Abstract

The new interactive software called Tamil quick Writer will easily convert spoken Tamil speech into Tamil text format. This project has been implemented using object oriented methodology. We used Unified Modeling Language (UML) to design our system and Hidden Markov Toolkit (HTK) to train and recognize the user voices. This system captures the known user voices and unknown user voices. These voices are filtered by the Microphone of frequency - 20 to 20000Hz (audible range of human ear). Known user voices are used to train the system and unknown user voices are used to recognize the system. At the end of the training process we obtained trained HMM. The trained HMM was used to recognize unknown user voices. These recognized words are converted in to Tamil text format using Tamil True Type fonts. We used C#.NET environment to implement our system. The user should have to install .Net frame work 1.1, HTK 3.2.1 and any Tamil true type font before starting the system, but we are providing these installations for our users. Any user who speaks Tamil can use this system for Tamil text conversion. A Pentium II computer will suffice this system. The system provides more facilities to the user(e.g changing font size, font color etc...). It also provides room for further investigations related to the project.

Main features of our System

- Speaker independent
- Continuous Speech Recognition
- Transfer the Tamil voice into Tamil text format.
- Work out recognition and text conversion simultaneously.

Advantages

- Easy to get exact information from others and document them (e.g.: Journalism).
- Over come misinterpretation of speeches, verdicts.
- System is safe and highly secure.
- Reduces man power and workloads.
- We can save important speeches for prolonged period.
- Disabled people can easily express their ideas and obtain it in word as a documented format.

Introduction

Our Project will give an analysis, on speech training and recognition, via the particular system “Tamil Quick writer”, which will convert Tamil speech to Tamil text format. The aim of the project is to introduce new intelligence based software to transcript the Tamil speech in Tamil text format. Having carried out a comprehensive literature survey in Hidden Markov Model (HMM) technique, we identified some toolkits available for speech recognition out of which we found the best tool is HTK.
Problems Encountered
- Pronunciation differs from person to person and place to place.
- Difficult to represent the digits E.g. saying a number in digit (124) will be printed as ‘onetwofour’
- Training process will be supported only through user voices recorded via HSlab.
- Difficult to transcribe Tamil font using Unicode.
- Difficult to use Tamil true type font directly in HTK.

Steps involved in second phase: [see: Fig1]
- Getting unknown user voices.
- Recognizing those voices using trained HMM.
- Using text conversion to get the expected out put.

System overview

![Diagram of System overview](image)

Our System consists of two main phases, namely trained voices and recognized voices. Each phase has a number of steps to evaluate our project.

Steps involved in first phase: [see: Fig1]
- Getting different voices from different area people with different age limit via micro phone (known user voices).
- Recording voices using HTK Hslab tool.
- Training those voices using HTK training tools.
- Finally creating Trained HMM.

![Diagram of Steps involved in first phase](image)

We trained our system with some user voices, which are called known user voices and voices which we didn’t use for training are called unknown user voices. These unknown user voices should be recognized by our system. There are some procedures to recognize the unknown user voices (Fig: 2).

Those are,
- Open the application and start to speak in Tamil via a microphone.
- After finishing the speech stop the cmd process.
- Every pronunciation in the series of speech is checked with the trained HMM.
- Recognized Tamil words are obtained in English text format.
- English text format is converted into Tamil text format using Tamil True Type fonts.
Research methodology

I) Training and Recognition
We developed our system with the aid of the HTK book and some downloaded documents related to speech recognition. There are two major processing stages involved in HTK.

i) Training: -
HTK training tools are used for training process. These tools are used to estimate the parameters (E.g. mean, variance, etc.) of a set of hmms using training utterances and their associated transcriptions.

E.g: HERest – does the embedded training for a whole set of hmms.

On the Training set
------ Overall Results ----------------------------------------------
SENT: %Correct=99.76 [H=424, S=1, N=425]
WORD: %Corr=99.81, Acc=99.81 [H=8539, D=13, S=3, I=0, N=8555]

ii) Recognition: -
HTK recognition tools are used for recognizing the unknown voices.

E.g: HVite – gives the ready on the DOS prompt by using configuration file, word net file and dictionary file.

On the Testing set
------------------ Overall Results ------------------------------
SENT: %Correct= 41.67[H=5, S=7, N=12]
WORD: %Corr=88.24, Acc=11.76 [H=30, D=0, S=4, I=26, N=34]

The line starting with SENT: indicates that of the 12 test utterances, 5 (41.67%) were correctly recognised. The following line starting with WORD: gives the word level statistics and indicates that of the 34 words in total, 30 (88.24%) were recognised correctly. There was 0 deletion error (D), 4 substitution error (S) and 26 insertion error (I). The accuracy figure (Acc) of 11.76%.

II) Implementation
To build up our proposed system, we went through the following steps

1. Implement prerequisite installation components
   a. Install .Net framework 1.1
   b. Install HTK tool kit
   c. Install Tamil Fonts

2. Design Tamil lexicon
   a. Nouns lexicon
   b. Pronouns lexicon
   c. Numerals lexicon
   d. Action Verb lexicon etc.

3. Correlate HTK training model with user interface
   a. Set users machine environment variable
   b. Call HVite through end user application

4. Evaluate HVite output and identify possible words

5. Convert Tamil possible words (which come as English letters) to Tamil letters.

6. Do error handling; give proper error message to the system users.

7. Designing extra features for users convenience. E.g. Change font, save the output, modify changes, help document, print the output, etc.

Interface design

Fig: 3 indicate the Tamil text which comes from Tamil speech with error handling.
Future developments

- We will try to overcome some of the difficulties we came across.
- Try to complete the text translation as we planned in the second phase of our project. We have found a way of translating Tamil sentence to English sentence using the following steps.
  - Identify Tamil Language structure and identify its grammar patterns.
  - Identify English Language structure and identify its grammar patterns.
  - Associate English Structure with that of the Tamil to find out the salient commonness and differences.
  - Prepare a bilingual transfer dictionary.

E.g. அழகு வந்தே வாணியாடு You went to the market.

- We will try to develop our system with more facilities and add more words in Tamil for training process.
- Take more voices from people in different area.

Conclusion

Our system is user friendly and it helps people who face problems in the conversion of Tamil speech to Tamil text format. This project is an outcome of our dedicated one year effort with out the help of any previous models. We had to refer a lot of books, articles and go through trivial experiences. We understood the benefits of team work and the value of perseverance through this quality project. We reached our target with in our scheduled budget and time.

References


[6] Dr. Krishnamurthy, Mr. Elangovan and Mr. P. Chellappan, Kanithamizh Sangam, Association for Tamil Computing, India, http://www.kanithamizh.org/


[13] Transliteration of Tamil to English for the Information Technology ‘jAzhan’ Ramalingam Shanmugalingam (’appuAcci’) <appuacci@san.rr.com> San Diego CA 92131 USA.

[14] Automatic Speech Recognition (ASR) for Developing Regions Chuck Wooters wooters@icsi.berkeley.edu International Computer Science Institute