Large Vocabulary Continuous Speech Recognition for Urdu

Huda Sarfraz*, Sarmad Hussain*, Riffat Bokhari**, Agha Ali Raza**, Inam Ullah*, Zahid Sarfraz**, Sophia Pervez**, Asad Mustafa*, Iqra Javed*, Rahila Parveen*

ABSTRACT

This paper presents the development of acoustic and language models for robust Urdu speech recognition using the CMU Sphinx Open Source Toolkit for speech recognition. Three models have been developed incrementally, with the addition of speech data of up to two speakers per pass; one model using data from 40 female speakers only, one from 41 male speakers only, and one with both male and female speakers (81 speakers). This paper presents the current recognition results, and discusses approaches for improving these recognition rates.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Speech recognition and synthesis

General Terms

Your general terms must be any of the following 16 designated terms: Algorithms, Performance, Experimentation.

Keywords

Speech recognition, Sphinx, Urdu, Spontaneous speech.

1. INTRODUCTION

This paper describes the development of a speech recognition system for Urdu. Urdu has more than 100 million speakers in Pakistan, India, Middle East and Afghanistan [1]. Urdu is the national language and lingua franca of Pakistan. With a literacy rate below 50% [2], and an even lower rate for English literacy, the average Pakistani faces a double barrier of literacy and language while accessing information. Dialog systems present a solution for these barriers, and robust speech recognition is one of the essential components for developing such systems.

There has been some research in Urdu speech recognition, however, most of these efforts have been within limited context, e.g., using a small vocabulary, recognizing isolated words, recognizing in a noise-free environment and/or recognizing a single speaker's speech. For a practical dialog system, a speech recognizer should be able to recognize speech from an average speaker in a normal environment. This paper presents an investigation into creating robust speech recognition systems for Urdu. Robustness in speech recognition is described in [3] as "the need to maintain good recognition accuracy even when the quality of the input speech is degraded, or when the acoustical, articulatory, or phonetic characteristics of speech in the training and testing environments differ".

The work described in this paper is aimed at developing robust speech recognition systems for normal, everyday speech of male and female Urdu speakers, specifically of the Lahore suburban accent in office and home environments. With these constraints, the complexity of the problem is increased due to, (i) the large vocabulary size, (ii) spontaneity of the speech, (iii) environmental noise, and (iv) need for speaker independence.

The complexity level introduced due to the first two aspects is noted in [4], where a speech recognition task limited to a 100 word vocabulary and moderate spontaneity is categorized as easy, but with a 10,000 word vocabulary and high spontaneity is categorized as much more difficult.

Approaches for improving the robustness of speech recognition have been categorized into the four following areas by [5].

1. Robust speech features: focusing on developing features which are inherently less sensitive to noise/distortion.

2. Speech and feature enhancement: focusing on front-end signal or feature processing to suppress the impact of noise or distortion prior to speech recognition.

3. Recognizer model adaptation: focusing on adapting recognition models to noisy speech conditions.

4. Modified training methods: use of either noisy training data, mismatch between training/test data, or modifications which cause the trained models to be more effective for recognizing noisy speech.

The work presented here primarily used the fourth approach, i.e., used a corpus of training data recorded in a noisy environment. For acoustic model building and decoding, the CMU Sphinx Open Source Toolkit for Speech Recognition [6].

2. LITERATURE REVIEW

This section will present a brief survey of speech recognition technology for Urdu and related languages. A brief survey of open source speech technology is also given.

FIT'10, December 21-23, 2010, Islamabad, Pakistan.

* Affiliation: Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan; Email: firstname.lastname@kics.e.du.pk

** Affiliation: Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, Lahore, Pakistan; Email: firstname.lastname@nu.edu.pk

2.1 Urdu Speech Recognition Research

A single-speaker, large vocabulary, spontaneous Urdu speech recognition system using the Sphinx toolkit, which is a precursor to the work presented in this paper, is described in [7], where a combination of read and spontaneous speech training data is used for spontaneous speech recognition. The work in [8] presents another Urdu speech recognition system using the Sphinx toolkit, which is speaker independent but limited to a 52 isolated-word vocabulary. Urdu uses Arabic script for orthography, and [9] presents a speaker independent Arabic digit recognizer developed using the Sphinx toolkit, with emphasis on using an entirely Arabic environment (as opposed to previous systems using Romanized scripts) within Sphinx.

Some other research is also available on systems other than these Sphinx-based systems, e.g., [10] looks into pattern matching and acoustic phonetic modeling approaches for Urdu speech recognition and reports on a continuous Urdu speech recognizer with a 55-60% accuracy rate. The work in [11] presents a system for isolated digit recognition in Urdu and [12] uses a multilayer perceptron to recognize Urdu digits from a single speaker. Finally [13] presents an analysis of Urdu digits to be used for Urdu digit recognition.

Work has been done on speech recognition system for Hindi, which is very similar to spoken Urdu. The system gives a word accuracy of 75-95% for a 65,000 word vocabulary [14].

More generally, [15] gives some practical guidelines on building robust acoustic models using the Sphinx and HTK toolkits.

2.2 Speech Recognition Technology

This section will look at some of the open source solutions available for speech recognition problems.

The CMU Sphinx open source speech recognition toolkit has been used in the work presented in this paper [6]. The following components are available in the toolkit.

1. PocketSphinx: lightweight recognizer library, focusing on speed and portability
2. SphinxBase: support library
3. Sphinx4: adjustable, modifiable recognizer
4. CMUclmtk: language model tools
5. SphinxTrain: acoustic model training tools
6. Sphinx3: decoder for speech recognition research

Acoustic models built using SphinxTrain can be used by any of the decoders. Several tutorials are available, including tutorial projects, and training data is also available for English speech recognizers for use with Sphinx.

The Hidden Markov Model Toolkit (HTK) [16] is a portable toolkit for building and manipulating HMMs used primarily for speech recognition. It consists of a set of library modules and tools for speech analysis, HMM training, testing and result analysis. Extensive documentation is available, including tutorials and training data for English. The toolkit is available in source form but there are some licensing restrictions.

Julius [17] is a high performance large vocabulary continuous speech recognition decoder using n-grams and context dependent HMMs, developed for Japanese speech recognition. It uses standard formats for compatibility with other open source speech recognition toolkits such as those described in this section.

Speech recognition resources are also available through the Institute for Signal and Information Processing (ISIP) Internet-Accessible Speech Recognition Technology Project [18].

3. TRAINING AND TESTING METHODOLOGY

An 82 speaker speech corpus [19] including about 45 hours of speech from 42 male and 40 female speakers was used to train and test models using the Sphinx speech recognition toolkit. SphinxTrain was used to train the acoustic models and the Sphinx3 decoder was used for testing purposes. The CMU Statistical Language Modeling (SLM) Tookit [20] was used to prepare language models for decoding. This section gives some details about the training and testing process.

3.1 Training and Test Data Preparation

The speech corpus for the training and testing was developed as described in [19]. Speech data was recorded in wav format at 16 kHz. This transcribed, in Urdu script, was then converted to the format required by Sphinx using the Sphinx Files Compiler described in [7] and a transcription lexicon. The input required by SphinxTrain to build the acoustic models included:

1. A set of transcribed speech files
2. A dictionary file, containing transcriptions for all the words in the vocabulary
3. A filler dictionary file, containing entries for all non-speech sounds, e.g., vocalic pauses, throat clearing etc.
4. A phone file, including all the phones used in the transcriptions

The transcription lexicon was used with the Sphinx Files Compiler in order to generate phonemic transcriptions from Urdu orthography automatically. This transcription lexicon already included transcriptions of a base set of words, and suggested transcriptions for new words were auto-generated by the Sphinx Files Compiler, reviewed and added to the transcription lexicon as the data was processed.

The speech transcriptions are also used for language model building using the SLM toolkit [20].

Speech data for each speaker was prepared individually, and a training/testing data merging tool was developed in order to create training sets with different combinations of speakers, as needed.

3.2 Acoustic and Language Model Training

Acoustic and language models were created and tested for each individual speaker in the speech corpus, and for combinations of speakers. Individual speaker model building and testing allowed for easier removal of errors, and also gave indications about whether the addition of a particular speaker's data would cause problems in a cumulative set. Individual training and testing sets were built for all 82 speakers.

Two separate male and female speaker data training and testing sets were then created cumulatively. For the first 20 female speakers, one speaker set was added and tested with each pass. The same was done for the male speakers. After the addition of 20 speakers to each cumulative set, the change in results became somewhat predictable and the remaining speakers were added two at a time for each pass to speed up the process.

After the female (40 speakers) and male (42 speakers) training and testing sets were complete, they were also combined to determine which type of models (those trained with a single gender speaker data, and those trained with speech from both genders) performed better.

Default acoustic model training configuration was used as given by SphinxTrain and adjusted as necessary with the changes in speech data, as per the guidelines in the Sphinx3 manual [21]. For all the training sets, the HMM types were set to continuous, as recommended for use with Sphinx3 decoding. 3 state HMMs with a no-skip topology were used, as recommended for noisy data. The number of tied states (also referred to as senones in Sphinx documentation) was set to 1000 for the initial individual sets, and adjusted as recommended in the Sphinx3 manual as per the amount of speech data, shown in Table 1 [21]. States are shared to cater to data insufficiency problems in HMM states [22]. For further theoretical and practical details of HMM topology, which are beyond the scope of this paper, please see [23].

Table 1. Recommended number of tied states in the Sphinx3 manual [21].

Amount of training data (hours)	No. of tied states
1-3	500-1000
4-6	1000-2500
6-8	2500-4000
8-10	4000-5000
10-30	5000-5500
30-60	5500-6000
60-100	6000-8000
Greater than 100	8000 are enough

Throughout the preparation of the individual and cumulative sets, the configuration was set as described above, and the number of tied states was only changed and tested for the final few sets. The testing was only done for a limited number of variations because, with about 40 hours of training data for the final sets, training of acoustic models would take over six hours on a Dell PowerEdge T100.

The SLM toolkit version 2 [20] was used to create corresponding language models for each training set. Trigram models with Witten Bell discounting were created for all sets.

Some sets were also re-trained and tested after adding more training data and after reviewing and refining transcriptions.

Reviewing of a transcription set could reveal alignment errors (between the speech files and transcriptions) and also transcription errors. Refining of transcriptions involved insertion of diacritical marks in the Urdu orthography to disambiguate pronunciation issues. Diacritical marks are used in Urdu script to indicate vowel sounds. In written script, many diacritical marks are not needed because readers can identify the word due to context. During the manual Urdu transcription, minimal diacritical marks were transcribed for all words which could possibly be interpreted as two different spoken words. During the refining process, any such sets of words where the diacritical marks had been missed were identified. The proper diacritical marks were then inserted, and the phonemic transcription generated accordingly. This problem can be illustrated by a simple example of the Urdu words shown in Table 2.

Table 2. Diacritic insertion for Urdu script.

Urdu script	Urdu script with diacritics	Phonemic transcription	Meaning
ا س	اُس	ʊ s	That
ا س	اِس	ɪ s	This

The refining process also included fine-tuning of the transcriptions to match the pronunciations of the speakers. In essence, this meant converting the phonemic transcriptions into phonetic ones, to better match the spoken data. As explained in [7], only phonemic transcriptions were done during the first pass, as phonemic transcription can be done through a fast semi-automated process. This refinement process was conducted for some speaker sets to determine the impact it had on the results.

3.3 Decoding

The Sphinx3 decoder was used for testing the models, and default decoding configuration was used for most of the intermediate sets, which included a language weight of 23, a beam width value of 1e-120 and a word beam value of 1e-080. For some of the training and testing sets, the language weight was varied between 5 and 26, and beam width value was varied between 1e-100 and 1e-900. The variations were tested in some of the initial individual sets, and in the final cumulative sets.

4. Results

Some of the significant results of the tests described in the previous section are presented here.

Table 3 shows the results of the best and worst performing individual speaker sets, trained with 1000 tied states, and decoded with a language weight of 23 and beam width of 1e-120 (default values for Sphinx3).

Test results for a male speaker set, as presented in [7], are also include for comparison purposes, although this set was recorded in noise-less environment. These results show the significant impact of simply increasing amount of training data per speaker.

Table 3. Results for individual speaker sets.

	No. of training utterances	Word error rate
Male set (without noise)	3174	29.1%
Best male set	547	39.8 %
Worst male set	153	98.1 %
Best female set	517	54.2 %
Worst female set	156	95.4 %

Table 4 shows some key results for multiple speaker sets. It was not possible to conduct tests with all combinations of all values, because as noted earlier, the training process could take more than 6 hours with the amount of training data being used. The decoding process also slowed down considerably with the increase in testing data. So, value changes that were showing more potential for improvement during preliminary investigations were examined in more detail. An exhaustive process, however, was not undertaken. The beam width value for all displayed results is 1e-120. Some tests were conducted with varied beam width values, but no significant changes were noted. The results of the best test run of the combined male data set consisting of 41 speakers is shown in the first row. This set had a vocabulary size of 12,098, and included 18,835 training utterances. The acoustic models were trained using 5250 tied states, and a language weight of 23 was used during decoding to yield an error rate of 60.2%. Similarly, the last row shows the worst test run of the combined male and female data set consisting of 81 speakers.

Table 4. Results for multiple speaker sets.

	Voc. size	No. of training utterances	Tied states	LW	Word error rate
Best 41 M	12098	18835	5250	23	60.2
Worst 41 M	12098	18835	1000	23	64.9
Best 40 F	10981	11173	1000	17	65.6
Worst 40 F	10981	11173	1000	25	78.9
Best 81 M&F	14445	30983	5250	23	68.8
Worst 81 M&F	14445	30983	1000	23	79.0

Figure 1 shows the notable improvement in word error rates for the data sets that were re-trained and tested after the transcription data had undergone a review and refinement process. The word error rates drops significantly for two speakers (speaker 2 and 3) and there is a minute drop in word error rates for speakers 1 and 4. The bulk of the review and refinement process for which the results are shown consisted of diacritic insertion where needed.

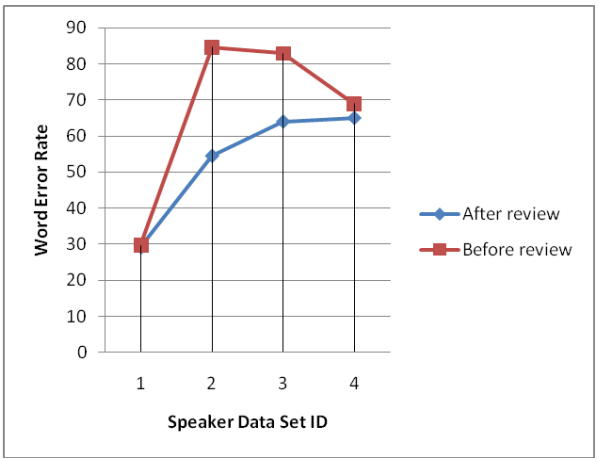

Figure 1. Decrease in error rates after transcription review and refinement.

Only those data sets have been shown where all other configurable parameters (language weight, amount of data etc.) were identical. However, the change is still inconsistent, possibly due to the variation in speech data for each speaker, e.g., some speakers spoke faster than others, some more fluently etc. This effect, however, has not been analyzed quantitatively.

5. DISCUSSION AND CONCLUSION

The results section shows that better results can be achieved by increasing the training data and refining the transcriptions.

One notable problem encountered during the refinement process was the insertion of diacritics for words adopted from the English language. All English words occurring in the vocabulary were transcribed using Urdu orthography. Urdu diacritics can cater to all Urdu vowels, but the existence of the vowel ɛ is questionable in Urdu [24]. In cases where an English word using the vowel ɛ formed a minimal pair with an Urdu or an adopted English word using the vowel æ, it was not possible to distinguish between the two using Urdu orthography, which was being used to generate phonemic transcriptions automatically. This can be illustrated using the example of the English words “bed” and “bad” both of which were spoken by speakers during speech acquisition for the corpus [19]. In Urdu script, either of these two can be differentiated from “bead” by inserting a zabr on the first consonant. However, the zabr causes the word to be interpreted as both “b ɛ d” and “b æ d”, with ambiguity resolved easily through context during reading. In order to enable the Sphinx Files Compiler to differentiate between the two, numbers were appended to the Urdu orthographic form. This convention was adopted from the format used in Sphinx input files for pronunciation variations, and was thus conveniently integrated into the rest of the process. There were similar problems with some other minimal pairs occurring due to usage of English words.

The Sphinx3 manual [21] also notes the importance of entering all relevant noise models in the training data. The results presented here only account for breath sounds and lumps all vocalic non-speech sounds into one category for simplicity. Including models for all types of noise present in the training data, updating transcriptions accordingly should also have a significant impact on the results.

Last but not least, further investigation into the training and decoding parameters will also yield better results. This will be taken up further in the future.

6. ACKNOWLEDGEMENTS

This work was carried out at the Center for Research in Urdu Language Processing (www.crulp.org), National University of Computer and Emerging Sciences, Lahore (www.nu.edu.pk) in collaboration with Carnegie Mellon University (www.cmu.edu), and was funded by the HEC-USAID Pak-US Joint Academic and Research Program (www.hec.gov.pk) and the PAN Localization Project (www.panl10n.net).

7. REFERENCES

[1] M.P. Lewis (ed.). Ethnologue: Languages of the World, Sixteenth edition. Dallas, Tex.: SIL International, 2009. Online version: http://www.ethnologue.com/.

[2] Government of Pakistan, Statistics Division, Federal Bureau of Statistics. Pakistan Statistical Yearbook 2009. www.statpak.gov.pk/depts/fbs/publications/yearbook2009/yearbook2009.html, accessed July 2010.

[3] R. Cole (ed). Survey of the State of the Art in Human Language Technology. Cambridge University Press and Giardini, 1997.

[4] A. Nakamura, S. Matsunaga, T. Shimizu, M. Tonomura, Y. Sagisaka. Japanese Speech Databases for Robust Speech Recognition. Proceedings of the ICSLP'96. Philadelphia, PA, pp.2199–2202, Volume 4.

[5] J. H. L. Hansen, R. Sarikaya, U. Yapanel and B.L. Pellom. Robust Speech Recognition in Noise: An Evaluation Using SPINE Corpus. Eurospeech '2001, Aalborg, Denmark, September 2001.

[6] CMU Sphinx Open Source Toolkit for Speech Recognition Project by Carnegie Mellon University, http://cmusphinx.sourceforge.net/, accessed July 2010.

[7] A. Raza, S. Hussain, H. Sarfraz, I. Ullah, Z. Sarfraz, "An ASR System for Spontaneous Urdu Speech". Submitted to O-COCOSDA 2010.

[8] J. Ashraf, N. Iqbal, N.S. Khattak and A.M. Zaidi. Speaker Independent Urdu Speech Recognition Using HMM. In Proceedings, Natural Language Processing and Information Systems: 15th International Conference on Applications of Natural Language to Information Systems, Cardiff, UK. June 23-25, 2010.

[9] H. Satori, H. Hiyassat, M. Harti and N. Chenfour. Investigation Arabic Speech Recognition using CMU Sphinx System. The International Arab Journal of Information Technology, Vol. 6, No. 2, April 2009.

[10] M.U. Akram and M. Arif. Design of an Urdu Speech Recognizer Based Upon Acoustic Modeling Approach. In proceedings, IEEE INMIC 2004, pp. 91-96.

[11] S. Azam, M. Mansoor, Z.A. Shahzad, M. Mughal and S. Mohsin. Urdu Spoken Digits Recognition Using Classified MFCC and Backpropagation Neural Network. In proceedings, IEEE Computer Graphics, Imaging and Visualization, 2007.

[12] A. Ahad, A. Fayyaz, T. Mehmood. Speech Recognition Using Multilayer Perceptron. In proceedings, Student Conference, ISCON, IEEE, 2002.

[13] S.K. Hasnain. Recognizing Spoken Urdu Numbers Using Fourier Descriptor and Neural Networks with MATLAB. Second International Conference on Electrical Engineering, University of Engineering and Technology, Lahore, Pakistan. 25-26 March 2008.

[14] C. Neti, N. Rajput and A. Verma. A Large Vocabulary Continuous Speech Recognition System for Hindi. In proceedings, 3rd IEEE Workshop on Multimedia Signal Processing, Copenhagen, 2002.

[15] J. Kacur and G. Rozinaj. Practical Issues of Building Robust HMM Models using HTK and Sphinx Systems. In Speech Recognition, Technologies and Applications. F. Mihelic and J. Zibert. I-Tech (ed.), Vienna, Austria. November 2008.

[16] HTK, http://htk.eng.cam.ac.uk, accessed July 2010.

[17] Julius, http://julius.sourceforge.jp/en_index, accessed July 2010.

[18] Institute for Signal and Information Processing (ISIP) Internet-Accessible Speech Recognition Technology Project, www.isip.piconepress.com/projects/speech, accessed July 2010.

[19] H. Sarfraz, S. Hussain, R. Bokhari, A.A. Raza, I. Ullah, Z. Sarfraz, S. Pervez, A. Mustafa, I. Javed and R. Parveen. Speech Corpus Development for a Speaker Independent Spontaneous Urdu Speech Recognition System. O-COCOSDA 2010.

[20] The CMU Statistical Language Modeling (SLM) Toolkit. www.speech.cs.cmu.edu/SLM_info.html, accessed July 2010.

[21] Manual for the Sphinx3 Recognition System. www.speech.cs.cmu.edu/sphinxman, accessed July 2010.

[22] Robust Group's Open Source Tutorial: Learning to Use the CMU Sphinx Automatic Speech Recognition System. www.speech.cs.cmu.edu/sphinx/tutorial.html, accessed July 2010.

[23] X. Huang, A. Acero, H.W. Hon. Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall, New Jersey, 2001.

[24] S. Hussain. Letter-to-Sound Conversion for Urdu Text-to-Speech System. In proceedings, Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004, Geneva, Switzerland, 2004.