Trie Based Approach For Isolated Word Speech Recognition

G.Madhavi [1], T.Jalaja [2]
1 Assistant Professor, Dept. of IT, SCETW, Abids, Hyderabad, India
2 Assistant Professor, Dept. of CSE, VCE, Ibrahimbagh, Hyderabad, India

Abstract:

The Speech recognition is more commonly known as Automatic Speech Recognition (ASR) which is defined as the building of system for mapping acoustic signals to a string of words. Most Indian languages are syllabic in nature, hence syllable unit and its organization plays an important role while searching in recognition process. In Isolated Word Recognition (IWR) each word is surrounded by pause and it consists of limited vocabulary and is modeled as sequence of syllables.

In this Paper we use Trie based approach for IWR. A Trie based approach is a method for matching an input sequence of syllables to one of the spoken words which are modeled by concatenating members of a set of syllables. A Trie is an ordered tree data structure that has been used for various applications, such as the construction of natural language dictionaries, the research of words to a compiler, database systems etc. In this structure, all the descendants of any one node have a common prefix associated with that node, while the root is represented by an empty node. It allows fast pattern matching by finding the longest match of a given signal. The advantage of Trie is it provides flexibility towards modification while new word models are included. Trie data structure is used to represents plurality of objects for identifying characteristics of the respective syllable units.

Keywords: Trie data structure, Automatic Speech Recognition, Isolated Word Recognition

1. Introduction

Nowadays, computer systems play a major role in our lives. They are used everywhere beginning with homes, offices, restaurants, gas stations, and so on. Communicating with a computer is done using a keyboard or a mouse, devices many people are not comfortable using. Speech recognition solves this problem and destroys the boundaries between humans and computers. Using a computer will be as easy as talking with your friend. Speech refers to the recognition and production of spoken language. Speech technology, at its most basic, includes one technology for recognition, known as automatic speech recognition (ASR).Speech recognition is the diagnostic task of recovering the words that produce a given acoustic signal. In other words, it is the problem of transforming a digitally-encoded acoustic signal of a speaker talking in a natural language (e.g., English) into text in that language. Given the uncertainty at many levels of this problem (e.g., introduced by background noise, digitization noise, speaker's accent), we can formally specify the problem as

$$\mathrm{argmax}_{\mathbf{words}} P\,(\mathbf{words} \mid \mathbf{signal})$$

where **words** is a string of words in a given natural language like English, and **signal** is a sequence of observed acoustic data that has been digitized and pre-processed.

This Paper is aimed at Isolated Word Speech Recognition. In Isolated Word Recognition (IWR) each word is surrounded by pause and it consists of limited vocabulary and is modeled as sequence of syllables.

Any recognition process requires selecting small units for recognition. The basic units for recognition can vary from phone to sentence level. The amount of data for training depends on the unit size selected. Our approach uses syllable as the basic unit. Syllabication is the breaking up of words into smaller pieces. Syllables units make the recognition process easy.

The trie is a data structure that can be used to do a fast search in a large text and that stores the information about the contents of each node in the path from the root to the node, rather than the node itself. A trie is a tree structure (or directed acyclic graph), the nodes of which represent letters in a word.

A *Trie* (short for re*Trie*val) is a multiway tree that is used for efficient searching. The idea is much like a thumb index dictionary; if you place your thumb in the cutout of the dictionary for a particular letter, you go to directly to that section of the dictionary, and can then locate all words beginning with that letter. Searching and traversing m-way trees using tries is a well known and broadly used technique.

We will use the dictionary example to analyze tries. A dictionary trie consists of nodes that have a branching factor of 27.

Tries can be implemented in a number of ways. The figure below shows a trie of the 31 most common words in English organized as a forest of trees. The table below is a trie of the same data organized in tabular form. Each letter of the alphabet is along the left side column of the table, and if we are searching for the word HER we go to the letter H entry in the first column of the trie, which tells us to go to column (5) to find the branch for the second letter in the word we are looking for which is E. Column 5 is also a 27-way branch node, and it tells us that the branch for E is column 11, where the entry for R - the third letter of the word we are looking for is the node we are looking for. For this 3 letter word, we took 3 branches.

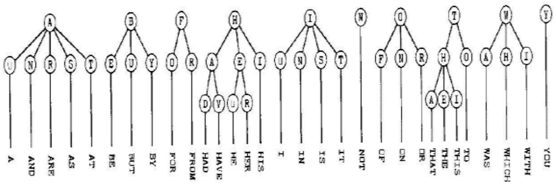
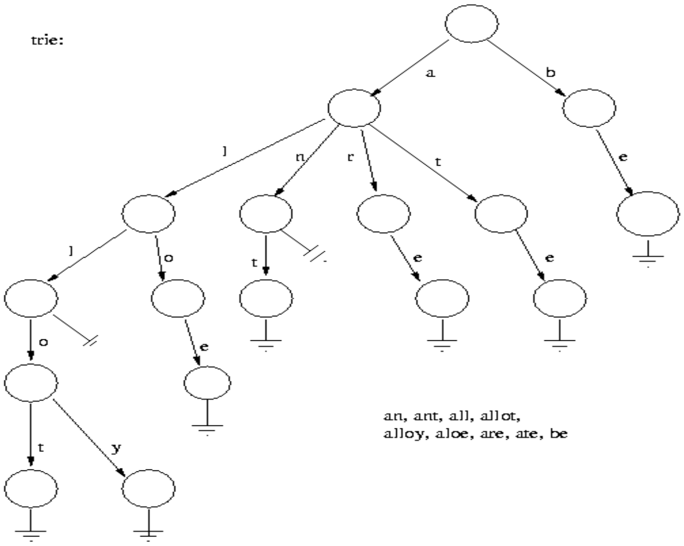


Figure 1.1: TRIE as a forest of trees for the 31 most common English words

The extra entry in each node (27th entry for a 26 letter alphabet) is kind of a NULL pointer that is used to signify that the path from the root to that node forms a legal entry in the trie. In the table, the word HE is found by having an entry in the null field of column 11. This is needed since HE is an entry we want to find, but the trie needs to keep indexing words that continue beyond the 2 letters HE such as HER or HEALTH or HEAT etc.A *trie* (from re**trie**val), is a multi-way tree structure useful for storing strings over an alphabet. It has been used to store large dictionaries of English (say) words in spelling-checking programs and in natural-language "understanding" programs. Given the data: an, ant, all, allot, alloy, aloe, are, ate, be

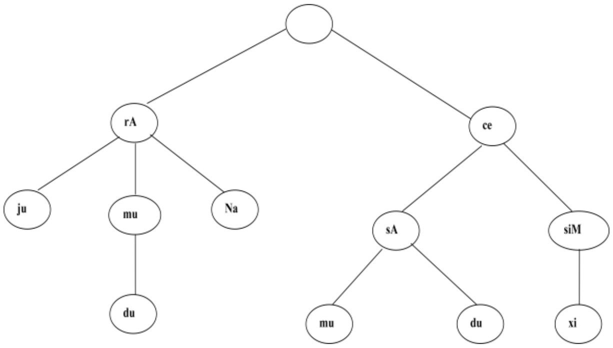
The corresponding trie would be as follows:



The idea is that all strings sharing a common stem or *prefix* hang off a common node. When the strings are words over {a..z}, a node has at most 27 children - one for each letter plus a terminator. The elements in a string can be recovered in a scan from the root to the leaf that ends a string. All strings in the trie can be recovered by a depth-first scan of the tree.

Example for Telugu words:

For Strings : {rAju, rAmudu, rAna,cesAmu, cesadu, cesiMxi}. The corresponding Trie is as follows:



1.1 Existing System

We have speech recognition systems developed in English language. Now current research is gearing up in other languages such as Telugu, Hindi, Bengali etc,. For most of the Indian languages a full fledged system need to be developed. This paper aims at building recognition system for Telugu language.

1.2 Proposed System

The idea of this paper is to search the possible word efficiently by using Trie data structure. A trie node, named for its successful use in information retrieval, is an array of pointers, one for each character in an alphabet. Each leaf node is the terminus of a chain of trie nodes representing a string. For string management, tries are fast with reasonable worst-case performance.

1.3 Objective of Proposed System

The main objective of this project is to **reduce the search space** while recognizing the words and improve the performance using Trie based data structure.

1.4 Challenges in Proposed System

➢ Identifying the syllable boundaries is one challenge.

➢ Applying constructive model is one more challenge.

➢ Finding optimal pattern matching algorithms and minimizing misrecognitions is another challenge

1.5 Advantages with Proposed System

➢ The main advantage is that it initially recognizes the syllable which is nearest syllable of possible word.

➢ The search space is reduced while recognizing the syllable.

➢ By using Tries for Isolated word speech recognition requires less space when they contain a large number of short strings, because the keys are not stored explicitly and nodes are shared between keys with common initial subsequences.

1.6 Applications of the proposed System

➢ To build efficient Isolated Word Speech Recognition system.

➢ To increase the vocabulary size.

1.7 Organization of the topics

This paper presents an architecture and implementation of a “Trie based approach for Isolated Word Speech Recognition”, which uses syllable as basic unit for Recognition. Section 1 discuss about the introduction to Paper, Section 2 discuss about the Introduction to Automatic Speech Recognition, background work of speech Recognition, Basic steps in ASR, Approaches to ASR ,Data Structures and the Tools used.

Section 3 describes the design details of the proposed system. Section 4 discusses about the implementation and results.

2. Automatic Speech Recognition

Speech recognition (also known as automatic speech recognition or computer speech recognition) converts spoken words to text. Automatic Speech recognition (ASR) is the process by which a computer identifies spoken words. Speech recognition is the technology that makes it possible for a computer to identify the components of human speech. The process can be said to begin with a spoken utterance being captured by a microphone and to end with the recognized words being output by the system.

2.1 Introduction to Automatic Speech Recognition

From human prehistory to the new media of the future, speech communication has been and will be the dominant mode of human social bonding and information exchange. The spoken word is now extended, through technological mediation such as telephony, movies, radio, television, and the Internet. This trend reflects the primacy of spoken communication in human psychology. In addition to human-human interaction, this human preference for spoken language communication finds a reflection in human-machine interaction as well. Most computers currently utilize a graphical user interface (GUI), based on graphically represented interface objects and functions such as windows, icons, menus, and pointers. Most computer operating systems and applications also depend on a user's keyboard strokes and mouse clicks, with a display monitor for feedback. Today's computers lack the fundamental human abilities to speak, listen, understand, and learn. Speech, supported by other natural modalities, will be one of the primary means of interfacing with computers. And, even before speech based interaction reaches full maturity, applications in home, mobile, and office segments are incorporating spoken language technology to change the way we live and work.

A spoken language system needs to have speech recognition capability. However, the component by themselves is not sufficient to build a useful spoken language system. An understanding and dialog component is required to manage interactions with the user; and domain knowledge must be provided to guide the system's interpretation of speech and allow it to determine the appropriate action. For all these components, significant challenges exist, including robustness, flexibility, ease of integration, and engineering efficiency.

We feel that speech recognition is important, not because it is 'natural' for us to communicate via speech, but because in some cases, it is the most efficient way to interface to a computer.

The main goal of speech recognition is to get efficient ways for humans to communicate with computers.

2.2 Basic Steps in ASR

There are three basic steps in the ASR, they are 1) parameter estimation (in which the test pattern is created) 2) parameter comparison 3) decision making. The function of the parameter measurement is to represent the relevant acoustic events in the speech signal in terms compact efficient parameters. Although the choice of which parameters to use is depends on other considerations (e.g. computational efficiency type of implementation and available memory)

A wide range of possibilities exists for parametrically representing the speech signal these include short time energy, zero crossing rate and other related parameters. The most important parametric representation of the speech is related to spectral envelope. Modern speech coding algorithms are based on utilization of frequency masking properties of human hearing. A central result from the study of the human speech perception is the importance of slow changes in speech spectrum for speech intelligibility. A second key to human speech recognition is the integration of phonetic information over relatively long intervals of time.

The basic and most widely used parametric representation are linear predictive coding (LPC), Perceptual Linear Predictive coding (PLP), RASTA (RelAtive SpecTrA) and Mel-frequency cepstrum (MFCC).

2.3 Approaches to Speech recognition

The transformation of speech into feature vectors is followed by the process of recognizing what was actually spoken. There are several approaches to this problem. These include: template matching, knowledge-based approaches, stochastic approaches and connectionist approaches. These methods are not mutually exclusive.

2.4 Data Structures

Data structures are some objects that we generate to store data in them, these data can be more than one variable (Arrays can be considered as built in data structures) They also contain algorithms to process those stored data.

In computer science, a **data structure** is a particular way of storing and organizing data in a computer so that it can be used efficiently. Different kinds of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks. For example, B-trees are particularly well-suited for implementation of databases, while compiler implementations usually use hash tables to look up identifiers.

Data structures are generally based on the ability of a computer to fetch and store data at any place in its memory, specified by an address — a bit string that can be itself stored in memory and manipulated by the program. Thus the record and array data structures are based on computing the addresses of data items with arithmetic operations; while the linked data structures are based on storing addresses of data items within the structure itself. Many data structures use both principles, sometimes combined in non-trivial ways.

The implementation of a data structure usually requires writing a set of procedures that create and manipulate instances of that structure. The efficiency of a data structure cannot be analyzed separately from those operations.

This observation motivates the theoretical concept of an abstract data type, a data structure that is defined indirectly by the operations that may be performed on it, and the mathematical properties of those operations (including their space and time cost).

2.5 Tools Explored

This project is developed using following technologies:

- Praat tool: Recording
- Java Language

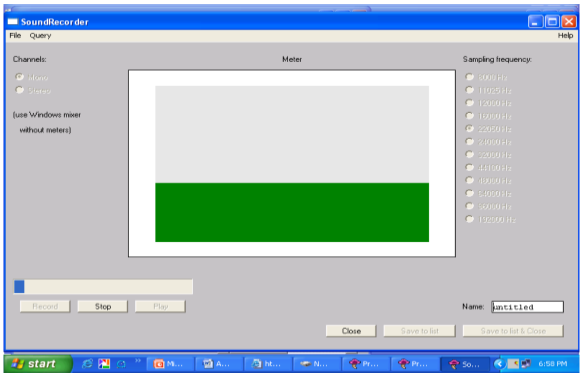


Figure 2.1: Recording a mono sound in Praat tool

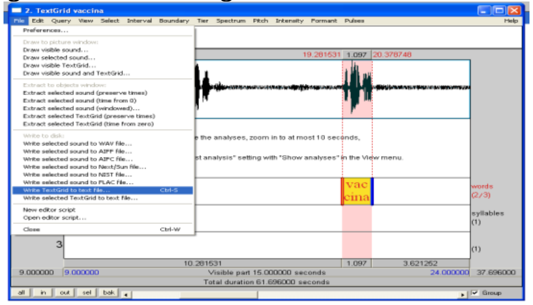

2. TextGrid vaccina
File
Edit
Query
View
Select
Interval
Boundary
Tier
Spectrum
Pitch
Intensity
Formant
Pulses
Help
Preferences...
Draw to picture window:
Draw visible sound...
Draw selected sound...
Draw visible TextGrid...
Draw visible sound and TextGrid...
Extract to objects window:
Extract selected sound (preserve times)
Extract selected sound (time from 0)
Extract selected sound (windowed)...
Extract selected TextGrid (preserve times)
Extract selected TextGrid (time from zero)
Write to disk:
Write selected sound to WAV file...
Write selected sound to AIFF file...
Write selected sound to AIFC file...
Write selected sound to Next/Sun file...
Write selected sound to NIST file...
Write selected sound to FLAC file...
Write TextGrid to text file... Ctrl-S
Write selected TextGrid to text file...
New editor script
Open editor script...
Close Ctrl-W
19.281531
1.097
20.378748
e the analyses, zoom in to at most 10 seconds,
st analysis" setting with "Show analyses" in the View menu.
vac
cina
words
(2/3)
syllables
(1)
3
(1)
10.281531
1.097
3.621252
9.000000
9.000000
Visible part 15.000000 seconds
24.000000
37.696000
Total duration 61.696000 seconds
all
in
out
sel
bak
Group

Figure 2.2: Saving the textgrid file to textfile with .TextGrid extension

3. System Design

3.1 Design of the System

In the proposed system there are two modules, where the first module trains the system and builds the models at syllable unit. In the second module it process the new input, identify the syllable units and recognize them and identify the word. The block diagram of the system is shown in the figure 3.1

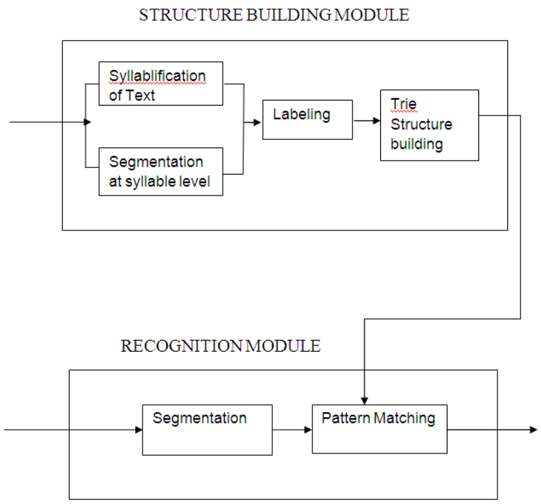


Figure 3.1 Proposed System

3.2 Design of Trie Data Structure

A trie is a compact data structure for representing a

- set of strings, such as all the words in a text.
- String decomposes into sequence of letters

3.3 Node structure in Trie

Variables of Node:

Name of syllable – String

Bit field - Integer

Model - String[]

Wav file - String

Node down

Node right

Operations performed:

- Insertion
- Comparison
- Display

Insertion: The word models built are stored in Trie structure.

Comparison: While testing the stored models of trie structure are compared with test samples.

Display : Recognized words are displayed.

3.4 Recognition

The word to be recognized is given as input to the system in the form of syllable units. The system takes first syllable of given word and finds distance with first syllables of all words in trie structure, whichever is minimum it considers that syllable and takes that path, then it considers second syllable of given word and compare with second syllable of chosen path if it matches it continues the process till the last syllable of given word, otherwise it chooses different path(i.e different syllable from trie structure).

4. CONCLUSION

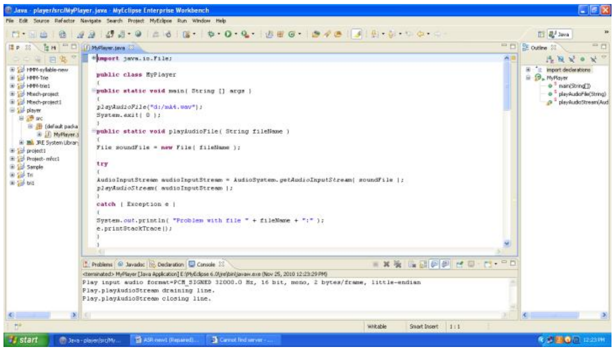
A Trie based approach is used for matching an input sequence of syllables to one of the spoken words which are modeled by concatenating members of a set of syllables. The input to the system is a word in form of syllable units. The vocabulary size is 10 words for the recognition. In Trie data structure node consists of name of syllable, link to acoustic signal and successor node information. To identify a segment of the input sequence with the element of the selected node that generates the acceptance value, the succeeding search is performed. At leaf node, sequence from root to leaf is concatenated and the corresponding word with minimum distance is the corresponding word for the given input.

Algorithms implemented in this for getting the required results are:

- MFCC Algorithm
- Segmental K-means Algorithm
- Trie Data Structure

- Forward algorithm

The following are the screenshots of the results:


Java - player/src/MyPlayer.java - MyEclipse Enterprise Workbench
File Edit Source Refactor Navigate Search Project MyEclipse Run Window Help
MyPlayer.java
HMM-syllable-new
HMM-Trie
HMM-trie1
Mtech-project
Mtech-project1
player
src
(default packa
MyPlayer.j
JRE System Library
project1
Project- mfcc1
Sample
Tri
tri1
Outline
import declarations
MyPlayer
main(String[])
playAudioFile(String)
playAudioStream(Aud
Problems Javadoc Declaration Console
Writable Smart Insert 1:1
start Java - player/src/My... ASR-new1 (Repaired)... Cannot find server - ...
12:23 PM

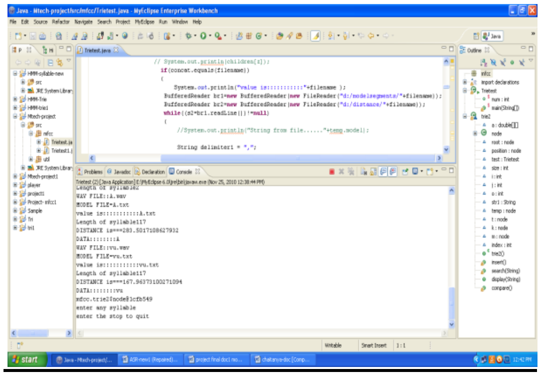
```
import java.io.File;

public class MyPlayer
{
public static void main( String [] args )
{
playAudioFile("d:/mk4.wav");
System.exit( 0 );
}
public static void playAudioFile( String fileName )
{
File soundFile = new File( fileName );

try
{
AudioInputStream audioInputStream = AudioSystem.getAudioInputStream( soundFile );
playAudioStream( audioInputStream );
}
catch ( Exception e )
{
System.out.println( "Problem with file " + fileName + ":" );
e.printStackTrace();
}
}
```

```
<terminated> MyPlayer [Java Application] E:\MyEclipse 6.0\jre\bin\javaw.exe (Nov 25, 2010 12:23:29 PM)
Play input audio format=PCM_SIGNED 32000.0 Hz, 16 bit, mono, 2 bytes/frame, little-endian
Play.playAudioStream draining line.
Play.playAudioStream closing line.
```

Figure 4.1: Playing an Audio file



```
                    // System.out.println(children[z]);
                     if(concat.equals(filename))
                     {
                        System.out.println("value is:::::::::::"+filename );
                     BufferedReader br1=new BufferedReader(new FileReader("d:/modelsegments/"+filename));
                     BufferedReader br2=new BufferedReader(new FileReader("d:/distance/"+filename));
                     while((s2=br1.readLine())!=null)
                     {
                        //System.out.println("String from file......"+temp.model);

                        String delimiter1 = ",";
```

```
Trietest (2) [Java Application] E:\MyEclipse 6.0\jre\bin\javaw.exe (Nov 25, 2010 12:38:44 PM)
Length of syllable2
WAV FILE::A.wav
MODEL FILE=A.txt
value is:::::::::::A.txt
Length of syllable117
DISTANCE is===283.5017108627932
DATA::::::::A
WAV FILE::vu.wav
MODEL FILE=vu.txt
value is:::::::::::vu.txt
Length of syllable117
DISTANCE is===167.96373100271094
DATA::::::::vu
mfcc.trie2$node@1cfb549
enter any syllable
enter the stop to quit
```

Figure 4.2: Showing Trie structure for word 'A-vu'

REFERENCES

1. Lecture notes of Research and Training Unit for Navigational Electronics on ASR, 27th - 29th August 2009, OU, Hyderabad.

2. Fundamental of Speech Recognition by L.R. Rabiner and B.H. Juang.

3.A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE, vol. 77, no. 2, pp. 257–286, 1989 by L.R.Rabiner.

4.www.cs.brown.edu/research/ai/dynamics/tutorial/Documents/HiddenMarkov Models.html

5. Speech and Language Processing, Pearson Education Asia by Daniel Jurafsky and James H. Martin (2002).

6.http://en.wikipedia.org/wiki/Speech_recognition

7.http://www.cs.columbia.edu/~allen/F09/NOTES/tries.pdf

8.http://en.wikipedia.org/wiki/Trie

9.http://www.essays.cc/free_essays/b3/nyr219.shtml

10.http://www.electronicsforu.com/EFYLinux/efyhome/cover/Jan2003/speech-recognition.pdf