INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

EUCLIDEAN DISTANCE METHOD: A TOOL FOR SPEECH RECOGNITION

AMOL R. MADANE[1], DR. UTTAM D. KOLEKAR[2]

1. Tata Consultancy Services Ltd., Pune,
2. Principal, Smt. Indira Gandhi College of Engineering, Koper-khairane, Mumbai

Accepted Date: 05/03/2015; Published Date: 01/05/2015

Abstract: The aim of the project is to develop a prototype system for 'speech recognition' technique. The algorithm creates its own dictionary of basic phonemes. Input signal is broken into segments and compared with the dictionary by Euclidean distance method to find the closest match. The proposed system is tested and found to be accurate in noise free environment.

Keyword: Euclidean, Method, Speech

Corresponding Author: MR. AMOL R. MADANE

Access Online On:

www.ijpret.com

How to Cite This Article:

Amol R. Madane, IJPRET, 2014; Volume 2 (9): 1190-1196

PAPER-QR CODE

INTRODUCTION

Speech recognition applications that have emerged over the last few years include voice dialing (e.g., Call home), call routing (e.g., I would like to make a collect call), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), domestic appliances control and content-based spoken audio search .A range of software products allows users to dictate to their computer and have their words converted to text in a word processing or e-mail document. One can access function commands, such as opening files and accessing menus, with voice instructions. Some programs are for specific business settings, such as medical or legal transcription.

Speech recognition is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words. The recognized words can also serve as the input to further linguistic processing in order to achieve speech understanding.

Some systems require speaker enrollment--a user must provide samples of his or her speech before using them, whereas other systems are said to be speaker-independent. Recognition is generally more difficult when vocabularies are large or have many similar-sounding words.

When speech is produced in a sequence of words, language models or artificial grammars are used to restrict the combination of words. The simplest language model can be specified as a finite-state network, where the permissible words following each word are given explicitly.

Figure below shows the major components of a typical speech recognition system. The digitized speech signal is first transformed into a set of useful measurements or features at a fixed rate, typically once every 10--20 msec (.05-.1 kHz) These measurements are then used to search for the most likely word candidate, making use of constraints imposed by the acoustic, lexical, and language models. Throughout this process, training data are used to determine the values of the model parameters.

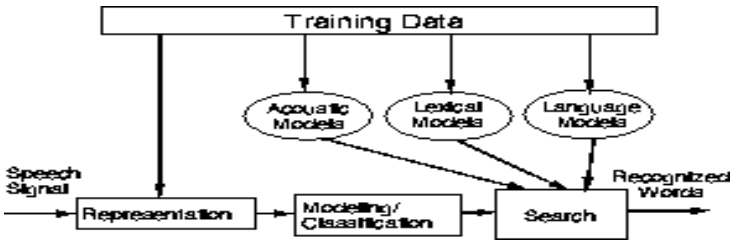


Figure 1: Components of a typical speech recognition system

Literature Survey-

Speech recognition systems attempt to model the sources of variability in several ways. At the level of signal representation, researchers have developed representations that emphasize important speaker-independent features of the signal, and de-emphasize speaker-dependent characteristics. At the acoustic phonetic level, speaker variability is typically modeled using statistical techniques applied to large amounts of data. Speaker adaptation algorithms have also been developed that adapt speaker-independent acoustic models to those of the current speaker during system use. Effects of linguistic context at the acoustic phonetic level are typically handled by training separate models for phonemes in different contexts; this is called context dependent acoustic modeling. Word level variability can be handled by allowing alternate pronunciations of words in representations known as pronunciation networks. Statistical language models, based on estimates of the frequency of occurrence of word sequences, are often used to guide the search through the most probable sequence of words.

The dominant recognition paradigm in the past fifteen years is known as hidden Markov models (HMM). An HMM is a random variable model, in which the generation of the underlying phoneme string and the frame-by-frame acoustic realizations are both represented probabilistically as Markov processes. Neural networks have also been used to estimate the frame based scores; these scores are then integrated into HMM-based system architectures, in what has come to be known as hybrid systems. These methods involve complex mathematical functions, but essentially, they take the information known to the system to figure out the information hidden from it.

Algorithm-

In this model, each phoneme [the smallest sound units of which words are composed] is like a link in a chain and the completed chain is a word. However, the chain branches off in different directions as the program attempts to match the digital sound with the phoneme that's most likely to come next. During this process, the program assigns a probability score to each phoneme, based on its built-in dictionary and user training.

Program training: If a program has a vocabulary of 60,000 words (common in today's programs), a sequence of three words could be any of 216 trillion possibilities. Even the most powerful computer can not search through all of them without some help. That help comes in the form of program training. These statistical systems need lots of exemplary training data to reach their optimal performance - sometimes on the order of thousands of hours of human-transcribed speech and hundreds of megabytes of text. These training data are used to create

acoustic models of words, word lists, and multi-word probability networks which are mentioned above. These details can make the difference between a well-performing system and a poorly-performing system -- even when using the same basic algorithm. While the software developers who set up the system's initial vocabulary perform much of this training, the end user must also spend some time training it. They must also train the system to recognize terms and acronyms particular to the business setting.

Characteristics-

1. Weaknesses and Flaws

No speech recognition system is 100 percent perfect; several factors can reduce accuracy. Some of these factors are issues that continue to improve as the technology improves. Others can be lessened -- if not completely corrected -- by the user.

2. Low signal-to-noise ratio

The program needs to "hear" the words spoken distinctly and any extra noise introduced into the sound will interfere with this. A quiet environment is suitable. Low-quality sound cards, often pick up noise from the electrical signals produced by other computer components.

3. Overlapping speech

Current systems have difficulty separating simultaneous speech from multiple users.

4. Intensive use of computer power

Running the statistical models needed for speech recognition requires the computer's processor to do a lot of heavy work. The vocabularies needed by the programs also take up a large amount of hard drive space.

Proposed algorithm and its discussion-

The audio signal of a particular word has the shape of its waveform independent of the voice i.e. a word spoken by different persons are similar in terms of the shape of waveforms. The algorithm makes use of this fact where it first creates its own ‘Dictionary’, where the basic phonemes are stored. The input word is then broken into approximate segments and each segment is compared with the reference phonemes to find the closest match. To achieve more statistical precision, the segments are shifted and compared in a looped manner. The combination of highest scoring segments is then declared as the recognized word.

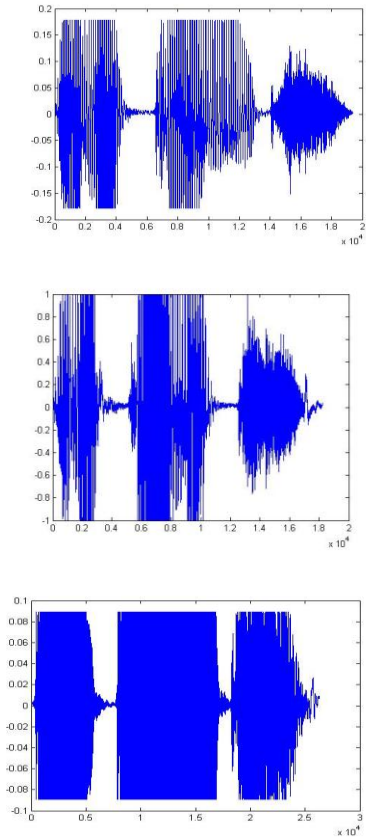


Figure 3: Waveform-"Electronics" by User1, 2, 3

Creation of Directory-

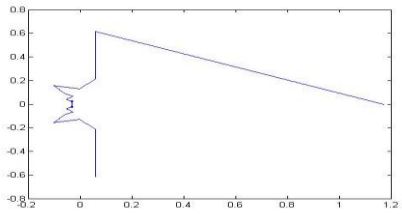
The selected reference words are broken into segments .Each of these segments represents the basic 'phoneme' for the system. (e.g. good morning ='go'+'od'+'mo'+'or'+'ning'). Each reference segment is put under noise removal and gain control. These audio signal segments are sampled using the MATLAB function 'wavread'. These are stored as matrices of sampled values in double precision format (values range from -1 to 1). Using FFT algorithm, these sampled values are transformed to the frequency domain.

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi i}{N} kn}$$

The dictionary now has a set of matrices in which each column corresponds to a phoneme represented in the frequency domain.

Sampling the Input-

The input audio signal is also broken into divisions, each division being of length approximately of that of a basic phoneme. Each of these segments is then sampled using 'wavread' for number of values equal to that used for the dictionary. Again FFT is performed.



The input signal may not always be of the same length as that of the reference word. To overcome this variability, sampling segments are shifted and processed several times in a loop.

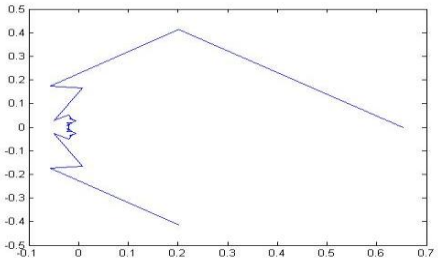


Figure 6: Plot-phoneme "el" Samples:40-80

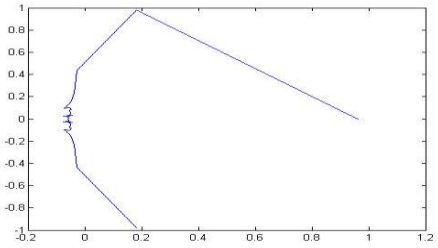


Figure 7: Plot-user1 input "Electronics"samples:1-40

Comparison-

The Euclidean distance (squared) between the input vector and reference vector is found out. The reference giving minimum distance is considered as a match.

Recognize the Word

This stage the phonemes with closest match to the input are known. A combination of these phonemes, as stored in the dictionary gives the resultant word.

For better accuracy the input signal should be as noise-free as possible. Also the gain of the signal should be within specific limits.

CONCLUSION-

This project successfully implements a simple algorithm for 'speech recognition' suitable for a small database. Different words spoken by different users were tested and recognized correctly. The performance was observed to be 70% accurate with false alarming around 15%. The false alarming can be reduced if the score calculating technique is made more precise i.e. increment in the score should be proportional to the proximity of the match. Also increase in the number of samples renders better results; at the same time it also increases the runtime.

REFERENCES-

1. Tor M. Aamodt, "A Simple Speech Recognition Algorithm for ECE341", April 15, 2003

2. Renato De Mori & Fabio Brugnara, Survey of the State of the Art in Human Language Technology, 2006.

3. Douglas G. Danfowth & David R. Rogosa, Äpplications of Learning Models to Speech Recognition over Telephone", Learning Research and Development Center, University of Pittsburgh, 2009.