Sphinx-4 Speech Recognition System Enhancements

Bhuvaneswari J, Krishnamurthy Karapattu, Munvar Ali Shaik, Udai Singh Shekhawat, Varun Nagpal

Larsen & Toubro Infotech Ltd,
# 25-31, EPIP Phase II, KIADB Industrial Area, Whitefield, Bangalore - 560066, India
Email: sphinx@lntinfotech.com
varun.nagpal@lntinfotech.com

Abstract– Sphinx-4 is a flexible, modular and pluggable framework based on Hidden Markov model (HMM) for speech recognition. Sphinx-4 is speaker independent framework which is freely available as open source. This paper targets the enhancements and additions that can be made to this open source framework to improve the performance of speech recognition. The performance of the speech recognition is measured on various parameters. The graphs of the performance results are displayed in the paper. Further enhancements are explained which can be added to the framework to overcome its constraint of vocabulary. Another concept of adaptive recognition for non native speakers with dual acoustic model is also explained to enhance the performance of speech recognition.

I.INTRODUCTION

The Sphinx-4 speech recognition system has been jointly developed by Carnegie Mellon University, Sun Microsystems Laboratories, and Mitsubishi Electric Research Laboratories (MERL). It has been built entirely in the Java programming language. The aim of this paper is to describe the modification, enhancement and additions that can be done to the framework to improve the performance and to measure and give the performance testing results done on speech recognition based on various parameters. There are three primary modules in the Sphinx-4 framework: the FrontEnd, the Decoder, and the Linguist. The FrontEnd takes one or more input signals and parameterizes them into a sequence of Features. The Linguist translates any type of standard language model, along with pronunciation information from the Dictionary and structural information from one or more sets of AcousticModels, into a SearchGraph. The SearchManager in the Decoder uses the Features from the FrontEnd and the SearchGraph from the Linguist to perform the actual decoding, generating Results. At any time prior to or during the recognition process, the application can issue Controls to each of the modules, effectively becoming a partner in the recognition process. For further details kindly refer section 'A' in the reference section i.e. Section: V.A.

The main blocks in Sphinx-4 Decoder Framework as shown in Figure.1 are the FrontEnd, the Decoder, and the Linguist. Supporting blocks include the Configuration Manager and the Tools block. Even the communication between the blocks, as well as communication with an application, is depicted in the above diagram.

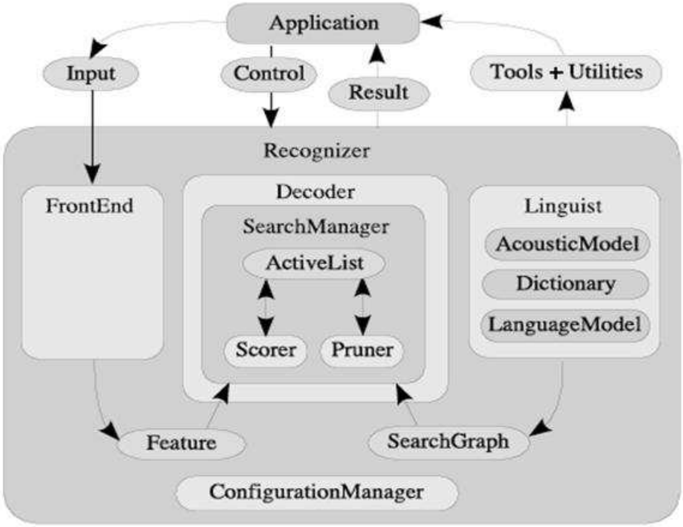

Figure 1: Sphinx-4 Decoder Framework.

II.MEASURE OF PERFORMANCE

The comprehensive performance testing of sphinx-4 was done based on the various parameters [explained in section 2.4]. Separate acoustic model, language model and dictionary were built for the testing purpose.

A. Acoustic Model

We used our own build acoustic model, which had 53 hours of training of 25 males and 25 female's voice. This acoustic model training was done with Indian accent English and with people from different age group varying from 20 years to 40 years and the training was done in a silent room with a high quality microphone. Sphinx Train tool was used to create the acoustic model from the training.

B. Dictionary

We built our own dictionary containing 900 words with 450 as the key words and 450 as the connecting conjunction words.

C. Language Model

Language model was built with same 900 words which were there in the dictionary.

D. Testing Parameters

The performance of the system was measure based on the following parameters.

1.Loudness/Amplitude of speech
2.Back ground noise
3.Different accents of speech
4.Duration of speech
5.Number of words in dictionary

E. Test results

The test results in the form of the following graphs were obtained by testing the sphinx-4 with our own created acoustic model, dictionary and language model. The testing was done in a silent room with high quality microphone. The original sphinx-4 framework was used without the speaker adaption module plugged into it. The accuracy of the performance was calculated based on the correct recognition of the keywords. In all the test cases the testing speech consisted of more than 80 percent of the keyword and 20 percent of connected words.

The charts shown can be referred for the various tests been taken on the tool. These are compared with the accuracy factor so that one can understand the various physical parametes which can effect the performance of the tool

Figure 2 shows the comparison between accuracy of Sphinx-4 system with respect to the pitch of input signal. The comparison is done for Low, Medium and High pitch signals.From the response it is clear that Accuracy of the system depends on the Amplitude of the input signal. The loudness of the input signal should be Medium else Accuracy will be decreased.

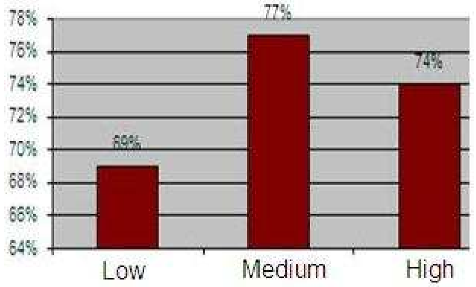

Figure 2: Amplitude (X-axis)v/s Accuracy(Y-axis).

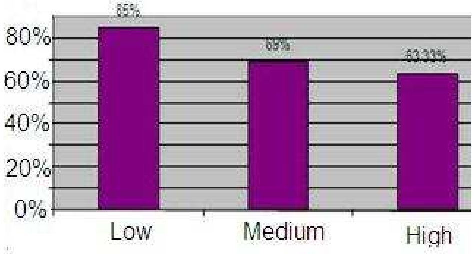

Figure 3: Disturbance(X-axis) v/s Keyword Accuracy(Y-axis).

Figure 3 shows the effect of Disturbance in the surrounding while recording the audio signal which is used as an input signal to the system. Accuracy of Sphinx-4 system decreases with the increase in presence of noise i.e Disturbance in the input signal.

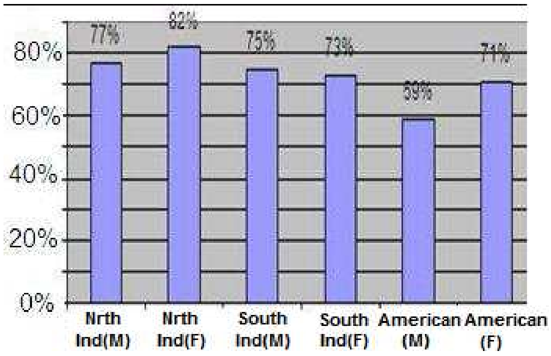

Figure 4: Accent(X-axis) v/s Keyword Accuracy (Y-axis).

Figure 4 shows the effect of Human Accent on accuracy of the system. Accuracy of the system differs for different Accents. We did testing on Accents like American, Indian for male and female respectively.

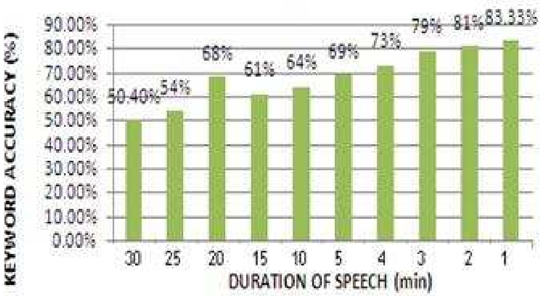

Figure 5: Duration of speech (X-axis) v/s Keyword Accuracy(Y-axis).

Figure 5 shows the effect of Duration of the Recorded input Signal on accuracy of the system. From this test it was found that Accuracy of the system decreases with the increase in the time for which the input signal is recorded. For example: Accuracy: 83.33 for 1 min of recorded signal while Accuracy: 50.40 for 30 min of recorded signal

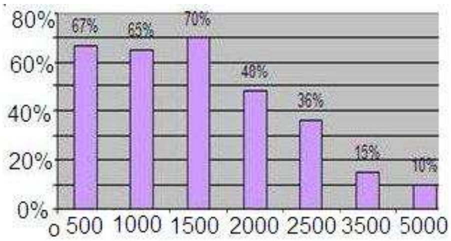

Figure 6: Number of new words in dictionary (X-axis) v/s Accuracy(Y-axis).

Figure 6 shows the effect of New words been added to the dictionary of the system on its accuracy. It was found that Accuracy of the system decreases with the increase in number of New words added in dictionary. For example: Accuracy: 70 percent for 1500 words while Accuracy: 10 percent for 5000 words.

III. ENHANCEMENTS AND ADDITION TO SPHINX-4 FRAMEWORK

A. Adaptive recognition for Non-Native Speakers

With extensive training and better acoustic modeling, current speaker-independent recognition systems have been successful in providing improved recognition for native speakers other than the one with which a system is trained, but recognition accuracy obtained using non-native speakers remains substantially worse. Some form of automatic speaker adaptation is required for truly speaker-independent recognition. By applying "machine learning techniques" self improving system can be made which can adapt and train itself automatically in real time while recognition with some input by a user.

1) Methods of Speaker Adaptation

- Selection and Mapping Techniques.

The mapping approach attempts to find a transformation between a set of reference parameters and the parameters obtained from a new speaker. The transformation may be in the form of a function which maps individual templates onto the reference, or by cluster selection where a set of templates is selected based on minimum distance criteria.

- Parameter Modification Techniques.

Parameter modification approach of speaker adaptation is done by updating the current system parameters based on the observations. Adaptation by parameter modification in HMM is same as applying Baum Welch Algorithm for a given observation set. Main challenge in this is after parameter re-estimation there may be degradation in a performance of system, because there are few training sets for re-estimation.

2) Technique we propose

We propose a solution which has slight resemblance to both of them. For adapting a speaker, we have made a separate trainer which generates separate models for each adapted words while recognition. Word are recognized simultaneously from reference [old model] as well as newly trained model [adapted model], scores of the recognized word are normalized by assigning their score to probabilistic score so that recognized word list can be compared and best one is selected. Thus without modifying reference model it generates result on basis of both the models, where parameters of newly adapted model can be modified according to speaker.

B. System Implementation

1) Training Module

Sphinx4 uses HMM models for recognition; it is not good to change the whole model on basis of few training data sets as it may adversely affect the performance of whole system. For this reason we have used Vector Quantization for training separate models for speaker adaptation. System trains itself while recognition if adaptation is enabled, codebook generated by training for the adapted word is latter used to generate result list which is scored appropriately with result list generated by Sphinx4 recognition engine for determining best result by Search Manager.

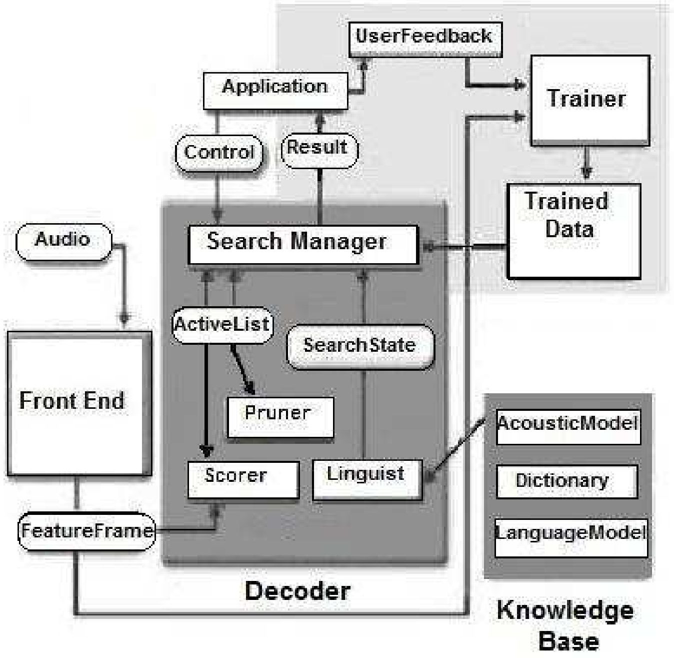

Figure 7: Sphinx-4 with Extension shaded in gray.

2) Search Manager

Results generated by both VQ and Sphinx4 are combined to get a common result list. VQ selects a word which is nearest to code vector from the trained codebooks, thus according to VQ word with minimum distance is best, whereas in Sphinx4 word with highest score along the HMM state path is best. We combine both result list by re-assigning there score to probabilistic score so that it can be compared.

3) Mathematical Explanations

Let the word score generated by VQ for words

$$w1_v, w2_v.......wn_v is wS1_v, wS2_v.......wSm_v$$

Since word with minimum score is best we modify the score such that word with maximum score is best in following way:

$$wSi^{'}_V = 1 - \frac{wSj_V}{\sum_{k=1}^{n} wSk_V}$$

Let the word score generated by Sphinx for words

$$w1_s, w2_s.......wm_s is wS1_s, wS2_s.......wSm_s$$

Thus new scores for these words will be:

$$wSj^{'}_s = \frac{wSj_s}{\sum_{k=1}^{n} wSk_s}$$

These two results are combined by giving appropriate weight age to both the scores and latter on word with highest score is selected as recognized word.

IV. FUTURE WORKS

Despite its importance, robust speech recognition has become a vital area of research. The desire for a speaker-adaptive, continuous-speech system places two important restrictions on the adaptation process. One is the necessity for a quick adaptation procedure. That means the algorithms must be able to adapt on a relatively small amount of speaker-specific training data, and that to with less computational load. Given this limited set of observations, the adaptation algorithm should be able to exploit any information available in the data quickly, and without sacrificing accuracy for speed. The second restriction is imposed by the continuous speaking style. With continuous speech, user feedback of phonetic or other sub word recognition unit labels is unrealistic. The adaptation algorithm must therefore operate in an unsupervised mode

V. REFERENCES

[1] http://cmusphinx.sourceforge.net/sphinx4

[2] http://www.speech.cs.cmu.edu/sphinx/twiki/bin/view/Sphinx4/WebHome

[3] http://www.aisee.com/apps/sphinx4.htm

[4] http://www.merl.com/projects/sphinx4

[5]L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of IEEE, pages 257-286, Feb 1989

[6]W. Walker P. Lamere P. Kwok B. Raj R. Singh E. Gouvea P. Wolf and J. Woelfel. Sphinx-4: A flexible open source framework for speech recognition. Sun Microsystems, Tech. Rep, Nov 2004