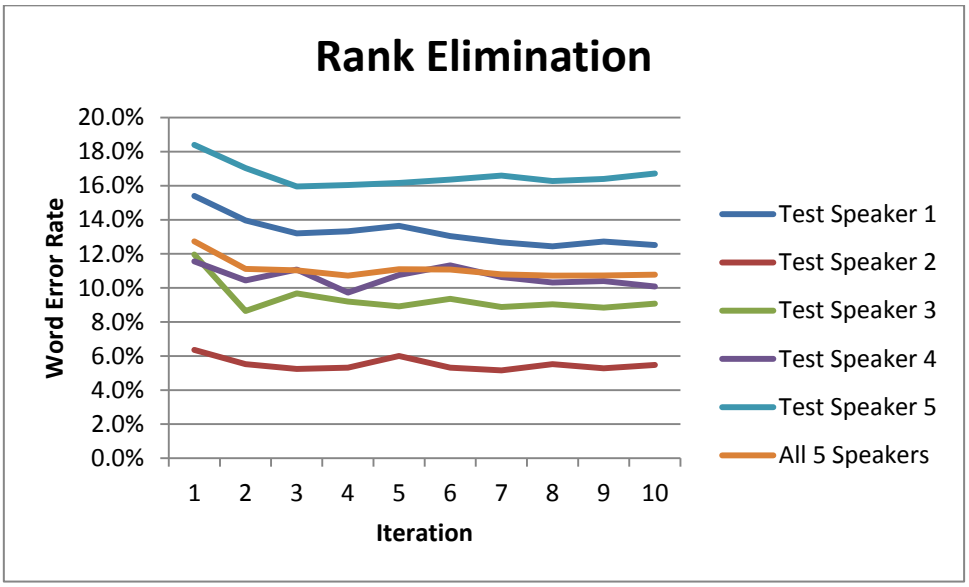

Figure 4.2 Recognition result on Urdu by removing all eager errors in rank order

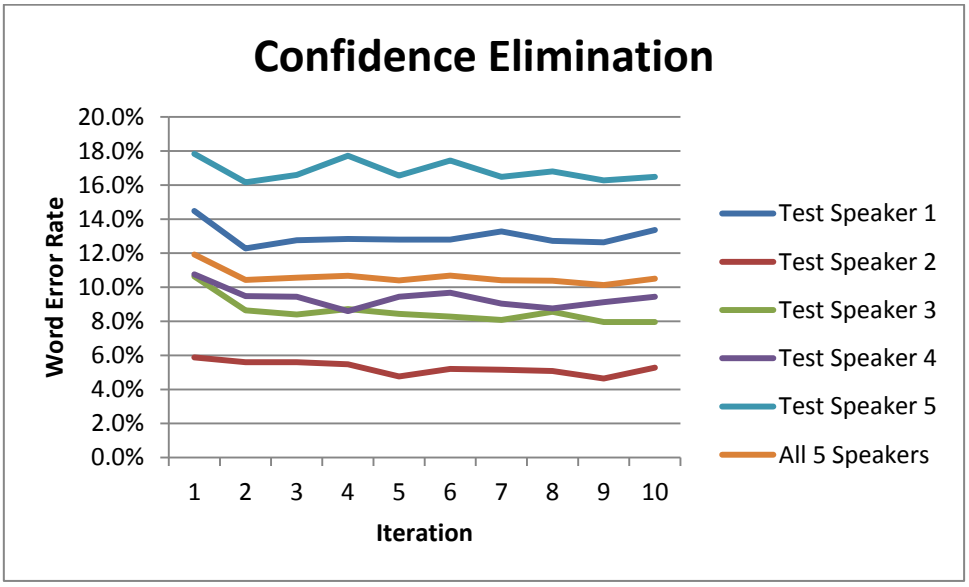

Figure 4.3 Recognition result on Urdu by removing all eager errors in confidence score order

Both methods are effective in reducing error rates. This indicates that we are on the right track. However, eliminating by rank seems to be more consistent.

Discriminative Training on Full Vocabulary

Now, we apply the training to all 100 word types and we get the following results.

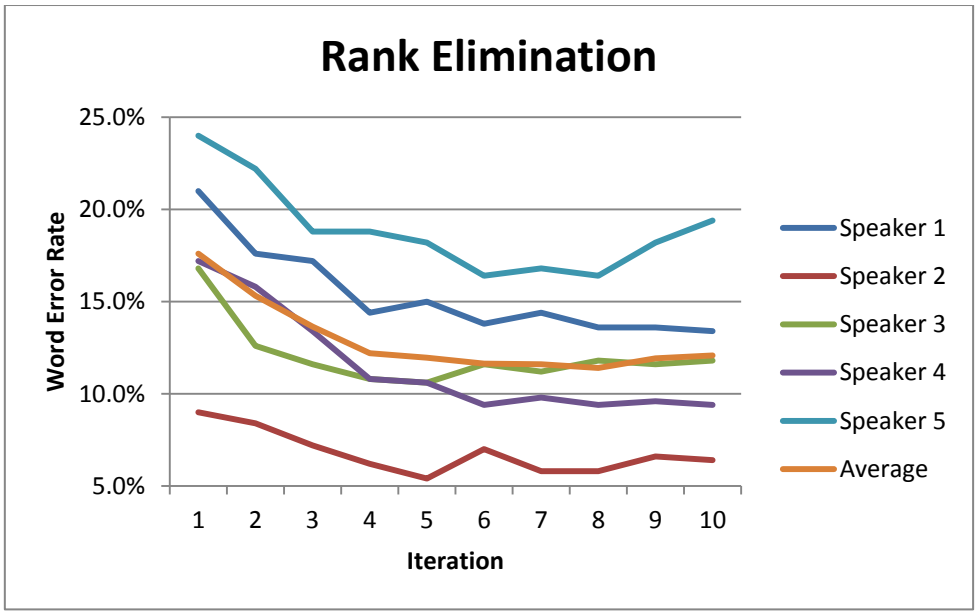

Figure 4.4 Recognition result on Urdu by removing all eager errors in rank order with full vocabulary

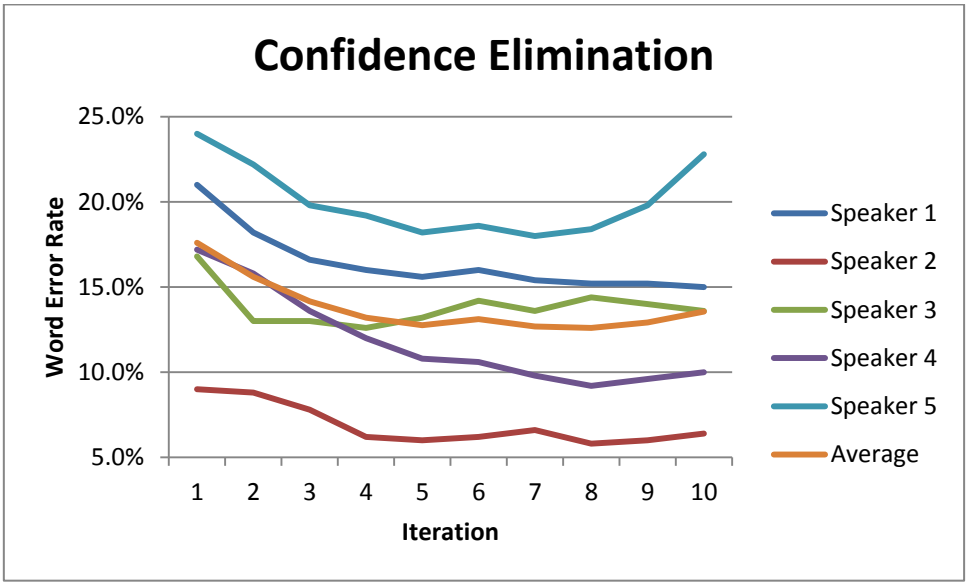

Figure 4.4 Recognition result on Urdu by removing all eager errors in confidence score order with full vocabulary

We note that eliminating by rank gives a slightly better result than eliminating by confidence score, as predicted. However, in both cases, we see a slight rise in error rates after the 7^{th} iteration and then a convergence towards the final word error rate for the final iterations or even an increase in word error rate. It is possible that further elimination after the 4^{th} iteration removes eager pronunciations that were used for recognizing correct words, resulting in the increase in error rates. It is encouraging that we manage to achieve below 15% word error rate even with the full set of 100 words.

4.2.3 Word Selection

Discriminative training gives us an N-Best list of identified pronunciations for each audio sample. So far, we have identified the word type of each audio sample by simply taking the top ranked pronunciation in the N-Best list and returning the word type that is described by that pronunciation. In other words, we have ignored the rest of the N-Best list as well as their confidence scores.

Top Pronunciation

We call this method of word selection "Top Pronunciation", which describes how we currently select the identified word type from an N-Best list.

Occurrences

We count the number of occurrences of a particular word type's pronunciations in the N-Best list. We identify the audio sample with the word type that has the highest number of occurrences.

Confidence Score of Word

For every pronunciation in the N-Best list belonging to a word type, we sum up its word confidence score for that word type. We identify the audio sample with the word type that has the highest confidence scores.

Confidence Score of Phrase

For every pronunciation in the N-Best list belonging to a word type, we sum up its phrase confidence score for that word type. We identify the audio sample with the word type that has the highest confidence scores.

Baseline Word Selection

We ran improved SALAAM on the full vocabulary with the different word selection heuristics.

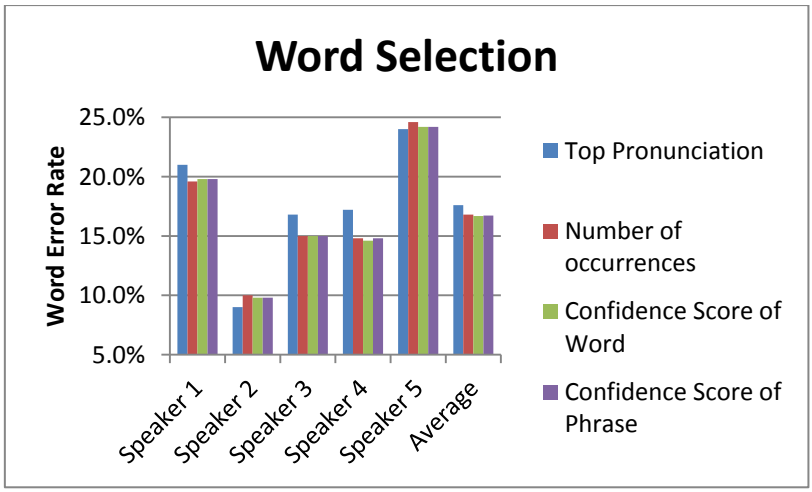

Figure 4.5 Recognition result on Urdu with improved SALAAM and word selection with full vocabulary

We see that on average all the word selection methods work better than the original Top Pronunciation. Since the methods, Confidence Score of Phrase and Confidence Score of Word, seem to perform equally well, we will only consider Confidence Score of Word and call it Confidence Score in future training.

4.2.4 Discriminative Training with Word Selection

We vary methods of elimination and word selection, taking the average from each result as to form the overall comparison.

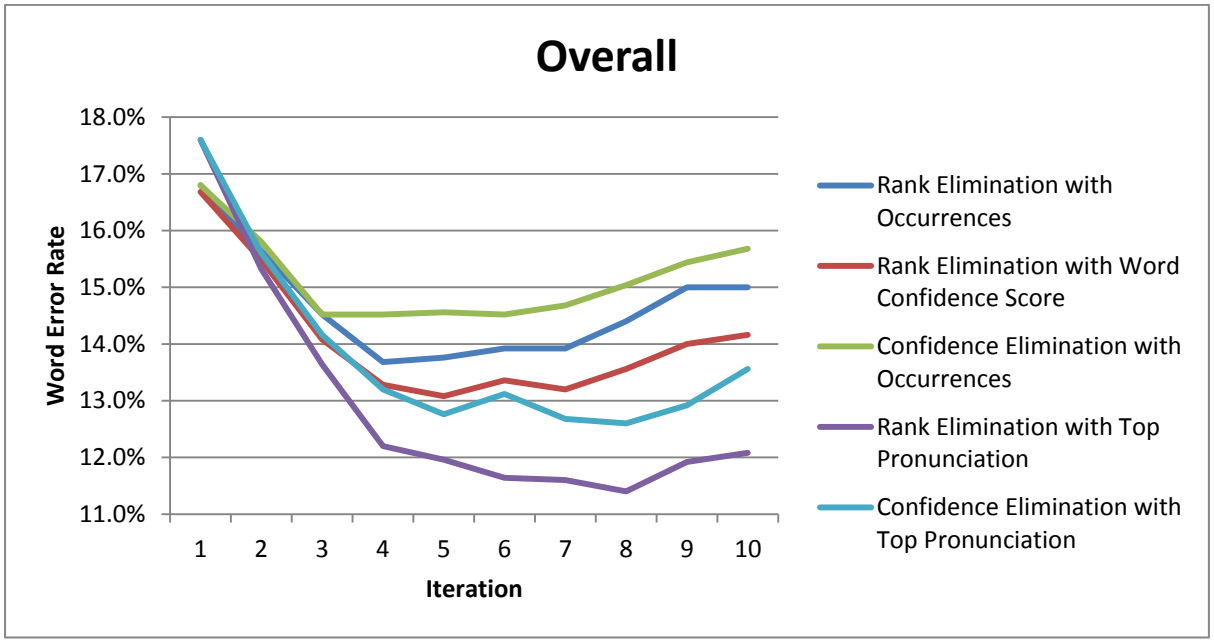

Figure 4.6 Recognition result on Urdu with various heuristics on full vocabulary

We note that our baseline results were correct; Top Pronunciation performs worst at the start. However, we are intrigued by its swift improvement over the next few iterations. It is clear that Top Pronunciation far outperforms the other word selection heuristics. Previously, we have seen that Confidence Elimination does not perform as well as Rank Elimination. However, with the appropriate word selection heuristic, it does much better than Rank Elimination. It is possible that word selection is a more significant heuristic than elimination. Overall trends remain the same, with word error rates dropping significantly over the first 4 iterations and then stabilizing over the 5^{th} to 7^{th} iterations and finally increasing after that.

From this result, our guess that MSS' ranking of pronunciations consists of more than just confidence score is likely to be justified because word selection based on ranking differs greatly from that based on confidence scores.

Our current procedure is to use Rank Elimination with Top Pronunciation and halt at the 8^{th} iteration.

5. CONCLUSION

Results show that discriminative training reduces word error rate significantly. While the original SALAAM method performs admirably well, the addition of discriminative training fine-tunes the resulting grammar to reduce word type identification conflicts such as eager errors.

Given that the heuristics are chosen to be independent of source and target languages as well as speech recognizer used, they are sufficiently generic to fulfill our stated objectives.

Although Urdu is the only language trained so far, Urdu falls into our targeted category of a resource scarce language and SALAAM is ready to be deployed for speech recognition applications in Pakistan.

We hope to develop a framework so that SALAAM can be deployed through a web browser interface to simplify interaction with the system. It is conceptually easier to use a web interface rather than load up a terminal or call a custom program to operate SALAAM. Already, audio samples can be uploaded for training purposes as described in section 4.1. This will allow speech recognition technology to reach parts of the world that will truly benefit from it.

5. REFERENCES

[1] E. Barnard and M. Davel and G. Huyssteen (2010), "Speech Technology for Information Access: a South African Case Study", *Proceedings of AAAI Artificial Intelligence for Development (AI-D'10)*.

[2] Plauche, M., Nallasamy, U., Pal, J., Wooters, C., & Ramachandran, D. (2006), "Speech Recognition for Illiterate Access to Information and Technology", *Proc. 115 International Conference on Information and Communications Technologies and Development, 2006.*

[3] Jahanzeb Sherwani, Sooraj Palijo, Sarwat Mirza, Tanveer Ahmed, Nosheen Ali, and Roni Rosenfeld (2009), "Speech vs. touch-tone: telephony interfaces for information access by low literate users", *Proceedings of the 3rd international conference on Information and communication technologies and development (ICTD'09)*, IEEE Press, Piscataway, NJ, USA, 447-457

[4] Patel, N. et al. (2009), "A Comparative Study of Speech and Dialed Input Voice Interfaces in Rural India", *Proceedings from CHI 2009 Conference, Boston*

[5] Fang Qiao, Jahanzeb Sherwani, and Roni Rosenfeld (2010), "Small-vocabulary speech recognition for resource-scarce languages", *Proceedings of the First ACM Symposium on Computing for Development (ACM DEV '10)*, ACM, New York, NY, USA, , Article 3 , 8 pages.