INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 2)

Available online at: www.ijariit.com

Journal of Speech to Text Conversion

Dhanush Kumar S
kumardhanush13@gmail.com
Easwari Engineering College, Chennai, Tamil Nadu

Lavanya S
lavanyasivasankaran96@gmail.com
Easwari Engineering College, Chennai, Tamil Nadu

Madhumita G
madygv14@gmail.com
Easwari Engineering College, Chennai, Tamil Nadu

Mercy Rajaselvi V
mercyeec@gmail.com
Easwari Engineering College, Chennai, Tamil Nadu

ABSTRACT

Education is the basic asset of every individual in the society. Unlike others, the visually impaired student community is forced to be dependent on this aspect. As it is rightly said that '' Nine tenth of Education lies in Encouragement '', we propose SCRIBE BOT, a chat bot that helps this community ace their examinations. Scribe bot is an intelligent bot that is trained to read out questions and acquire the corresponding answers. The proposed system is a collaboration of artificial intelligence and machine learning coupled with voice recognition. We have used an algorithm which is a DNN-HMM hybrid. The necessary functionality like authentication which is implemented using rule-based approach is also included. The speech-to-text conversion and contextual pattern recognition are implemented as well. Thus our system overcomes the need for volunteers, bridging the gap between volunteer requirement and the visually impaired students. The proposed system can not only be beneficial for the visually impaired but also for the physically disabled community.

Keywords: *Artificial Intelligence, Machine learning, Voice recognition, DNN (Deep Neural Network), HMM (Hidden Markov Model), Rule-based, Speech-to-text conversion, Contextual pattern recognition.*

1. INTRODUCTION

In order to help the visually impaired student community ace their examinations, we have proposed the system ScribeBot which is a chat bot. The system is trained with deep neural networks to obtain high accuracy speech to text conversion for serving the examination purpose. It is implemented using Raspberry Pi 3. The technology we have used to implement this system is explained below.

2. DEEP NEURAL NETWORKS

Deep-learning networks are different from the other single-hidden-layer neural networks by their depth**;** that is, the number of hidden layers through which data flows for pattern recognition.

Initially, neural networks like the perceptrons were shallow, composed of one input and one output layer, and at most one hidden layer in between. “Deep Learning” is often referred to neural networks with at least three hidden layers.

In these networks learning of each node is achieved by acquiring knowledge from the output of the previous node which is fed as input to the current node. This is a form of neighborhood learning. The more deep the layer is, the more advanced features it can recognize as it combines features from the previous layers together for the desired feature recognition.

Deep-learning networks perform automatic feature extraction, which requires no human interaction. This is hence more useful than the most traditional machine-learning algorithms that are present. When training on unlabelled data, each node layer in the network learns features automatically by repeatedly trying to reconstruct the input from which it draws its samples. In the process of minimizing the difference between the network’s guesses and the probability distribution of the input data itself, the network completes its learning process.

The networks learn to recognize the relationship between relevant features and desired best results. Thus they try interpreting what exactly is given in the input and helps achieve the target.

Having trained with the labeled data, the network is then exposed to unlabelled data so as make the system intelligent. The more the data that is fed the system, the better its accuracy is. Deep Neural Networks are thus superior to all the other systems as they can deal with large quantities of unlabelled data.

These networks have an activation function that converts the given input to the desired output. Most popular activation functions include logistic, or softmax, a classifier that assigns the probability to a particular output. This is called Predictive Modelling.

3. LONG SHORT TERM MEMORY

Although standard RNNs are powerful in theory, they can be very difficult to train. There are many techniques such as Hessian-free optimization that can be applied to the networks to improve its training ability. In order to facilitate this, we can modify the architecture of the network.

One of the reasons training networks is difficult is that the errors computed in back propagation are multiplied by each other once per time step. If the errors are small, the error quickly dies out, becoming very small; if the errors are large, they quickly become very large due to repeated multiplication.

4. LITERATURE SURVEY

Hendrik Meutzner et.al.[1] have proposed using HTK for improved audio CAPTCHAs based on audio perception and language processing. We have only taken the automatic speech recognition part from this journal which is used to get past the CAPTCHAs on the websites. The authors have trained acoustic models to teach the system to understand the speech patterns. They used Hidden Markov model toolkit (HTK) to train the acoustic models. For each state in the Hidden Markov model, the output probability looks like

$b(om) = P(om|qm = i)\ i = 1, 2,..., Q$,

where Q is the maximum number of states in the model and om represents the feature vector, referred to as the observation hereinafter. For training the speech recognizer, the goal is to estimate the set of model parameters λ. This can be achieved by optimizing a maximum likelihood criterion

$\lambda ML = \arg\max \lambda \{P(o1 ... oT |\lambda)\}$

After training, the individual segment models are combined into a larger compound HMM to represent a user-defined grammar. Then, the sequence of observed features o1 ... oT can be decoded by searching for the optimal state sequence $q* 1 ... q* T$ through the compound HMM

$[q1 ... qT]* = \arg\max q\ 1...q\ T \{P(q\ 1 ... q\ T |o1 ... oT ,\lambda)\}$,

Results Achieved:

		Word Accuracy [%]				Sentence Accuracy [%]			
		$\mathcal{D}=0$		$\mathcal{D}=1$		$\mathcal{D}=0$		$\mathcal{D}=1$	
Features	N_{Init} [%]	Direct	Rescore	Direct	Rescore	Direct	Rescore	Direct	Rescore
MFCC	25	74.37	75.32	77.97	78.85	11.11	14.14	19.19	21.72
	50	73.48	74.56	77.40	78.35	10.10	11.62	18.18	23.23
	75	74.94	75.51	78.60	79.04	14.14	16.67	22.22	23.23
	100	73.93	74.94	77.53	78.54	12.12	14.14	21.21	21.72
PLP	25	78.03	**79.55**	81.31	**82.64**	16.67	19.19	26.77	**29.80**
	50	**78.98**	79.29	**82.13**	82.45	**19.19**	19.19	**28.28**	29.29
	75	78.35	78.91	81.82	82.26	13.64	17.68	23.74	27.78
	100	77.97	78.79	81.31	81.94	18.18	**21.72**	26.26	**29.80**

Figure 2: Learning Results of Neural Networks

Himangshu Sarma et.al.[2] have proposed a speech recognition systems for Assamese Language using HTK. The authors of this paper were building the entire corpus data by themselves so that the model they were trying to achieve would come out perfect. The speech database along with the transcripted files were fed into the system to recognize the speech patterns. The system was designed to impose the speech rules on the transcriptions and also on the incoming database. The recognizer was trained with all these speech

data and the transcripted files to help it understand the assamese syllables well.The achieved accuracy was very low for most of the alphabets and only a few letters had high accuracy. The lowest accuracy that was recorded being 0% for the pronunciation 'ou' and the highest accuracy that was recorded was 91.7% for the letter 'n'.

Musab T. S. Al-Kaltakchi[4] uses several techniques which involves fusion and techniques which doesn't involve fusion. They used power normalized cepstral coefficients (PNCCs) and Mel Frequency Cepstral Coefficients. The acoustic model for the speaker identification was created using Gaussian mixture model-universal background model. The training dataset that they've used are as follows.

- TIMIT database.
- SITW database.
- NIST 2008 database.

They had considerable accuracy for each training datasets. The following accuracy is for 30db voice notes.

- TIMIT database – 90.83%
- SITW database – 80.83%
- NIST database – 92.5%

We can see that the NIST database has the highest accuracy among the three which might be because of the reason that the database had covered all the word phrases and sentiments in each decibels. The above accuracies were obtained even under particular noisy conditions.

Tan Lee et.al.[3] have proposed using tone information in continuous speech recognition. Automatic speech recognition and voice/ unvoiced boundary segmentation methods were used to recognize the cantonese dialects and speech patterns. A different type of hidden markov model was used here and that is context dependent tone modelling.By using Utterance wide normalization with context independent models they achieved an accuracy of 54% and with context dependent models they achieved a 60% accuracy.

Tejas Godambe[5] proposes the speech synthesis using some statistical models. They have used Hidden Markov Model for the speech synthesis. They have found out better results with the hybrid models which had Hidden Markov Model along with the exemplar based voice conversion techniques. They aimed at reducing the over the smoothness of the speech synthesis model.

5. PROPOSED WORK

The proposed system begins by first recognizing the person in front of it. This ensures the integrity and prevents identity theft. Further, it verifies the date and the subject of the examination by voicing out the same. As exam time commences it begins reading the questions. On reading each question it asks the student if he/she wants to skip the question and move to the next one or answer the question. If chosen to answer the bot converts the answer told to text and stores in the SD card as well as displaying it on the monitor. This bot is facilitated with features to review the questions when answered, is trained to identify the correct words to be accounted even if they are homophones and to provide timer assistance to the students.

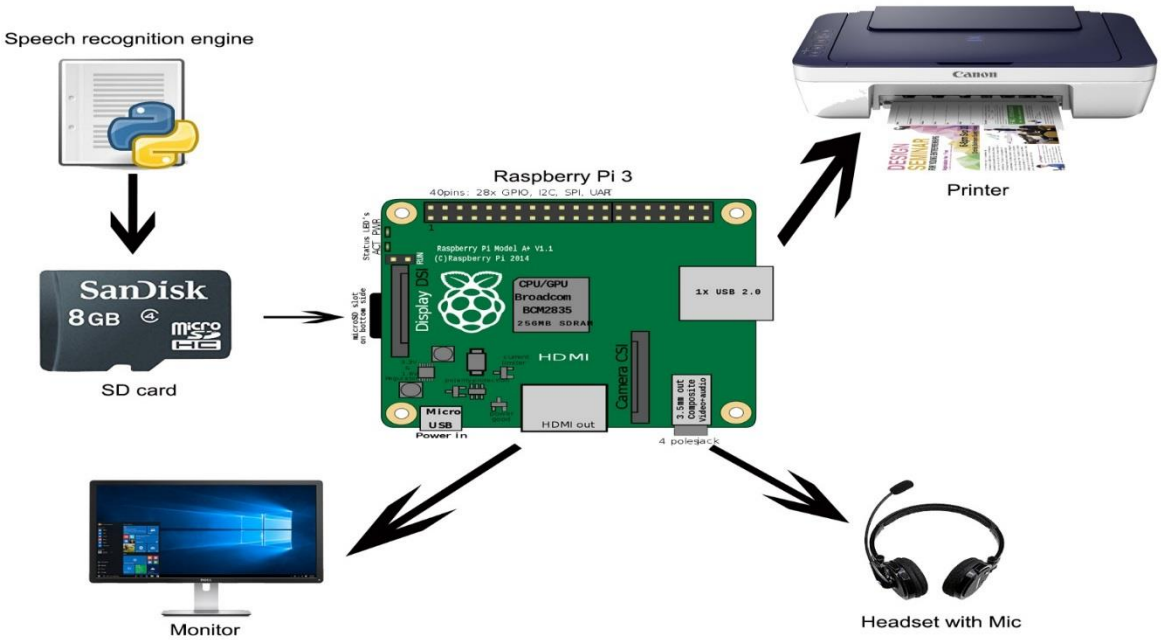

Figure 3: Architecture of ScribeBot

SCRIBEBOT is a Raspberry Pi 3 integrated with a headset with Microphone, a Monitor and a printer. The voice signals are stored as .wav files. The .wav format is then converted to text using Google speech. Then contextual pattern recognition is carried out by DNN-HMM hybrid algorithm. Finally, the answers are stored in excel sheets within the SD card. The exact implementation of the system is described in figure 4 given below.

6. IMPLEMENTATION OF SCRIBEBOT

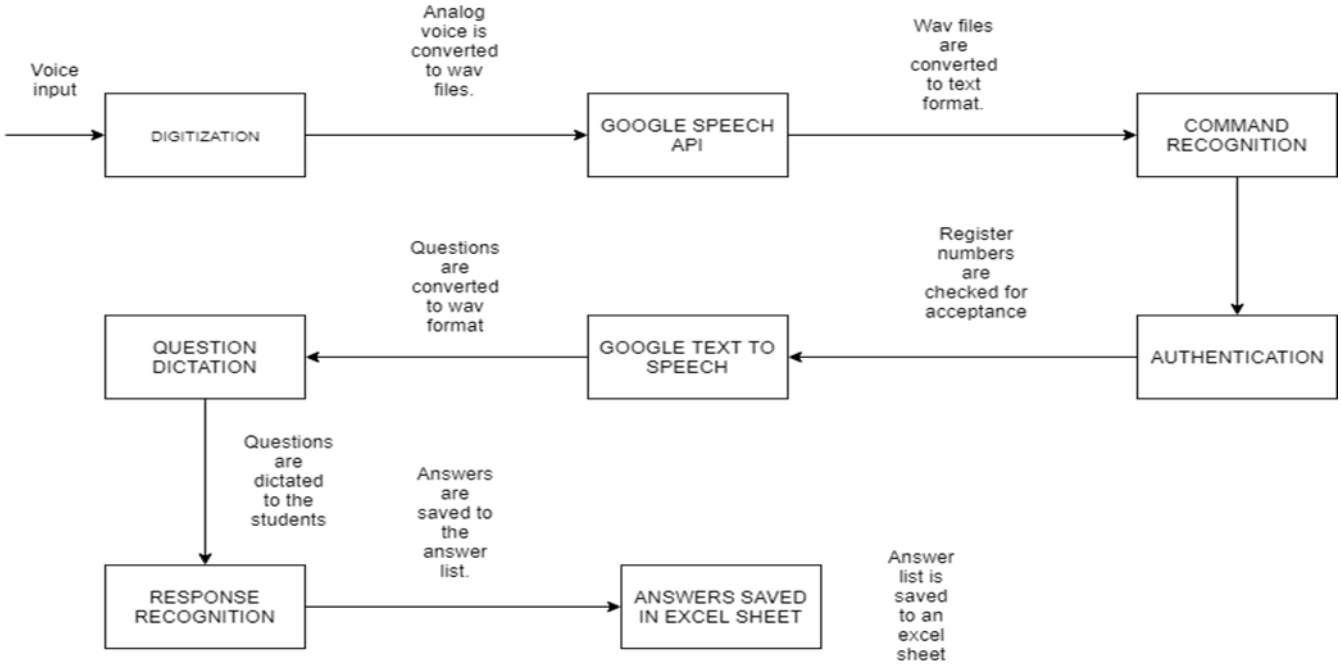

Figure 4: Implementation of Scribebot

The system we've created uses the advanced deep learning algorithms and neural networks which have been implemented in the Google speech API. The Google Speech API is a cloud platform which allows the developers to convert the speech data into text format for processing. We've created a chat bot interface for the visually impaired students which is capable of recognizing the commands given by the students and it also reads the questions from a excel file. The answers which are spoken by the students are again converted to text and stored in a text file for the teachers for correction. The two main processes involved in our work is the speech to text conversion and also the process of converting the text to speech so that the visually impaired students can know about the question that has been asked in the question sheet. The system has been implemented using python programming language which imports the Google Speech API package along with the Google Text To Speech package. The python script run on a Raspberry Pi 3 powered by the OS "Raspbian".

The system starts with some welcome phrases and asks the students their roll which is a unique number. The excel sheet with the student names and their roll numbers will be copied to the SD card which will be accessed by the python script. If the roll number matches the one in the excel sheet then the scribe bot proceeds further to the question dictating and answer recording process. During this stage, the first question from the excel sheet will be dictated and the system asks if it needs to be dictated again. The student can answer the question or they can say "Yes", if the student says "yes", the question will be dictated again or else the answer will be converted to text and saved in a list. The same process continues until all the questions are dictated and answers recorded for the corresponding question. At the end of this process, the student is asked whether the answers have to repeat again so that they can know what was recorded for each question. If the student says "yes", the questions and answers are dictated from the list one by one. Else the system exits by saying thank you and saves the answer list to the excel file.

We've created the scribe bot for working around with only some simple general knowledge questions. The system requires high-speed internet connection of at least 100mbps so that the responses are quick. A slow internet connection will result in the system to face the timeout consequences of the API. In cases where the system couldn't hear any voice, the student will be asked to say the answer again until the system understands the word completely for speech to text conversion.

7. EXPERIMENTAL RESULTS AND DISCUSSION

EXPERIMENTAL SETUP:

Raspberry Pi 3 model B is a single computer with wireless LAN and Bluetooth connectivity. It has a 1gb RAM, BCM43438 wireless LAN and Bluetooth low energy on board, 40-pin extended GPIO, 4 USB 2 ports, 4 pole stereo output and composite video port and

a full-size HDMI port.The SD card contains the OS and the resources, for example, the questions excel sheet. The Scribe bot which is implemented using python is put on the SD card which will start on the system start up.The monitor is not a part of our proposed system but it can be connected to view the answer files as they are created.The answer sheets can be printed out through a printer if the hard copy of the answers is needed.The headset is used to reduce the noise level in the surroundings and to make sure the answers and responses are clear to the students and to the scribe bot.

We have used the famous WER algorithm for calculating the error rate using Levenshtein distance.The Word Error Rate (short: WER) is a way to measure the performance of an ASR. It compares a reference to a hypothesis and is defined like this:

$$WER=S+D+INWER=S+D+IN$$

where

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions and
- N is the number of words in the reference

The WER is just the Levenshtein distance for words.

$m=|r|n=|h|m=|r|n=|h|$

$D0,0=0Di,0=i,1\leq i\leq mD0,j=j,1\leq j\leq nD0,0=0Di,0=i,1\leq i\leq mD0,j=j,1\leq j\leq n$

For $1\leq i\leq m,1\leq j\leq nDi,j=min$
$Di-1,j-1+0$ if $ui=vjDi-1,j-1+1$ (Replacement)$Di,j-1+1$ (Insertion)$Di-1,j+1$ (Deletion)For $1\leq i\leq m,1\leq j\leq nDi,j=min\{Di-1,j-1+0$ if $ui=vjDi-1,j-1+1$ (Replacement)$Di,j-1+1$ (Insertion)$Di-1,j+1$ (Deletion)

Pseudocode:

1. Set n to be the length of s.
    a. Set m to be the length of t.
    b. If n = 0, return m and exit.
    c. If m = 0, return n and exit.
    d. Construct a matrix containing 0..m rows and 0..n columns.
2. Initialize the first row to 0..n.
    a. Initialize the first column to 0..m.
3. Examine each character of s (i from 1 to n).
4. Examine each character of t (j from 1 to m).

5. If s[i] equals t[j], the cost is 0.
   If s[i] doesn't equal t[j], the cost is 1.

6. Set cell d[i,j] of the matrix equal to the minimum of:
   a. The cell immediately above plus 1: d[i-1,j] + 1.
   b. The cell immediately to the left plus 1: d[i,j-1] + 1.
   c. The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost.

7. After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

Reference	**Hypothesis**	**Error rate**
Two	**To**	**1**
Three	**Tree**	**1**
Three	**Free**	**2**
Four	**For**	**1**
Hi	**High**	**2**
Twenty	**Wendy**	**2**
AVERAGE ERROR RATE:1.5		

8. CONCLUSION AND FUTURE WORK

The aim of our approach is to implement the speech recognition using Google API. From the literature survey, we've come to a conclusion that Hidden Markov Models and Deep Neural Networks are the best bet to implement speech recognition and the toolkit we've chosen have been designed with the same hidden markov models and deep neural networks concept. The acoustic models is trained to teach the system to understand the speech patterns [1].The entire corpus data have is built entirely in order to make the speech-to-text conversion system to be perfect [2]. Feature extraction within speaker identification should be less influenced by noise or the person's health [3]. This is an important aspect which is being considered while building the system. Like in [4], a different type of hidden markov model was used here that is context dependent. The acoustic model for Indian accent is already available for use. The audio may contain bad acoustic (poorly articulated, dis-fluent, unintelligible, inaudible, clipped, noisy) regions as the audio is not particularly recorded for building TTS systems [5]. We need to create the speech recognition module and test it with the syllabus of blind student's examination and see how the whole exam process happens. We may also need to change the configurations to enable the engine to listen to the students for a long time because there are also some long answers in the question paper. Since we're running the engine in Python, it is easy to read the questions file and write the answers to another file without any malfunctions. The text file is stored in SD card. The toolkit has a live Speech Recognizer to start the recognition and the time limit can be set between each intervals the engine should start the recording and the toolkit has classes to convert the recorded .wav files to text format for printing it in the console, The converted string format can be compared with several cases and the user can be asked to repeat the command again. Once the voice is processed properly, the modules for conversion of the entire answer to a text file can be called subsequently one after another. We will have to test the engine by speaking to it with long answers and see if it's converting the voice exactly as spoken by the user.

9. REFERENCES

[1] Hendrik Meutzner, Santhosh Gupta, Viet-Hung Nguyen, Thorsten Holz and Dorothea Kolossa, " Towards improved audio CAPTCHAs based on audio perception and language processing " Association of Computing Machinery Transactions on Privacy and Security. Volume 19, No. 4, Article 10, NOVEMBER 2016.
[2] Himangshu Sarma, Navanath Saharia and Utpal Sarma, " Development and analysis of speech recognition systems for Assamese Language using HTK " Association of Computing Machinery Transactions on Asian low Resource Language Information Processing. Volume 17 No. 1, Article 7, OCTOBER 2017.
[3] Musab T. S. Al-Kaltakchi, Wai L. Woo, Satnam Dlay and Jonathon A. Chambers, " Evaluation of a speaker identification system with and without fusion using three databases in the presence of noise and handset effects ", EURASIP Journal on Advances in Signal Processing.
[4] Tan Lee, Wai Lau, Y. W. Wong and P. C. Ching. "Using tone information in cantonese continuous speech recognition ",Association of Computing Machinery Transactions on Asian low Resource Language Information Processing. Volume 1. No. 1 MARCH 2002.
[5] Tejas Godambe, Sai Krishna Rallabandi, Suryakanth V. Gangashetty, Ashraf Alkhairy and Afshan Jafri. "Developing a unit selection voice given audio without corresponding text ", EURASIP Journal on Audio, Speech and Music Processing.

