Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 6, June 2014, pg.116 – 120

RESEARCH ARTICLE

RESEARCH ON PHONEME SEQUENCES FOR LANGUAGE IDENTIFICATION AND CONCURRENT VOICE TRANSMISSION

Ms. Sunder[1], Ms. Pratima Sharma[2]
[1]Asst. Prof, Department of Computer Science Engineering, AITM, PALWAL, INDIA
[2]Student, AITM, PALWAL, INDIA
[1] simmibhamla@gmail.com, [2] pratima_7521@yahoo.in

Abstract: Language Identification is process of identifying the language being spoken from a sample of speech by an unknown speaker. Most of the previous work in this field is based on the fact that phoneme sequences have different occurrence probabilities in different languages, and all the systems designed till now have tried to exploit this fact. Language identification process in turn consists of two sub-systems. First system converts speech into some intermediate form called as phoneme sequences, which are used to model the language by doing their probabilistic analysis in the second sub-system. In this project both of the sub-systems are targeted. First some algorithms are discussed for designing language models. Then an attempt is made to design an algorithm for extracting phoneme sequences in form of more abstract classes derived by statistical tools like Gaussian Mixture Models (GMM) and Hidden Markov Model (HMM).

Keywords: Language Identification, Phoneme, Sequences, Tuning, Recognizer

1. INTRODUCTION

The problem of Language Identification (language ID) is defined as recognizing the language being spoken from a sample of speech by an unknown speaker[1]. The human is by far the best language ID system in operation today, with accuracy as high as hundred percent in case if they know the language and can make a pretty reasonable guess about them in case if they don't. This project has tried to develop this ability in machines.

Several important applications already exist for language ID. A language ID system could be used as a 'front-end' system to a telephone-based company, routing the caller to an appropriate operator fluent in the caller's language. Currently either a manual system or IVRS based system exists. But, both of them suffer from two main problems, however: speed and expense. It is highly expensive to employ the call routers for AT&T who, between them, must be able to correctly route 140 languages or Infocomm for more than 10 languages. For emergency services, this could be a fatal delay. Other application includes usage of such systems in war times when soldiers are doing rescue operations in alien lands, to communicate with local person. Another application which actually has been implemented during this project includes its usage in designing Content Verification System (CVS), which is used for verification of the speech data stored for different languages. As research in automatic speech recognition progresses, a language ID system would be necessary for any multi-lingual speech recognition system. One such system may be a fast information system, say at an airport, catering for multi-national clients. Another may be an automatic translation system. Both these systems would need to first recognize the language that was being spoken before they could process it.

There are number of ways to achieve this task of language ID, like based on spectral features of speech, or based on word lexicon or identifying presence of some distinct characteristics in different languages like special phonemes. Ones discussed here are based on phonetic characteristics.

2. LANGUAGE CHARACTERSTICS

2.1 Distinct Characteristics of Language

"Each language has a finite set of phonemes. As we learn our first language, we also learn to identify them. When listening to a foreign language, with phonemes not found in our first language, the presence of such sounds is readily apparent to us. Examples are the "clicks" found in some sub-Saharan African languages.

As the vocal apparatus used in the production of languages is universal, there is much overlap of the phoneme sets, and the total number of phonemes is finite. But there can be differences in the way the same phoneme is interpreted in two different languages. For example, in English, /l/ and /r/ (as in "leaf" and "reef") are two different phonemes, whereas in Japanese they are not".

On the contrary, the frequency of occurrence of phones and the phonotactic rules in languages can differ significantly. Phonotactic rules govern the way different phonemes are combined. For example, phoneme clusters /sr/ and /sp/ are quite common in Tamil and German respectively (the latter could be represented as /shp/ in English), but are rare in English. This is what we have tried to Exploit and use to design an algorithm for language identification, in this project.

2.2 Principal

Speech recognizers make it possible for the computers to understand human speech. Speech recognizers categorize vocabulary as: active vocabulary and vocabulary. Active vocabulary denotes list of words the user can be expected to say at any instant. While vocabulary denotes list of words the user may speak while working with the application.

Speech recognizer loads a set of sound reference patterns that the application expects user to say. The recognizer classifies the unknown sound and reports the best possible match with reference patterns. The probability of occurrence of a word within given acoustic observations i.e. P(W/A), is given as follows (using Baye's rule format) :

$$P(W/A) = \frac{P(A/W)\,P(W)}{P(A)}$$

where P(A/W) is called the acoustic model that estimates probability of a sequence of acoustic observations on word string W

P(W) is the language model that describes probability of a sequence of Words

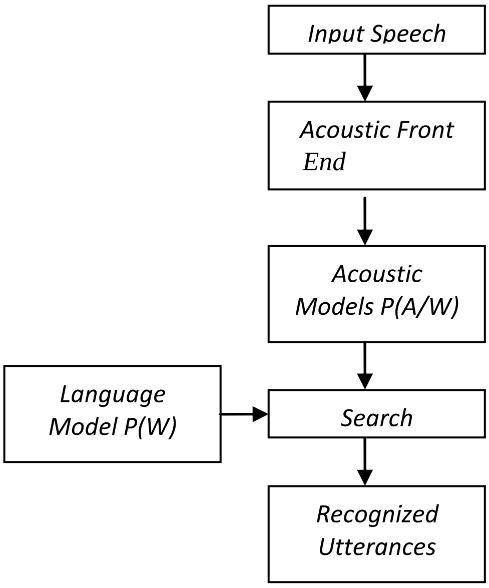


Figure 1: Speech Recognizer Model

3. LANGUAGE IDENTIFIER

Language IDs works as a single entity in many applications, but it is, in itself a set of three black boxes; front-end processing system, phoneme recognizer, and language models. Speech Data is given as an input to these set of boxes and then it flows into the system as shown in the figure. Implementation of every system is hidden from others; only interfaces are standardized as we do in case of OSI Layers of networking. By standardization, we mean the format of data, which will be passed from one system to another, is fixed.

3.1 Front-End Processing

Main purpose of front-end processing is the feature vector extraction. Many different algorithms exist for speech recognition and language identification. A common need between them is some form of parameterized representation (feature vectors) of the speech input. These feature vector streams may then be used to train or interrogate the language models which will follow the feature extraction module in a typical language identification system [2]. It is obvious that there exist an infinite number of ways to encode the speech, depending upon which particular numerical measures are deemed useful. Over the many years of speech recognition research, there has been a convergence towards a few (spectrally based) features that perform well. Of these, Linear Prediction (LP) and Cepstral measures are most widely used[3].

3.2 Phoneme Recognizer

The basic aim behind this system is to generate the phoneme sequences from the vector sequences. There are 56 phonemes, and their different combinations can represent all possible speeches in various languages.

4. LANGUAGE MODELS

Generally language models are designed in two phases: Training and recognition phase. Input given to these models is the phoneme sequences obtained from the recognizer and the output expected from them is language in which the input speech is. In the design two phases are discussed, in which system parameters are optimize.

- Training Phase
- Tuning Phase

4.1 Training Phase

Training phase is the most important phase among all the three, and decides how well or bad the whole system is going to perform. Main aim of this phase is to extract maximum possible information about a language from its training data. There are number of ways to do it. One of them is by finding the probability with which a given phoneme from a set occurs in that language. There are two issues to deal with in this probabilistic approach which are as follows:

i. **Method of finding the probabilities:** There are number of ways in which probabilities related to a phoneme can be found. One of them is P(ph|X), which represent the probability of phoneme ph occurring in language X. This value is found by counting the number of times a given phoneme occurs in training data and then dividing it by the total number phonemes in the whole data and is calculated for every phoneme.

ii. **Type of probabilities:** Now there are various ways of capturing the language specific information from the training data. While selecting the appropriate method there are some parameters you should consider. First of them is the kind of application, language models are designed for. And second is the computational resource available. Like in our case where we are trying to design a language ID, we know that it's the order in which phonemes occur, makes one language different from other and then there are some phonemes which are specific to some languages and never occur in others.

4.2 Tuning Phase

In this Phase, optimal values of all the system's hyper-parameters are found out, and all the experiments which are done in this phase are performed on tuning data which is totally disjoint from the one which is used for training and testing.

Penalty for zero probability phonemes

Consider a case when a phoneme sequence R(a b c d) occur in the testing data. Then its unigram probability according to (1) will be:

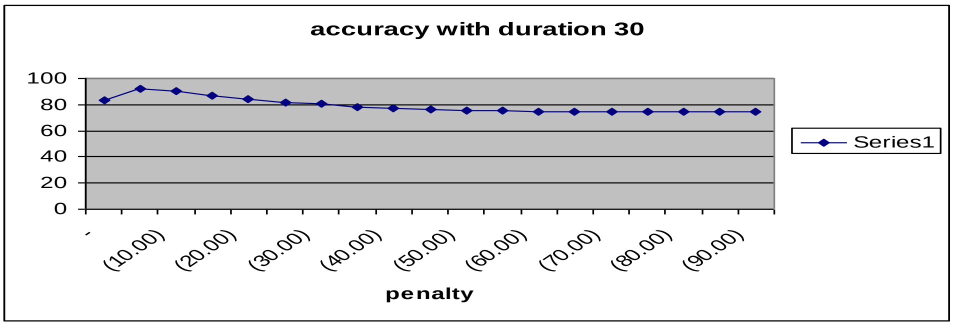
Unigram_Prob(R|X) = R(a|X)* R(b|X) *R(c|X)*R(d|X)

Now it might be possible that one of the phoneme in the test sequence actually never appeared in the whole training data of a given language X, and thus the probability of that phoneme is zero for X. Then what will happen? Now ideally we should not do anything in this regard and whenever there is such phoneme in a sequence, we should make the probability of whole sequence as zero, but the catch is that our phoneme recognizer is also not perfectly accurate so we should take a case a wrong conversion of speech to phoneme sequence. So what we should actually do is to assign a very small but non-zero value to all phonemes having zero probability. This approach has two advantages. First that now despite having a zero-probability phoneme, a given phoneme sequence is still eligible for recognition, and second because of a small value of probability it will try to bias the results in negative side, which ideally should have been the case.

Given below is graph for different values of penalty on x-axis and accuracy with which system works in y-axis. Penalty values are represented in terms of

x = exponent (r)

where r is actual penalty in terms of probability and values in x axis are its logarithmic counterpart. Now you can clearly see a peak in the graph at x= -10.



Graph 1: Penalty and accuracy

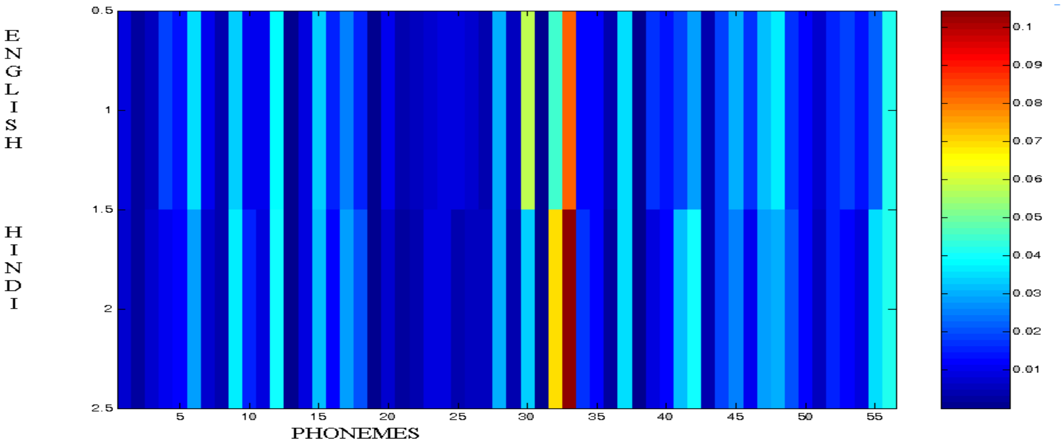
5. DISCUSSION

5.1 Recognition based on distinct phonemes

Recognition based on distinct phonemes aims at finding those phonemes which have some unique characteristics related to occurrence in different languages. There might be some phonemes which are exceptionally high probable in some language while its occurrence is rare in other language, and then a recognizer can be built from the list of such phonemes. An attempt was made to find out such phonemes in English & Hindi. For this the probability files which were generated after training phase in last section were analyzed by plotting some graphs using Matlab.

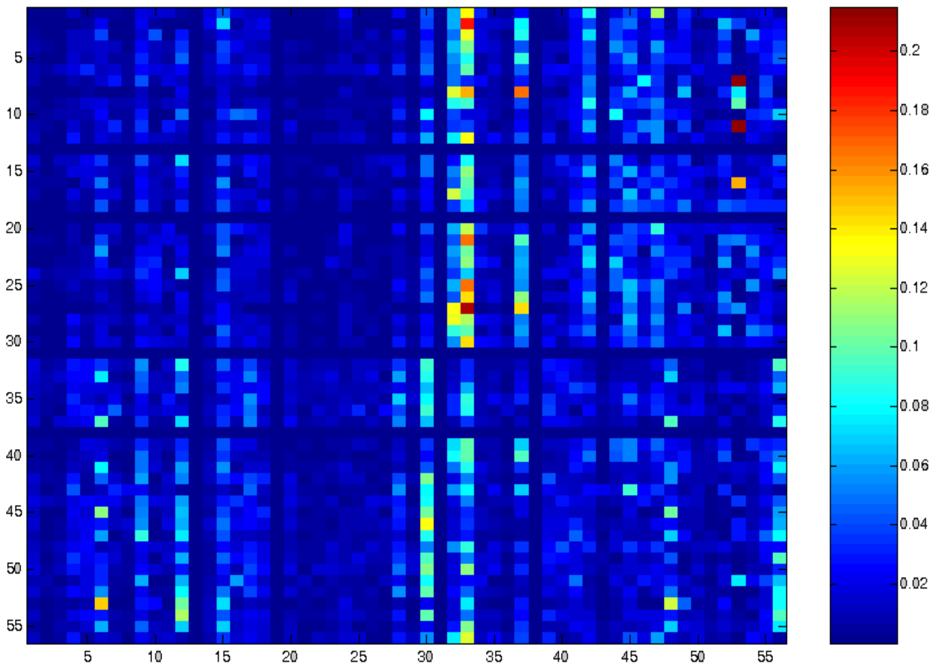
Graphs for Unigram and Bigram probabilities

Graph given below is of unigram probabilities for both English and Hindi. The probability values are mapped in to a color based on the mapping shown in the right hand, with underlying principle that higher the probability value, brighter the color will be for that phoneme. In this graph phonemes are represented in x-axis while languages are shown in y-axis.



Graph 2: Unigram and Bigram probabilities

Second graph is of bigram probability values for English. In this graph also color coding is same as used in the previous case. In this case the phoneme coming first is shown in y-axis and phoneme coming second is shown in x-axis.



Graph 3: Phonemes Probability in Bigram for English

6. CONCLUSION

The purpose of developing language ID is to make the process of language recognition mechanized and hence enable many speech based applications to use it as a black box. And the absence of absolutely correct way of recognizing languages by machine makes this field of language ID, a challenging research area.

REFERENCES

[1] Vimala c, Dr. V.Radha, “A Review on Speech Recognition Challenges and Approaches”, World of Computer Science and Information Technology Journal, 2012.
[2] Santosh K.Gaikward, Bharti W.Gawali, Pravin Yannawar, “A Review on Speech Recognition Technique”, International Journal of Computer Applications, 2010.
[3] Kuldeep Kumar, R.K.Aggarwal, “Hindi Speech Recognition System using HTK”, International Journal of Computing and Business Research, 2011.
[4] “Speech Recognition Technology Choices”, A Vocollect White Paper, 2010