# Indian Institute of Technology, Guwahati



Department of Computer Science and Engineering Project report

On

# “Speech based Tic-Tac-Toe Game”

Based on

Speech recognition system

Course: CS-566 Speech Processing

**Submitted to** Prof. P. K. Das

**Submitted by:**

Mayank Singh Parmar (214101027)

Amit Pawar (214101006)

**TABLE OF CONTENT**

1. Abstract
2. Introduction
3. Proposed Methodology
4. Experimental Setup
5. Result

## ABSTRACT

This document defines a set of evaluation criteria and test methods for speech recognition systems used to play a Tic-Tac-Toe Game.

## INTRODUCTION

In this report we concentrate on the speech recognition programs that are human-computer interactive. When software evaluators observe humans testing such software programs, they gain valuable insights into technological problems and barriers that they may never witness otherwise. . Testing speech recognition products for universal usability is an important step before considering the product to be a viable solution for its customers later. This document concerns Speech Recognition accuracy in playing Tic-tac-Toe game which is a critical factor in the development of hands-free human- machine interactive devices. There are two separate issues that we want to test: digit and word recognition accuracy and software friendliness. Major factors that impede recognition accuracy in the environment noise sources and system noise.

But, what is speech recognition?

Speech recognition works like this. You speak into a microphone and the computer transforms the sound of your words into text to be used by your word processor or other applications available on your computer. The computer may repeat what you just said or it may give you a prompt for what you are expected to say next. This is the central promise of interactive speech recognition. You also had to correct any errors virtually as soon as they happened, which means that you had to concentrate so hard on the software that you often forgot what you were trying to say.

The new voice recognition systems are certainly much easier to use. You can speak at a normal pace without leaving distinct pauses between words. However, you cannot really use “*natural speech*” as claimed by the manufacturers. You must speak clearly, as you do when you speak to a Dictaphone or when you leave someone a telephone message. Remember, the computer is relying solely on your spoken words. It cannot interpret your tone or inflection, and it cannot interpret your gestures and facial expressions, which are part of everyday human communication. Some of the systems also look at whole phrases, not just the individual words you speak. They try to get information from the context of your speech, to help work out the correct interpretation.

The goal of this project is to define a set of evaluation criteria and test methods for the interactive voice recognition systems for playing a Tic-Tac-Toe game.

## PROPOSED METHODOLOGY

Basic requirements to develop this project are as follows:

* Windows OS
* Microsoft Visual Studio 2010
* C++ 11 integrated with VS 2010
* Recording Module

With the availability of above software, we further procced in modelling the logic. The prerequisites of this project are

* Basic i/o operations on file
* Pre-processing of speech data
* Feature extraction
* Modelling of extracted feature
* Enhancing model

Above discussed topics are broadly elaborated in experimental setup section.

With the availability of above tools, we further proceeded. Below is the steps for our project.

Application Flow

1. The game will start only after either one of the users utters a specific word  
'​BEGIN​'.  
2. Each utterance of the user will be considered only after a '​Start Recording​'  
is displayed to the user, which will work as a marker for start of recognition  
and ‘​Stop Recording’ ​which will be the end of speech recognition. This  
duration is kept 3 seconds.  
3. The game begins by showing a grid of 3 X 3 filled by ‘-’ (Unmarked). As the  
game progresses, users speak the positions one by one. Condition of  
winning is checked after each move.  
Numbers spoken from 0 to 8 determine the position on the grid.  
4. Both the players speak and hence play alternately, as in the game, with '​X​'  
and '​O​' symbols assigned to them accordingly.  
5. A player's turn consists of uttering a single grid-position (from ​0-8) to be  
marked, followed by a confirmation from the application side, to which  
he/she replies either '​yes​' or '​no​'.  
6. If 'Yes' is detected and the spoken digit position is valid, then the game  
smoothly proceeds ahead.  
7. While, if 'no' is detected or the spoken position is invalid, in the sense that  
that position was already covered or out of the scope of the game, then the  
given turn is repeated for that player until he utters a valid 'yes' in future.  
8. After each successful turn, the grid is updated on the console screen  
(corresponding to the uttered position) according to the mark represented  
by the player of that turn. (either X or O)

Models

Since this is a 2 player game, we developed 2 pre-trained voice models for each  
word listed below. Turn alternates between 2 players and an appropriate model is  
applied for recognition.  
1. Begin  
2. Digits : 0-9  
3. Yes  
4. No

Trainable Module

On the start of the application, the user can select an option to build his/her own  
model.  
On selecting this, the user will be prompted to speak 10 iterations of one of the  
words listed above after further selecting it from the menu shown on the console.  
This goes through the model building pipeline.  
This involves: Preprocessing and Clipping, Cepstral coefficients, Observation  
Sequence and finally HMM modelling.  
This new set of 10 observations are trained using one of the previous pre-trained  
models for that word so that it just fine-tunes that word according to the user and  
makes it better for correct recognition.  
The user can then start the game and select the ​Custom User​ option, when the  
user selection screen is displayed, to use this custom built model.

## EXPERIMENTAL SETUP

This project is divided into following modules:

1. Training Module
2. Testing Module
3. Live Training Module
4. Training Module

The flow for training over data is as follows:

* 1. Record the data as 20 utterances of each digit and word
  2. Extract frames for every utterance
  3. Using local distance analysis (in vector quantization) calculate the observation sequence.
  4. Pass this observation sequence to HMM for model designing.
  5. Now enhance the model using HMM re-estimation algorithm.

Now reference model is ready for our project. The training of data is not integrated with GUI application. This is different module which will just evaluate reference model.

1. Testing Module

System will give instruction what is going on and user is required to follow it. The flow of testing is as follows:

* 1. Live recording of data is done when system instruct.
  2. Testing the data with pretrained models.
  3. Verifying the digit or word detected with user.
  4. If verification is successful display the digit or word.
  5. If verification fails, record the input again.

## RESULT

We are getting spoken digit or word which are already stored in a data file for playing a Tic-Tac-Toe game.