Automatic Urdu Speech Recognition using Hidden Markov Model

Asadullah[1], Arslan Shaukat[1], Hazrat Ali[2], Usman Akram[1]
asadullah73@ce.ceme.edu.pk,arslanshaukat@ceme.nust.edu.pk,
hazratali@ciit.net.pk, usman.akram@ceme.nust.edu.pk
[1]National University of Sciences and Technology (NUST), H-12 Islamabad, Pakistan
[2]Department of Electrical Engineering, COMSATS Institute of Information Technology, Abbottabad, Pakistan

Abstract—**In this paper, we present an approach to develop an automatic speech recognition (ASR) system of Urdu isolated words. Our experimentation is based on a medium vocabulary speech corpus of Urdu, consisting of 250 words. We develop our approach using the open source Sphinx toolkit. Using this platform, we extract the Mel Frequency Cepstral Coefficients (MFCC) features and build a Hidden Markov Model to perform recognition task. We report percentage accuracy for two different experiments based on 100 and 250 words respectively. Experimental results suggest that better recognition accuracy has been achieved with this approach, as compared to the previous results reported on this corpus.**

Keywords-Automatic speech recognition, Hidden Markov Model, Mel-Frequency Cepstral Coefficients, Urdu words recognition.

I. INTRODUCTION

Speech is the most important mode of communication amongst humans as well as between humans and machines. Recently, advances in automatic speech recognition (ASR) techniques have enabled machines to effectively communicate with humans. English remains to be the most widely spoken languages of the world. While ASR research work prevails for English, gaining much attention from the research community, very less progress has been made on the speech recognition task of Urdu language.

Urdu is one of the largest spoken languages in the world and is also the national language of Pakistan. Urdu phonetics and phonology differs widely from English language. Speech dataset for Urdu is a fundamental requirement for any development on Urdu ASR. This research work is based upon the Urdu dataset designed in [1]. The dataset is a medium scale vocabulary of Urdu words. Unlike previous work in [2] [3], which has reported results on a subset of the dataset, we use all the 250 words in the data set. For feature set, we extract the Mel Frequency Cepstral Coefficients (MFCC), as these features have proven to be most effective for ASR. Our ASR framework is based on the widely used Carnegie Mellon University Sphinx 4 Toolkit (CMU Sphinx) [4] which builds Hidden Markov Model (HMM) to perform the recognition[1]. We compare our results with the previous work on this dataset while retaining similar experimental setup for fair comparison wherever possible. We observe that the recognition accuracy for our framework is higher than the previous accuracy reported in the literature on the same dataset.

Rest of the paper is organized as below. In section II, we present an overview of the related work on Urdu speech recognition. In section III, experimental setup and methodology have been mentioned. We report the results and discuss the recognition accuracy for various words in Section IV. Section V presents our conclusions.

II. RELATED WORK

Scientists have done lot of research work in ASR especially for English language, but there has been very less work in Urdu due to the unavailability of resources, data sets and lack of interest of researchers. An initial framework for Urdu ASR was proposed by Akram & Arif [5]. The accuracy of this system is in the range of 55%. Azam et al [6] have presented isolated digits Urdu ASR system using Artificial Neural Networks (ANN). This system is limited to only speaker dependent speech utterance. Ahad et al [7] developed Urdu digits recognition framework using multilayer perceptron (MLP) and have reported higher accuracy. This, however, is limited to digits recognition utterance by single speaker only. Husnain [8] has reported Automatic Urdu Digits Recognition System using feed forward Neural Network with high accuracy. The dataset contains speaker independent digits with 15 different speakers. Raza et al [9] have contributed to the development of Urdu ASR System. They developed corpus for Urdu ASR, which was context independent and phonetically rich and balanced. A more promising work has been done by Ashraf et al [10] using Urdu Isolated Words. They used CMU Sphinx Toolkit and MFCC features in their work.

While the work, mentioned in the literature above reports achievements on the Urdu ASR task, they lack information on free corpus and do not point out corpus resource, which may be then adopted and used by others for fair comparison. In our work, we use the corpus developed by Ali et al [1], which is freely available for future research work. Ali et al [1] have presented a speaker independent corpus, which contains 250 most popular words of Urdu Language.

III. EXPERIMENTAL METHODOLOGY

In this section, details about our proposed experimental methodology are mentioned. This includes the corpus that we have used, features that have been extracted and CMU Sphinx toolkit.

A. *Urdu Speech Corpus*

The Urdu speech corpus used in our experimentation is developed by [1]. The dataset contains 250 isolated words uttered by ten speakers, out of which eight speakers are male and two speakers are female. The average length of an audio file is 0.5 seconds and the average file size is 16kb. The recording was done in a noise free studio using Sony Linear PCM Recorder at a sample rate of 44100 Hz and then the wav files are converted to mono (with single channel) at a sampling rate of 16000 Hz. Some attributes of this corpus are mentioned in Table I.

Table I: Speaker attributes for the Urdu corpus used in the experiment.

Speakers	**Gender**	**Nature**
Speaker 1	Male	Non native
Speaker 2	Male	Non native
Speaker 3	Male	Non native
Speaker 4	Female	Native
Speaker 5	Female	Native
Speaker 6	Male	Non native
Speaker 7	Male	Non native
Speaker 8	Male	Non native
Speaker 9	Male	Non native
Speaker 10	Male	Non native

B. *Mel Frequency Cepstral Coefficients (MFCCs)*

MFCCs are the most widely used features for speech recognition task. This feature set is based upon the human perception of hearing. As speech is produced by human vocal tract, therefore vocal tract acts like a filter for speech production. The envelope of speech produced by human vocal tract is a representation of the short-term power spectrum of speech. MFCCs tend to determine the envelope of the speech. Therefore, MFCCs are used as features for speech recognition. The key steps for MFCC extraction are outlined in Figure 1. In our work, we calculate the MFCCs using Sphinxtrain**,** the Sphinx supplementary package.

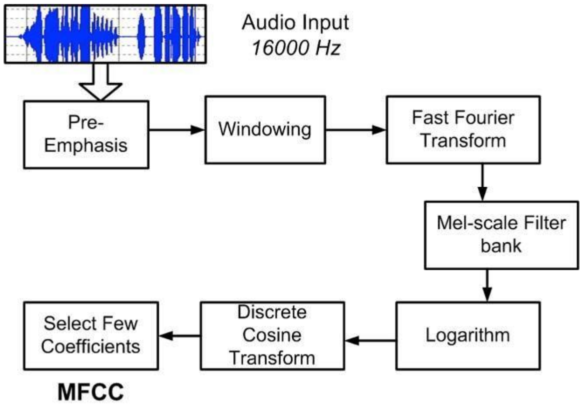

Figure 1: MFCC extraction procedure

C. *CMU Sphinx Overview*

Sphinx is a modular, flexible and pluggable toolkit developed by CMU, HP, MIT and Sun Microsystems for Automatic Speech Recognition research using HMM [11-14]. Most of the previous toolkits were based upon a single approach. Baker developed HMM for its own system Dragon [11]. Earlier versions of sphinx were based upon discrete HMM [12], Semi-continuous HMM [13] and continuous HMM [14]. Others systems used lex tree for N-gram language models [15].

Figure 2 shows the architecture of Sphinx. Sphinx can truly be regarded as a modular tool, thus providing flexibility towards speech recognition research. It has three main modules; Front End, Linguist and Decoder.

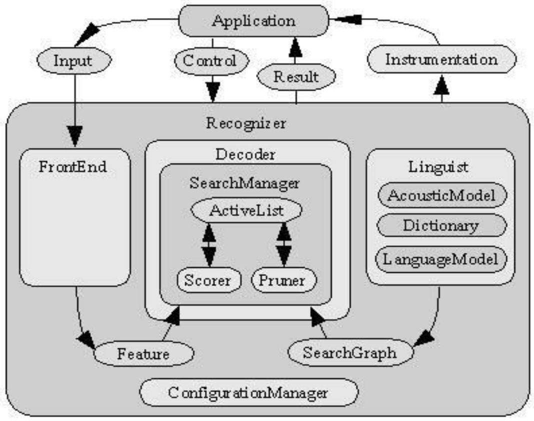

Figure 2. Sphinx Architecture (Courtesy: Walker, *et al.*, [16])

1) The Front End

The job of Front End is to transform an input audio/speech signal to a sequence of features set. Figure 3 show that this module contains a chain of multiple data processing units. The last data processor produces an output feature vector, which is used for decoding purposes.

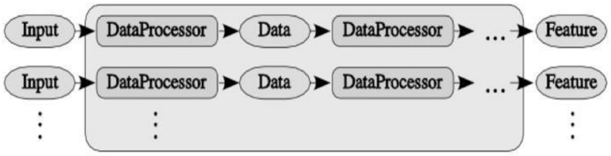

Figure 3: Sphinx Front End (Courtesy: Walker, *et al.*, [16])

2) The Linguist

The linguist is a pluggable module and its parameters can be fine tuned for system performance by using Sphinx configuration manager. It can produce search graph by using words from language model, sub phonetic units from acoustic model and pronunciation from dictionary.

Language Model (LM): This model contains the probability of the words used in the language. CMUCLTK tool is used to create Language model. *Dictionary***:** The pronunciation of words found in a language model is provided by dictionary. These pronunciations can then be used to break the words into a sequence of sub-word units.

Acoustic Model (AM): This model uses HMM to produce a sequence of observations for speech signal. The sequence of outputs/observations from HMM can be used to score against the features, which is produced by Sphinx Front End.

Sphinxtrain is used for producing the Acoustic Model from training data.

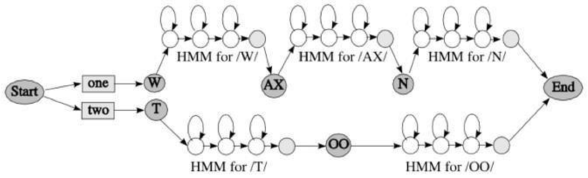

Figure 4: Search Graph

*Search Graph***:** The search graph is used in decoding process. As shown in Figure 4, it is a directed graph, which contains search states/nodes. The arrow represents the transition probability from one state to another state. Words in rectangle are taken from language model, sub-word units in highlighted circles are taken from dictionary and white circles are taken from acoustic model HMM.

3) The Decoder

The job of the decoder is to take features from the Front End and search graph from the Linguist to produce result hypothesis.

IV. RESULTS AND DISCUSSION

We performed some experiments based upon different arrangements of datasets using 10 fold cross validation approach. The first two experiments are based upon the 100-words of dataset for comparison with the previous research work [2], while the other two experiments are based on the 250 words dataset. We have used individual speakers in training and testing as well as mixture of speakers in training and testing as explained later.

For 100-words dataset, the experiment comprises of 1000 samples (100 audio files for each speaker with a total of ten speakers). We adopt a 10 fold cross validation approach and calculate the mean and variance for accuracy and WER. The results are reported in Table II. Accuracy and Word Error Rate (WER) are calculated as:

$$WER = (S+D+I)/N \times 100, \qquad (1)$$

where S stands for substitution of a word, D stands for deletion of a word, I stands for insertion of a new word and N stands for the total number of words. Accuracy is the opposite of WER. 100% Accuracy shows 0% WER and 0% Accuracy shows 100% WER.

In second experiment, we have used 9 speakers in training and 1 speaker in testing using 10th fold cross validation and reported the results as shown in the Table III. The performance evaluation in this way ensures that the reported accuracy is for a speaker independent framework and any bias towards a particular speaker is catered for.

Table II. Results for 100-words dataset

Testing Data	**Accuracy (%)**	**WER (%)**
1st Fold	74.0	26.0
2nd Fold	76.0	24.0
3rd Fold	84.0	16.0
4th Fold	78.0	22.0
5th Fold	76.0	24.0
6th Fold	82.0	18.0
7th Fold	76.0	24.0
8th Fold	78.0	22.0
9th Fold	80.0	20.0
10th Fold	78.0	22.0
Mean	**78.2**	**21.8**
Variance	**3.05**	**3.05**

We have also performed the experiment for 250 words using a mixture of speakers in training and testing. This experiment comprises of total 2500 words using 10th fold cross validation approach. In each fold, 2250 words are used for training and the remaining 250 words are used for testing. We report the percentage accuracy for each fold in Table IV.

Table III. Results for 100-words dataset

Testing Data	**Accuracy (%)**	**WER (%)**
Speaker_1	88.0	12.0
Speaker_2	85.0	15.0
Speaker_3	89.0	11.0
Speaker_4	67.0	33.0
Speaker_5	61.0	39.0
Speaker_6	90.0	10.0
Speaker_7	67.0	33.0
Speaker_8	87.0	13.0
Speaker_9	81.0	19.0
Speaker_10	78.0	22.0

Mean	**79.3**	**20.7**
Variance	**10**	**10**

Table IV. Results for 250-words dataset

Testing Data	**Accuracy (%)**	**WER (%)**
1st Fold	78.0	22.0
2nd Fold	76.0	24.0
3rd Fold	76.0	24.0
4th Fold	80.0	20.0
5th Fold	77.6	22.4
6th Fold	76.8	23.2
7th Fold	79.6	20.4
8th Fold	74.8	25.2
9th Fold	80.0	20.0
10th Fold	78.0	22.0
Mean	**77.7**	**22.3**
Variance	**1.8**	**1.8**

Table V. Results for 250-words dataset

Testing Data	**Accuracy (%)**	**WER (%)**
Speaker_1	83.2	17.6
Speaker_2	80.0	20.0
Speaker_3	87.2	12.8
Speaker_4	63.6	36.4
Speaker_5	65.0	35.0
Speaker_6	86.4	13.6
Speaker_7	62.0	38.0
Speaker_8	75.2	24.8
Speaker_9	73.2	26.8
Speaker_10	70.8	29.2
Mean	**74.66**	**25.34**
Variance	**8.88**	**8.88**

Again for speaker independent setup, we have performed the experiment using 9 speakers (2250 samples) in training and 1 speaker (250 samples) in testing with 10th cross validation approach and reported the results as shown in Table V.

We also investigate the accuracy on a word-to-word basis for 250-words using 10 fold cross validation. We observe that only 2 words suffer 0 % accuracy. As shown in Figure 5, 40% words (100 words) are recognized with accuracy of 80% or above. Only 11% words (29 words) give accuracy below 50%. The detailed distribution of the number of words with respect to the percentage accuracy is shown in Figure 5.

We compare our results with previous results from Ali et al [2],[3]. The results reported in [2],[3] give recognition accuracy for the first ten words of the dataset. In Table VI, we observe that except for word no. 05, 09, 10, the accuracy achieved in this work outperforms the previous results. The framework reported in previous work had utilized only 100 words of the dataset while the results in this work are reported for all the 250 isolated words available in the dataset. Besides, we have reported results over 10-fold cross validation for fair calculation of accuracy. The 10-fold cross validation is missing in the previous work by [2],[3] on the same dataset. Similarly the overall result is higher than the previous one as can be seen from Table II and Table III. Our overall accuracy is 78.2% and the previously published work [3] accuracy is 70.69%.

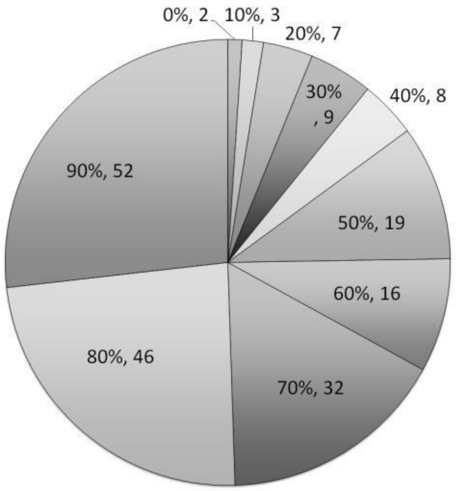

Figure 5: Distribution of number of words vs Percentage Accuracy. Note: The percentage values represent the accuracy and the integer values represent the number of words.

Table VI: Comparison with previous results using 100-words dataset

Word No.	**Percentage Accuracy (as in [2])**	**Percentage Accuracy (as in [3])**	**Percentage Accuracy (Our Approach)**
01	66.6	0	80
02	33.33	0	90
03	33.33	66.67	70
04	100	100	50
05	66.6	66.67	80
06	66.6	0	70
07	33.3	66.67	50
08	66.6	0	70

09	66.6	66.67	30
10	66.6	66.67	40

V. CONCLUSIONS

In this paper, we have developed an approach for automatic speech recognition of Urdu isolated words. This paper provided an extension to the previously published work on the Urdu dataset by using much larger number of words. It has also improved the results compared to those reported before. We also observed a better overall recognition accuracy values (78.2% for 100 words), compared to the accuracy reported previously (70.69% for 100 words). With the Urdu speech corpus freely available, we expect that this work will provide a good baseline for future research on Urdu automatic speech recognition. With the recent achievements in speech recognition using deep learning techniques, it becomes a very interesting task to explore the use of deep learning models for automatic speech recognition of Urdu.

REFERENCES

[1] H. Ali, N. Ahmad., K. M. Yahya, and O. Farooq, ‘A Medium Vocabulary Urdu Isolated Words Balanced Corpus for Automatic Speech Recognition’, Proceedings of 4th International Conference on Electronic Computer Technology (ICECT 2012), pp. 473-476.

[2] H. Ali, N. Ahmad, X. Zhou, “Automatic Speech Recognition of Urdu Words using Linear Discriminant Analysis", Journal of Intelligent and Fuzzy Systems, Accepted –vol. 28, no. 5, July 2015 pp.2369 - 2375

[3] H. Ali, N. Ahmad, X. Zhou, K. Iqbal, &S. Muhammad Ali, ‘DWT features performance analysis for automatic speech recognition of Urdu’, Springer Plus, 2014, Volume 3 (204).

[4] [Online] http://cmusphinx.sourceforge.net

[5] M. U. Akram, M. Arif, ‘Design of an Urdu Speech Recogniser based upon Acoustic Phonetic Modeling Approach’. Proc. 8th International Multitopic Conference (INMIC 2004), December, 2004, pp. 91 – 96

[6] S. M. Azam, Z. A. Mansoor, M. S. Mughal, S. Mohsin, ‘Urdu Spoken Digits Recognition Using Classified MFCC and Back propagation Neural Network’, Proc. Computer Graphics, Imaging and Visualization (CGIV 07), August 2007 pp. 414 – 418

[7] A. Ahad, A. Fayyaz, Mehmood T. Speech recognition using multilayer perceptron. In: Proceedings. IEEE Students Conference, ISCON ’02. Aug 2002, pp. 103 – 109.

[8] Hasnain S, Awan M. Recognizing spoken urdu numbers using fourier descriptor and neural networks with matlab. In: Second International Conference on Electrical Engineering, (ICEE 2008). March 2008, pp. 1–6

[9] Raza, S. Hussain, H. Sarfraz, I. Ullah, Z. Sarfraz, “Design and development of phonetically rich urdu speech corpus”, 2009 Oriental COCOSDA International Conference on Speech Database and Assessments.

[10] Ashraf J.; Iqbal, N.; Sarfraz Khattak, N.; Mohsin Zaidi, A., “Speaker Independent Urdu Speech Recognition using HMM”, Proc. The 7th International Conference on Informatics and Systems (INFOS), March 2010 pp. 1 – 5.

[11] Baker, J.K.: ‘The Dragon system - an overview’, in IEEE Transactions on Acoustic, Speech and Signal Processing,, 1975, Vol. 23, pp. no. 1 pp. 24 – 29.

[12] K. F. Lee, H.W.H., and R. Reddy: ‘An overview of the SPHINX speech recognition system’, IEEE Transactions on Acoustics, Speech and Signal Processing, 1990, Vol. 38, pp. no. 1, pp. 35-45

[13] X. Huang, F.A., H. W. Hon, M. Y. Hwang, and R. Rosenfeld: ‘The SPHINX-II speech recognition system: an overview’, Computer Speech and Language, 1993, Vol. 7, no. 2, pp. pp. 137-148

[14] P. Placeway, S.C., M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer: ‘The 1996 HUB-4 Sphinx-3 system’, in Editor (Ed.)^(Eds.): ‘Book The 1996 HUB-4 Sphinx-3 system’ (1997, edn.), pp.

[15] Rabiner, L.: “A tutorial on hidden markov models and selected applications in speech recognition”, Proceedings of the IEEE Vol:77 Issue:2, February 1989.

[16] Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., Wolf, P., Woelfel, J.: Sphinx4: a flexible open source framework for speech recognition. Technical report, Mountain View, CA, USA (2004)