

Hindi ASR for Travel Domain

Sunita Arora, Babita Saxena, Karunesh Arora, S S Agarwal
Centre for Development of Advanced computing, Noida, India
{sunitaarora, babita, karunesharora, ssagarwal }@cdacnoida.in

Abstract

This paper presents our experiments for a baseline speech recognizer for Hindi language. The recognizer is developed using Julius Speech recognition engine. Julius is a high performance; two pass large vocabulary continuous speech recognizer (LVCSR) which performs recognition taking an acoustic model and a language model as input. HTK is used for building acoustic model and SRILM toolkit is used for building language model. This system recognizes spoken sentences in the travel domain. The acoustic model is trained on 26 hours of audio data of 30 speakers. The Language Model is trained with 167057 words. The vocabulary size of the recognizer is 9103 words. The system is tested on 20 speakers and the performance of the system is reported.

1. Introduction

In order to build a LVCSR system, perfect integration of a highly accurate acoustic model, a large scale language model and an efficient decoder is required.

Julius is an open source-two pass large vocabulary continuous speech recognizer (LVCSR). It can perform almost real time, hi- speed decoding based on a N-gram language model and tri-phone context dependent acoustic model as an input utilizing a small amount of memory. It also supports a variety of features such as tree-based lexicon, cross word context handling, N-gram factoring, enveloped beam search, Gaussian selection and Gaussian Pruning.[1]

The system is being developed for the Hindi language. In this paper we present our work on building acoustic and language models for Hindi language. Hindi belongs to the Indo Aryan family of languages and is written in the Devanagari script. There are 11 vowels and 35 consonants in standard Hindi. In addition, 5 Nukta consonants are also adopted from Farsi/Arabic sounds.

Hindi is mostly phonetic in nature and therefore usually has a one to one mapping between the orthography and their pronunciation. They possess a large no of phonemes like retroflex and aspirated stops which are absent in English and other European Languages.

This paper is organized as follows. Section-2 describes the Acoustic model. Section-3 describes language model. Section-4 describes the decoding process. Section-5& Section-6 focuses on the observation, results and conclusion of developed Hindi ASR system. Future work is stated in Section-7.

2. Acoustic model

The acoustic model is built using HTK 3.4 toolkit [6]. In a statistical framework for speech recognition, the problem is to find the most likely word sequence

$$\hat{W} = \arg \max_w p(W / A) \quad (1)$$

With a Bayesian approach to solving the above problem, we can write

$$\hat{W} = \arg \max_w p(A / W) p(W) \quad (2)$$

Equation 2 gives two main components of a speech recognition system, the acoustic model and the language model. The first term of equation-2 is computed by acoustic model and the second term is computed by language model. The Acoustic Model training procedure is shown in Fig-1. The acoustic model is created by training the recorded audio data out of which MFCC feature vectors are extracted and their delta and delta-delta features are considered. The HMMs are trained over 61 context independent phonemes. Details of phoneme frequency in the training corpus is shown in table-2 in appendix-A. A phoneme 'sil' is used for silence at beginning and end of a sentence. The training data is collected from 30 female speakers in a clean noise free environment, which consisted of approximately 26 hours of speech

recording. In all 8567 sentences containing 74807 words are recorded by the speakers uniformly distributed over all age groups from 17 to 60 yrs. The utterances are recorded in 48 KHz, stereo 16 bit format. Two mics were used one for recording left channel and the other for right channel. The recording is done using M-Audio Fast Track Pro USB Audio Interface. It is then channel separated and down sampled to 16 KHz and then single channel is used to train acoustic models. Each HMM definition file represents a single stream single-mixture diagonal covariance left-right HMM with five states. The primary feature vector size of 13 is considered along with their delta and delta-delta features.

Prototype models for 61 phonemes are built using flat start approach. These models were further refined by applying nine iterations of the standard Baum-Welch embedded training procedure. These models are then converted to triphone models and two iterations of Baum-Welch training procedure are applied, then these states are tied using decision tree based approach and two iterations of Baum-Welch training procedure are applied. Now the number of mixtures is incremented to 14 and seventeen iterations of the standard Baum-Welch training procedure were applied.

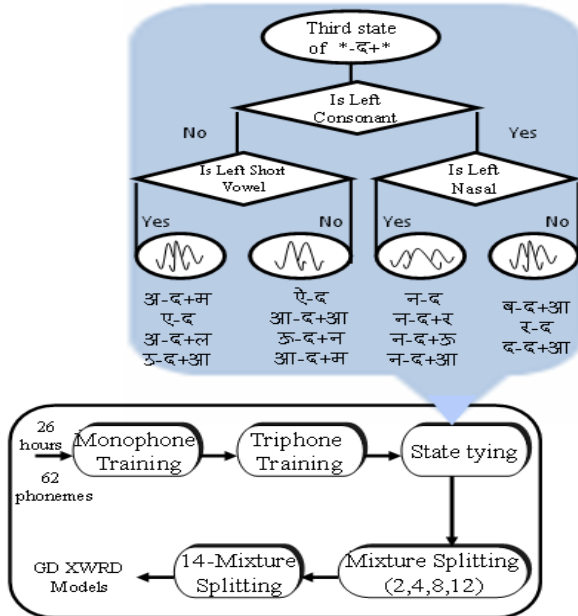


Figure 1: Acoustic model training procedure

Despite the speech data is recorded in noise free studio environment and pronunciation mistakes were taken care of, a few errors were found after recording. These errors were compensated by changing the transcription accordingly as per the pronunciation, by manually listening to the speech.

3. Language model

In speech recognition, language model tries to capture the properties of a language, and to predict the next word in a speech sequence i.e. the model that computes the probability of a word given some previous words. The second components of equation 2, $p(W)$ is the prior probability computed by language model. This is probability of a sequence, as shown below.

$$p(w_1, w_2, w_3, w_4, w_5 \dots w_n) = p(W)$$

By Chain rule the probability of n^{th} word is:

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2) \dots P(w_n|w_1^{n-1})$$

$$= \prod_{k=1}^n P(w_k|w_1^{k-1})$$

SRILM language modeling toolkit [2] is used to train both bi-gram and reverse tri-gram language models over 167057 words. Both the models were generated in the standard ARPA format. Word bi-gram model is used in the first pass while reverse word tri-gram was used in the second pass. To speed up the recognition process both the models were converted into a single binary file.

The lexicon used is in the HTK dictionary format consists of 9103 number of distinct words although the default maximum vocabulary size is 65535 words. Each word in the lexicon is described as a combination of individual sub-word units as per the acoustic model. Multiple pronunciations of a word were written as separate words.

4. Decoding

Julius recognition engine is used for decoding the utterances. Julius works in two passes. The first pass is a high-speed approximate search, which uses bi-gram frame synchronous beam searching algorithm. In the first pass, a tree-structured lexicon assigned with the language model probabilities is applied. Pre-computed uni-gram factoring values are assigned to the intermediate nodes and bi-gram probabilities on the word-end nodes. The second pass is a high precision tri-gram N-best stack decoding. The tree trellis search in the second pass recovers the degradation caused by the rough approximation in the first pass.

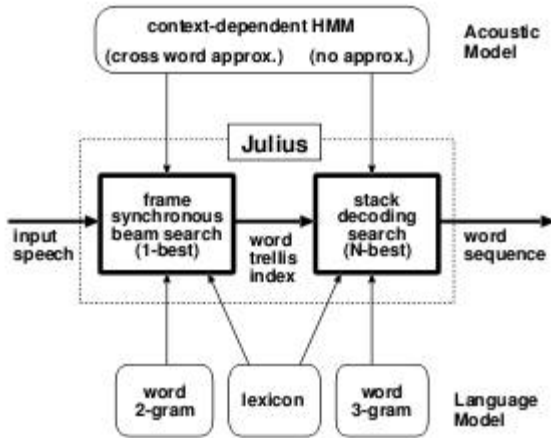


Figure 2: Julius decoding process

5. Observation and result

The performance of Hindi ASR is measured in terms of word recognition rate. The system is tested on 10 seen speakers and 10 unseen speakers where seen speakers stands for speakers which are from the training corpus itself and unseen speakers stands for the speakers which are not included in the training corpus. Testing is done on total number of 6256 sentences of travel domain. The same acoustic model, lexicon and language model is used for both seen and unseen data. The output files in the Julius format are converted to standard MLF format and the performance is analyzed using HTK toolkit's HResult tool. The percentage number of correct labels recognized is given by:

$$\%Correct = \frac{H}{N} \times 100\%$$

Where, H and N are the number of correct labels and total labels respectively.

Speaker	Sentences	Recognition rate
Training	3128	70.73%
Test	3128	60.66%

Table 1: Word recognition rate for 10 speakers in training and test sets

The word level recognition rate of 70.73% and 60.66% is observed for seen and unseen speakers respectively.

6. Conclusion

In this paper, we described our initial experiments with the domain specific large-scale Hindi speech recognition using Julius. A baseline speech recognizer was developed and the results found are quite encouraging.

7. Future work and scope

Our future work will be to further refining the word accuracy and supporting. A number of further experiments may be carried out to achieve accuracy improvement. Some of them are described below.

- Training corpus may be increased and phonetically balanced data may be used.
- Question set may be refined for decision tree based clustering.
- We have so far used single data stream HMM definitions, varying the number of streams to get an optimum value may be beneficial.
- Feature sets other than MFCCs may be tried such as Linear Predictive Coefficients (LPCs), Linear Predictive Cepstra (LPCepstra), log-scaled Filterbank energies (FBANK) and Perceptual Linear-predictive coefficients (PLPs)
- Size of Language model may be further increased and smoothing may be applied.

8. References

- [1] Akinobu Lee, Tatsuya Kawahara, Kiyohiro Shikano, *Julius — an Open Source Real-Time Large Vocabulary Recognition Engine*
- [2] Andreas Stolcke, "SRILM an extensible language modeling toolkit" STAR Laboratory, SRI International, Menlo Park, CA, U.S.A., <http://www.speech.sri.com/>
- [3] M Kumar, et al "A large-vocabulary continuous speech recognition system for Hindi", IBM Research and Development Journal, September 2004.
- [4] Rajat Mathur, Babita, Abhishek Kansal, "Domain specific speaker independent continuous speech recognition using Julius", ASCNT 2010.
- [5] Steve Young, Gunnar Ever, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Vaitchev, Phil Woodland, "The HTK Book", copyright 2001-2002 Cambridge University Engineering Department.

Appendix A:

Hindi Phone in WX notation	Hindi Alphabet in Devanagari Script	Frequency in Training corpus
a	अ	24610
A	आ	27333
AV	औ	644
az	अं	575
Az	आँ	1322
b	ब	4338
B	भ	1130
c	च	3322
C	छ	1060
d	ड	1311
D	ढ	79
dZ	ड़	744
DZ	ढ़	124
e	ए	17160
E	ऐ	5516
ez	एं	3312
Ez	ऐं	4025
f	फ	250
F	ज	32
g	ग	4122
G	घ	231
gZ	ग़	27
h	ह	14978
i	इ	10295
I	ई	9503
iz	इं	85
Iz	ईं	777
j	ज	3282
J	झ	1239

jZ	ज़	658
k	क	20682
K	ख	1460
kZ	क़	38
KZ	ख़	87
l	ल	7203
m	म	10901
n	न	9560
N	ण	451
o	ओ	5119
O	औ	1132
oz	ओं	580
Oz	औं	15
p	प	8665
P	फ	186
PZ	फ़	866
r	र	15287
R	ष	700
s	स	11231
S	श	1618
t	ट	3670
T	ठ	560
u	उ	4351
U	ऊ	1438
uz	ऊं	170
Uz	ऊँ	2390
v	व	2908
w	त	7860
W	थ	979
x	द	4556
X	ध	679
y	य	8112

Table 2: List of Phonemes and their frequencies in training corpus