Lexical Modeling for Non Native Speech Recognition using Neural Networks

Rahul Chitturi, Venkatesh Keri, Gopalakrishna Anumanchipalli, Sachin Joshi
Speech Lab, LTRC,
IIIT-Hyderabad, INDIA
{rahul_ch,venkateshk,gopalakrishna,sachin_sj}@students.iiit.ac.in

Abstract

Building an Automatic Speech Recognition System for a language requires a good amount of clean speech data and significant effort in training it. More so for foreign language speech recognition, where the native accent of the speaker brings about, a considerable difference in the pronunciation. In this paper, we propose a neural network based approach to model the lexicon of the foreign language with a limited amount training data. The training data, in this case, is necessarily a hand-crafted dictionary of the foreign language with the phone set of the native language. The neural network learns how the phones of the foreign language vary with different instances of context. The trained network is capable of deciphering the pronunciation of a foreign word given its native phonetic composition. The performance of the technique has been proved quite promising. As a test bed to prove our arguments, we chose the recognition of Indian accented English, using three Indian language acoustic models.

1 Credits

Thanks to our guide Mr. S.P.Kishore, LTI, CMU. We would like to acknowledge Mr. Satinder and Dr. Sitaram from HP Labs who gave the initiative for this project. This work is sponsored by Hewlett Packard Labs, Bangalore, India with the collaboration of Carnegie Melon University, USA and IIIT-Hyderabad, India. We are very thankful to all the members of our team (J.Natraj, S.V.Pavan Kumar, P.Dilip, Jithendra, and Gaurav Somani) who have worked on this.

2 Introduction

In large countries like India with large migrant populations, there is a wide range of strong accents among people whose first language is a foreign language. The speaker variations due to foreign accents complicate the task of automatic speech recognition. One reason is because the non-native speakers' pronunciation differs from that of the native speakers who are modeled during system training. Co-articulation is a chief problem in speech recognition. It is problematic inside of words, where phonemes are dropped, changed, or categorically replaced, but it is worse between words, often obscuring the boundaries. Possible solutions for speaker and accent independent speech recognition could be adaptation to the accented speech, processing through accent dependent recognition channels and the utilization of accent specific phonetic and phonological knowledge. This work is an attempt using the baseline large vocabulary continuous speech recognition systems (LVCSR) for the three Indian Languages Telugu, Tamil and Marathi [13] which are very recently developed.

The human articulatory system will be aligned in accordance with the features of their mother tongue. So, the speakers with foreign accents are expected to import some of the acoustic and phonological features from their naive languages into the speech production process of a new language. This requires, for system building, a lot of foreign accented speech data to be collected which is particularly tedious.

Considering these problems, we have developed a scheme for Indian English speech recognition systems with the three databases available in Telugu, Tamil and Marathi. In short, the phones of the US English Phones have to be mapped to those of the

language (Telugu, Tamil, and Marathi) that is being used for recognition. For automating the process, the Artificial Neural Network (ANN) paradigm is employed to get the corresponding Indian phone mapping for a US phone sequence. Consequently, the transcription of the US English that has to be recognized is converted automatically into phone sequences of respective Indian Languages.

3 Related Work

IBM India Research Lab (IRL) has developed first concatenative Indian English text-to-speech synthesis system which is capable of synthesizing speech in Indian accent and can pronounce Indian names with very high accuracy [6]. In the Indian context, LVCSRs were developed for Hindi by IBM [7] and Hewlett Packard Labs [8]. More recently, LVCSRs were also developed in Marathi, Tamil and Telugu [13]. Not much work has gone into building systems for Indian English. Typically, for mapping the English phone set to Indian phones, letter to sound rules [10] are used. But in the current work, we present a novel approach of employing Artificial Neutral Network Models for automating this mapping.

4 Database Description

The databases used for training the native language recognition systems had speakers from various parts of the respective states (regions) in order to cover all possible dialectic variations of the language. 52 sentences of the optimal text [14] were recorded by each speaker. Table 1 gives the number of speakers recorded in each language and in each of the recording modes – landline and cell phones. To capture different microphone variations, four different recording media were used while recording the speakers.

Table 1: Number of speakers in the three languages

Language	Landline	Cellphone	Total
Marathi	92	84	176
Tamil	86	114	200
Telugu	108	75	183

Language	Male	Female
Marathi	91	85
Tamil	118	82
Telugu	93	90

Table 2: Gender distribution among speakers

5 Baseline systems

All the experiments dealt in this paper use the sphinx II framework developed at CMU. Training is done using the sphinx-3 trainer. 5 state semi continuous HMMs (SCHMM) allowing skipstate were used as the acoustic models for each phone. Testing is done using the sphinx-2 decoder. Language Model (LM) is generated by the CMU Statistical Language Model (SLM) Toolkit [2]. Performances were reported in terms of the word error rate metric using the nist scorer.

The basic parts in speech recognition are the Acoustic model, the Language model and the lexicon. Generally, as the non-native speaker speaks a foreign language with an accent influenced by his mother tongue, acoustic models derived from the native language are used as those for the target language. The figure 1 shows the schematic diagram of the scheme employed for developing the recognition system.

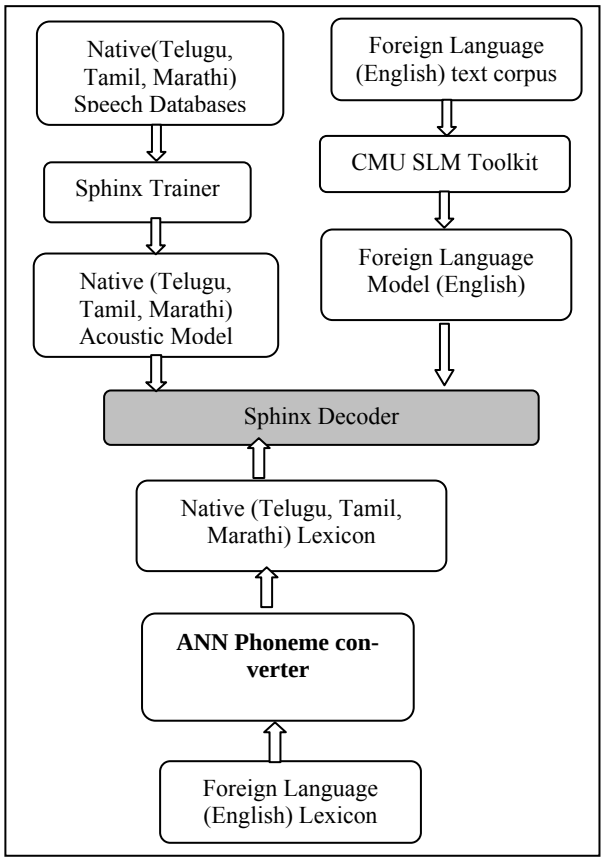


Figure 1: Schematic Diagram of the Indian English Speech Recognition system

5.1 Acoustic model (AM)

The first key constituent of the ASR systems is the acoustic model. Acoustic models capture the characteristics of the basic recognition units [3]. The recognition units can be at the word level, syllable level and or at the phoneme level. Many constraints and inadequacies come into picture with the selection of each of these units. For large vocabulary continuous speech recognition systems (LVCSR), phoneme is the preferred unit. Neural networks (NN) and Hidden Markov models are the most commonly used for acoustic modeling of ASR systems. We have chosen the Semi-Continuous Hidden Markov models (SCHMMs) [1] to represent context-dependent phones (triphones). These Acoustic Models were developed for the three Indian Languages Telugu, Tamil and Marathi. As mentioned earlier, these models were used as the seed models for the foreign language.

5.2 Language Model (LM)

The language model attempts to convey the behavior of the language. It aims to predict the occurrence of specific word sequences possible in the language. From the perspective of the recognition engine, the language model helps narrow down the search space for a valid combination of words [3]. Most ASR systems use the stochastic language models (SLM). These probabilities can be trained from a corpus. SLMs use the N-gram LM where it is assumed that the probability of occurrence of a word is dependent only on the past N-1 words. The LM was created using the CMU statistical LM toolkit [2]. The target language text corpus is used to train the LM in the current case.

5.3 Lexicon

Lexicon gives the pronunciation of the words in that language. For the present system, as the Acoustic Models are in native languages the lexicon has to be generated in the native languages. To build the native Lexicon from the foreign Lexicon, the Artificial Neutral Network Phoneme Converter is used, which maps the phones of English to the phones of Indian Languages depending on the contextual information. This is explained in the section 6.

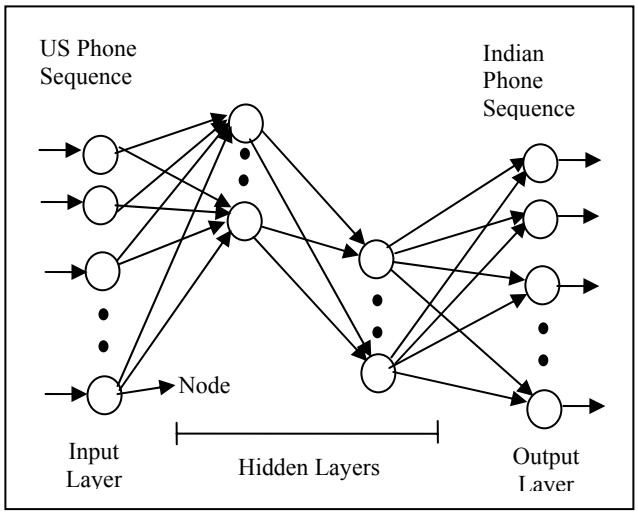
6 Phone mapping using ANN

The English phoneme set that is referred in our experiments is the phoneme set of CMU Pronouncing Dictionary version 0.6 [9]. For the Indian phonemes the ITrans3 format is used. There are 49 phones in Telugu, 35 in Tamil, 48 in Marathi and 40 in US English.

6.1 Basic Description of the ANN mapper

The algorithm which was used for ANN is Back Propagation [11]. Basic building blocks are layers, nodes. The architecture of ANN is shown in the figure 2.

Figure 2: Architecture of ANN for Lexical Modeling



6.1.1 Layers

The 3 different types of layers are Input Layer, Hidden layers & Output Layers .All these layers will be having different configurations depending on the need. All the total calculations are done in the hidden layers. These layers are interconnected by connections which may be one to one or one to many or many to many.

6.1.2 Nodes

Each layer contains number of nodes depending on the layer in which they are present. Number of nodes in the input layer depends on the Input Vector and number of nodes in the output layer depends on the Output Vector but the number of nodes in the hidden layers is not fixed and the performance very much depends on these nodes.

6.2 Building the modules for each US phone

This module sets all configuration files for each of the US phones. It conducts four experiments for all of the phones. The optimal model is finally used for recognition. The configuration file contains the number of hidden layers, number of nodes in each of the input hidden and output layers. It also contains the learning rate and the type of activations used in each of the input, hidden and output layers.

The different activations are Tangent (Hyperbolic) denoted by N, Sigmoid denoted by S and Linear denoted by L. they vary in their limits which are shown in table 3. These have to be configured depending on the necessity of the Training. For example the configurations which are used for the different experiments for each phone is shown in table 4 (5 L 15 N 47 N implies 5 nodes in Input layer, 15 nodes in Hidden layer and 47 nodes in Output layer).

Activation Type	Lower Limit	Upper Limit
Tangent (N)	-1	1
Sigmoid (S)	0	1
Linear (L)	-∞ (-inf)	∞ (inf)

Table 3: Types of Activations

Experiments	ANN Models	Learning Rate
Experiment 1	5L 15 N 47 N	0.001
Experiment 2	5L 30 N 47 N	0.001
Experiment 3	5L 45 N 47 N	0.0011
Experiment 4	5L 60 N 47 N	0.0015

Table 4: Different Configurations Used

6.3 Training the ANN

The format that's used for training the ANN system is explained in section 5.3.1. Training was done on 500 such Training Vectors for each model (i.e. English -Telugu, English - Tamil, English - Marathi).

6.3.1 Training Vector

(Normal Line)<s>(arctic_a0001 "Author of the danger trail, Philip Steels")<s>

(US Phone Sequence)<s>pau - pau - ao th er NUL - ah v - dh ax - d ey n jh er NUL - t r ey l - pau - f ih l ax p - s t iy l z - pau <s>

(Telugu Phone Sequence)<s>pau - pau - aa t a r - aa ph - d a - dz e n j a r - tz r ai l - pau - ph i l i p - s tz ii l s – pau <s>

(Tamil Phone Sequence)<s>pau - pau - aa t a r - aa p - t a – tz e n j a r - t' r ai l - pau - p i l i p - ch t' ii l ch - pau <s>

(Marathi Phone Sequence)<s>pau - pau – aa* t a r – aa* ph - dh a - d' ei n j a r - t' r ai l - pau - ph i l i p - s t' ii l s - pau <s>

6.3.2 Input Vector

The input given to ANN should follow some specifications of the Input Vector which is given below. This Input Vector can contain any number of states depending on the complexity of the training data and we are using five states. These states should be in the form of decimal values which are between 0 and 1. The values for each US phone should be set such that all the phones are evenly distributed over the interval 0 to 1. For example, a phone having the index value 23 will have the decimal value 23/45

Each of these Input Vectors is nothing but the Nodes of Input layer which is specified in the ANN architecture.

<PPPH> <PPH> <PH> <NPH> <NNPH>

Where <PH> is Present Phone, <PPPH> and <PPH> are the two previous phones, <NPH> and <NNPH> are the subsequent phones in the current phone's context. For example

0.6889 0.0222 0.822 0.0444 is same as <ey><l><pau ><f> <ih>

6.3.3 Output Vector

The output layer contains 'n' number of states which is equal to the number of Indian language phones (i.e. Telugu, Tamil and Marathi) respectively. In the output vector the value of that state is 1 for which the mapping from US phone to Indian Phone is correct and rest of the states are -1's. For each input Vector there should be a corresponding output vector. For 'n' number of Indian phones the output Vector is as below

<-1> <-1> <-1> …. <1> ….. <-1> <-1>
1^{st} 2^{nd} 3^{rd} k^{th} $(n-1)^{th}$ n^{th}

6.3.4 Training Method

The input and output vectors are given as inputs to the Training along with the configuration files. Training is an iterative process of learning. During training the error in the first iteration is calculated depending on the input and output and then before the next iteration the error is reduced depending on the learning rate and the activation. This is given to the second iteration and so on. This continues till the error in the present iteration is less than the previous iteration else it will stop and build the values for the trained module.

As we have 'n' number of states in output vector (i.e. 49 in Telugu, 35 in Tamil and 48 in Marathi) and in each output vector we have (n-1) wrong states (i.e. -1's) and 1 correct state (i.e. 1) so the neural network will be trained more for -1's rather than for 1's. So for training all the US phones at a time then there will be high ambiguity factor in selecting the correct Indian phone after training. In order to reduce this ambiguity factor multiple trainings are done separately for each of the US phones.

7 Results

The tests on each system were done using the speech data of the untrained 30% of the speakers' data. The utterances were decoded using the Sphinx II decoder. Appropriate tuning was done on the decoder to get the best performance. The evaluation of the experiment was made according to the recognition accuracy and computed using the word error rate [WER] metric which align a recognized word string against the correct word string and compute the number of substitutions (S), deletions (D) and Insertions (I) and the number of words in the correct sentence (N).

W.E.R = 100 * (S+D+I) / N

Tables 5-7 show the performance of each ASR in terms of their WER. It shows the comparison with the actual Indian Language ASR's and also the size of the vocabulary of the system is shown.

Case	WER%	vocabulary
Telugu recognition	15.1	25626
English recognition under Telugu Acoustic models	26.3	985

Table 5: Accuracy of the Telugu ASR.

Case	WER%	Vocabulary
Tamil recognition	17.6	16187
English recognition under Tamil Acoustic models	24.7	985

Table 6: Accuracy of the Tamil ASR.

Case	WER%	Vocabulary
Marathi recognition	20.7	21640
English recognition under Marathi Acoustic models	28.9	985

Table 7: Accuracy of the Marathi ASR.

The Performance of the ANN Phoneme mapper is tested on 100 testing vectors (which are of the same format as training vectors). Depending on the performance in each experiment, the best trained modules (phones) are used for speech recognizing. The overall performance of successful mapping for all the phones in Telugu, Tamil and Marathi are shown in the *Table 8.*

English	Telugu Phones	Performance % on Telugu Phones	Tamil Phones	Performance % on Tamil Phones	Marathi Phones	Performance % on Marathi Phones
aa	a / aa	83.33	a / aa	71.54	a / aa / aan:	61.23
ae	e / ai / aa	50	e / ai / aa	73.34	e / ai / aa/ aan:	69.45
ah	a	40%	a	58.76	a	82.16
ao	o / oo	50%	o / oo	72.12	o	79.33
aw	au	81.33	au	84.78	au	86.38
ax	a	91.87	a	75.76	a	82.25
ay	ai	63.99	ai	93.54	ai	96.67
b	b / bh	85	p	92.43	b / bh	95.83
ch	ch / chh	89	ch	95.63	ch / chh	83.33
d	dz / d	41.66	t	96.8	d' / d	98.77
dh	dhz / dh	65.35	t	96.37	dh' / dh	84.05
eh	e	89.05	e	93.78	e	87.36
er	a / aa / (a + r)	68.72	a / aa / (a + r)	62.46	a / aa /(a+r)	74.53
ey	e	50.00	e	91.23	e	91.52
f	ph	80.80	p	91.89	ph / phu	84.87
g	g / gh	75.00	k	87.67	g / gh	86
hh	h	88.88	h	92.56	h	94.47
ih	i	81.81	i	85.12	i	87.87
iy	ii	81.81	ii	86.34	ii	85.15
jh	j / jh	80.75	j	84.56	j / jh	61.54
k	k / kh / qs_	85.78	k	95.67	k / kh	97.3
l	l / lz	91.66	l / lz	85.34	l / l'	89.89
m	m	91.66	m	82.78	m	97.92
n	n	95.67	n	99.00	n	91.36

ng	nd_/ nj_	90.48	nd_/ nj_	89.38	nd_/ nj_ / dny / ng_	90.48
ow	o	88.89	o	92.79	o	78.87
oy	(o + ai)	50.12	(o + ai)	65.76	(o+ai)	50
p	p / ph	69.43	p	97.56	p	96.43
r	r / r-	94.77	r / r- / rh-	62.22	r / r-	97.22
s	s	90.47	ch	91.01	s	97.01
sh	sh / shh / qs_	89.04	sh	85.95	sh / shh	56.97
t	tz / t	75	tz	75.29	t / tz	83.71
th	thz / th	72.67	t	91.67	th / thz	81.67
uh	u	87.71	u	82.12	u	73.62
uw	uu	78.34	uu	50	uu	89.45
v	v	60	v	81.9	v	91.9
w	v	86.776	v	86.78	v	87.19
y	y	89.91	y	91.01	y	79.96
z	s / j	90.47	ch / j	58.14	s / j	67.14
zh	sh / j	50	sh / j	50	sh / j	73.45

8 Conclusions and Future Work

From our experiments, it has been proved that ANN is a promising technique for automating lexical modeling accented or non-native speech recognition. Our future work will focus on laying a generic framework for accented speech recognition and studying the limitations of the applicability of the technique. Work is also on towards extending the concepts of neural networks for cross-lingual and cross-dialectic speech recognition.

9 References

[1] L.Rabiner., *"A Tutorial on Hidden Markov models and Selected Applications in Speech Recognition"*, Proc. Of IEEE, Vol. 77 No. 2, 1989.

[2] Rosenfeld Roni, *"CMU statistical Language Modelling (SLM) Toolkit (Version 2)"*.

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

[4] T.Cormen, C. Leiserson and R. Rivest,. *"Introduction to Algorithms"*, The MIT Press, Cambridge, Massachusetts, 1990.

[5] Kalika Bali, Partha Pratim Talukdar, *"Tools for the development of a Hindi Speech Synthesis System"*, 5th ISCA Speech Synthesis Workshop, Pittsburgh, pp.109-114,2004

[6] www.research.ibm.com/irl/projects/speech.shtml

[7] Mohit Kumar, Nitendra Rajput, Ashish Verma, *"A large vocabulary continuous speech recognition system for Hindi," IBM Journal of Research and Development (Special Issue on IBM Research in Asia)*

[8] Singh, S. P., et al *"Building Large Vocabulary Speech Recognition Systems for Indian Languages"*, International Conference on Natural Language Processing, 1:245-254, 2004.

[9] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[10] Vincent Pagel, Kevin Lenzo, and Alan W Black. *Letter to sound rules for accented lexicon compression*. In ICSLP98, volume 5, pages 2015-2020, 1998.

[11] B. Yegnanarayana and S.P. Kishore, AANN - *An Alternative to GMM for Pattern Recognition*, Neural Networks, vol.15, no.3, pp. 459-469, April 2002.

[12] B. Yegnanarayana, S.P. Kishore, and A.V.N.S. Anjani, *Neural network models for capturing probability distribution of training data*, in Int. Conference on Cognitive and Neural Systems, (Boston), p. 6(A), 2000.

[13] GopalaKrishna et al. – *Development of Indian Language Speech Databases for Large Vocabulary Recognition Systems*, to appear in proceedings of SPECOM 2005, Greece

[14] Rahul et al. – "*Rapid Methods for Optimal Text Selection"*, proceedings of RANLP 2005, Bulgaria