An Approach to Accept Voice in Code Editor through Speech Recognition

G. Parthasarathy [1],A R. Rangeesh [2],V S. Sri Kishowr [3], R. Sriram [4], S. Vijay[5]
TRP Engineering College, Tiruchirapalli, India.

***Abstract*-- Since the emergence of technology, there has been a gap between Human Impairment and technology. We believe that our proposed model will fulfill this gap. It will be more useful for the physically impaired people.it also helps them to code just by using their voice. We are developing an IDE for Python language, where the user can able to code using their voice. Here we have an interactive IDE which helps others to interact with the code editor. This model will reduce the amount of errors occurred while typing and also helps the users to multitask. Like they can code while they are busy doing some other work.**

Keywords: Speech Recognition, Speech-Text, Collective Intelligence, IDE, python.

I INTRODUCTION

We are living in an era where we can do everything using technology. From having an own personal assistant to sending cars to the space. But even in this time there is less number of services available to the physically challenged people. Our intention is to create a IDE for python which enables user to code by using their voice.

Speech Recognition is one of the most incredible technology and it is used to execute commands in computer via voice. The use of automatic Speech Recognition approach has been increasing progressively, from a simple instrument that responds to a small set of sounds to a complicated system that responds to a spoken natural language.

Some applications of speech recognition systems are playing back simple information, Call Steering, Speech-to-text processing, voice user interface, Verification/Identification.

Speech Recognition is the Transformation of Verbal inputs known as words, phrases, or sentences into text content.

We are using the speech recognition systems in our day to day life. For example we have Google Assistant in every android phones which can be accessed by simply saying “Ok Google”. It serves the same purpose like Google search engine, but we can access all the things simply through our voice. Same technology is used in Apple mobiles with different name “Siri”. We have many Text Editors which uses the concept of speech-to-text. But there is no code editor with that possibility. We are proposing a model which takes voice as the input and writes its equivalent Python code in the code editor. This method will enrich the programming ability in every technocrat and helps them to understand what they are doing in it.

The speech recognition process is performed by a software component known as Speech recognition engine. The initial function of the speech recognition engine is to process spoken user input and translate it into text that an application can understand. Speech recognition engine requires two kinds of files to recognize speeches, which are described below.

A. Language Model or Grammar

A Language Model is a file containing the probabilities of sequence of words[4]. A Grammar is a much smaller file containing set of predefined combination of words. Language Models are used for ‘Dictation’ applications, whereas Grammar are used as desktop ‘Command and Control’ applications.

B. Acoustic Model

Contains a statistical representation of the distinct sounds that make up each word in the language Model or Grammar[5][11].Each distinct sound corresponds to a phoneme.

The above works is explained in the following sections. Section 2 describes the related work and the proposed method is explained in Section 3. The proposed algorithm and model is explained in Section 4. Section 5 gives the experimentation and result. Conclusion will be given in Section 6.

II RELATED WORK

The related work for our proposed method is given below: Scientist found it difficult to make a system respond appropriate, while the given commands operating via voice[1]. The problem with the speech recognition is it is difficult to analyze sound from a simple instrument to the complex sounds. Based on the advancement of technology most applications are developed like playing back simple information, call steering, speech to text processing, voice user interface, verification / identification, etc… Using this concept the author have developed a speech to text code editor for HTML. Hidden markov model is used to develop this model.

But in this model it is difficult to process the language other than English. It doesn’t work well when there is noise in the surrounding. This model still needs more improvement for commercial purpose.

Over past two decades, scientist and engineers are trying to combine then concepts of Hidden Markov Model (HMM) and Neural networks(NN). In speech recognition. This hybrid method takes advantage of both concepts there by increasing the flexibility and performance. The hybrid system proposed by bourlard and morgan applied a neural network to estimate the posterior probabilities of HMM states. Recently developed method called TANDEM recognition approach introduced by Hermansky has shown a large improvement in recognition performance [2]. This method produces a high accuracy of 74.9% accuracy in recognizing the spoken words. This improves the performance of continuous speech recognition using HMM.

By the above literature, we motivated and developed a speech to text code editor for Python. Next section explains about proposed model of our work.

III PROPOSED MODEL

Python is an emerging language in recent times, so we decided to develop an IDE for python which accepts input through voice and convert's it into an equivalent python code.

Fig.1 consists of instructions in two modes. Command mode and Coding mode

A. Command Mode

In this mode the obtained text is matched with the system commands such as run, save, open, etc..

B. Coding Mode

In this mode the obtained text is matched with the python syntax templates.

Using these modes we can able to interact with the IDE and run the porgrams.this method is even more easy to use than the previously existing IDE.

When the user gives voice input that is collected through mic, Noise cancellation is performed on the obtained voice note and then it is processed. It is checked to choose the modes. If it is a coding mode then the voice is processed and converted into equivalent python code. If it a command mode then terminal commands like save, run, download are executed.

In Fig 2 Houndify SDK is used to match the voice input with its equivalent python syntax. It uses collective intelligence concept to match the most relevant phrase.

The output is printed in the Trinket embedded python code editor on the result page.

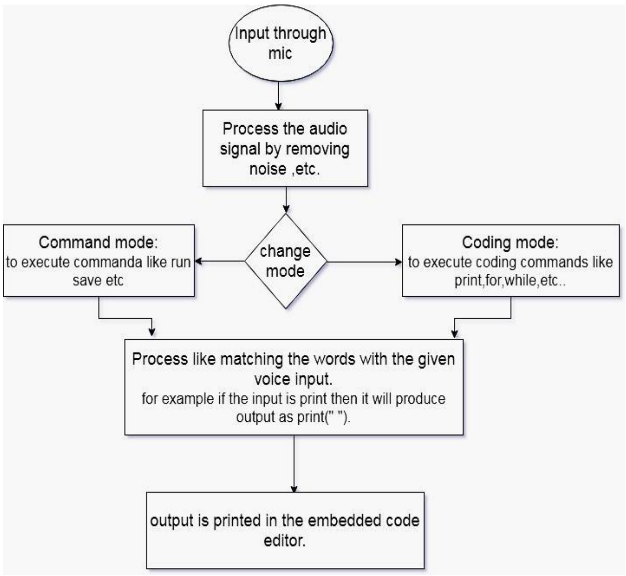

Input through mic
Process the audio signal by removing noise ,etc.
Command mode: to execute commanda like run save etc
change mode
Coding mode: to execute coding commands like print,for,while,etc..
Process like matching the words with the given voice input. for example if the input is print then it will produce output as print(" ").
output is printed in the embedded code editor.

Fig.1 Flowchart of our proposed model

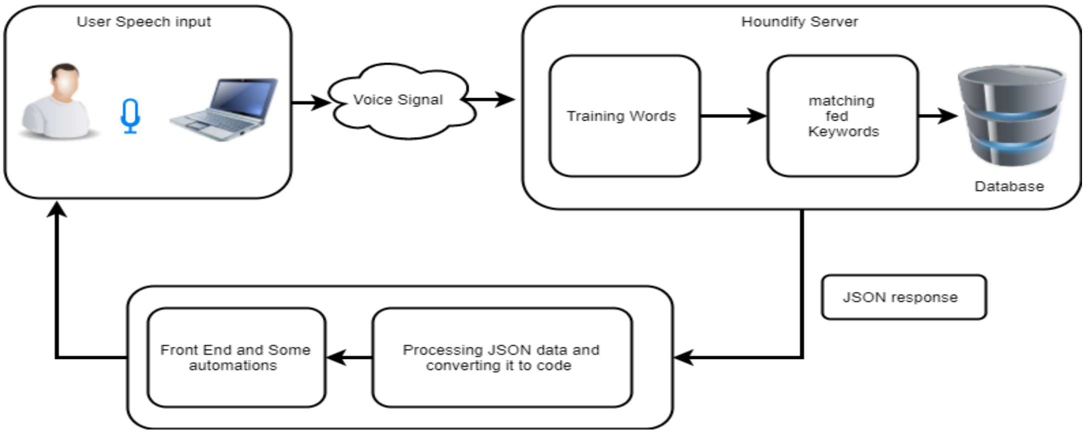


Fig.2 Architecture diagram of the proposed model

IV ALGORITHMS AND MODELS

A. Hidden Markov Model

It is modern general purpose algorithm[3]. It is widely used in speech recognition systems because of that statistical models are used by this algorithm, which creates output in the form of series of quantities or symbols. It is based on statistical models that output a series of symbols or quantities

B. Dynamic Time wrapping

The Dynamic Time Warping (DTW) is an algorithm, it was introduced in 1960s . It is an essential and ages algorithm was used in speech recognition System known as Dynamic Time Warping algorithm, it is used to measure the resemblances of objects/ sequences in the form of speed or time[6][9]. For instance similarity would be detected in running pattern where in film one person was running slowly and other person was running fast. This algorithm can be applied to any data; even data is graphics, video or audio. It analyzes data by turning into a linear representation. This algorithm is used in many areas: Computer Animation, Computer vision, data mining , online signature matching, signal processing , gesture recognition and speech recognition.

C. Neural Networks

Neural Networks were created in the late 1980s. These were emerging and an attractive acoustic modeling approaches used in Automatic Speech Recognition (ASR). From the era the algorithms have been used in different speech based systems such as phoneme categorization[10][11]. These algorithms are attractive recognition models for speech recognition because they formulate no assumptions as compares to Hidden Markov Models regarding feature statistical properties. This algorithm is used as preprocessing i.e.; dimensionality reduction and feature transformation for Hidden Markov Model based recognition.

D. Collective Intelligence

Collective intelligence (CI) is shared or group intelligence that emerges from the collaboration, collective efforts, and competition of many individuals and appears in consensus decision making. The term appears in sociobiology, political science and in context of mass peer review and crowdsourcing applications. Collective intelligence strongly contributes to the shift of knowledge and power from the individual to the collective.

Houndify provides SDK for Speech Recognition purpose.it uses Collective Intelligence technique. This concept is used in our proposed model.

V IMPLEMENTATION AND RESULT

The below shown snapshots uses houndify SDK which matches the spoken word with most accurate word and responses with json format which is then processed and converted into suitable python code and displayed in trinket IDE.

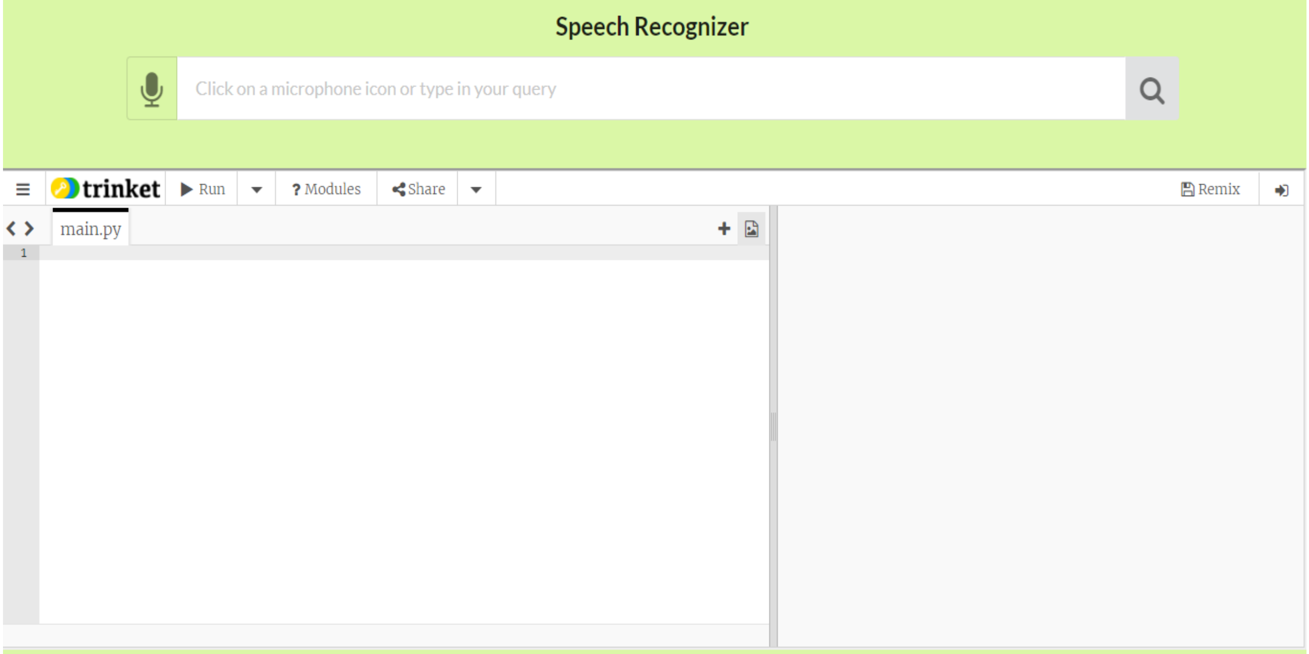
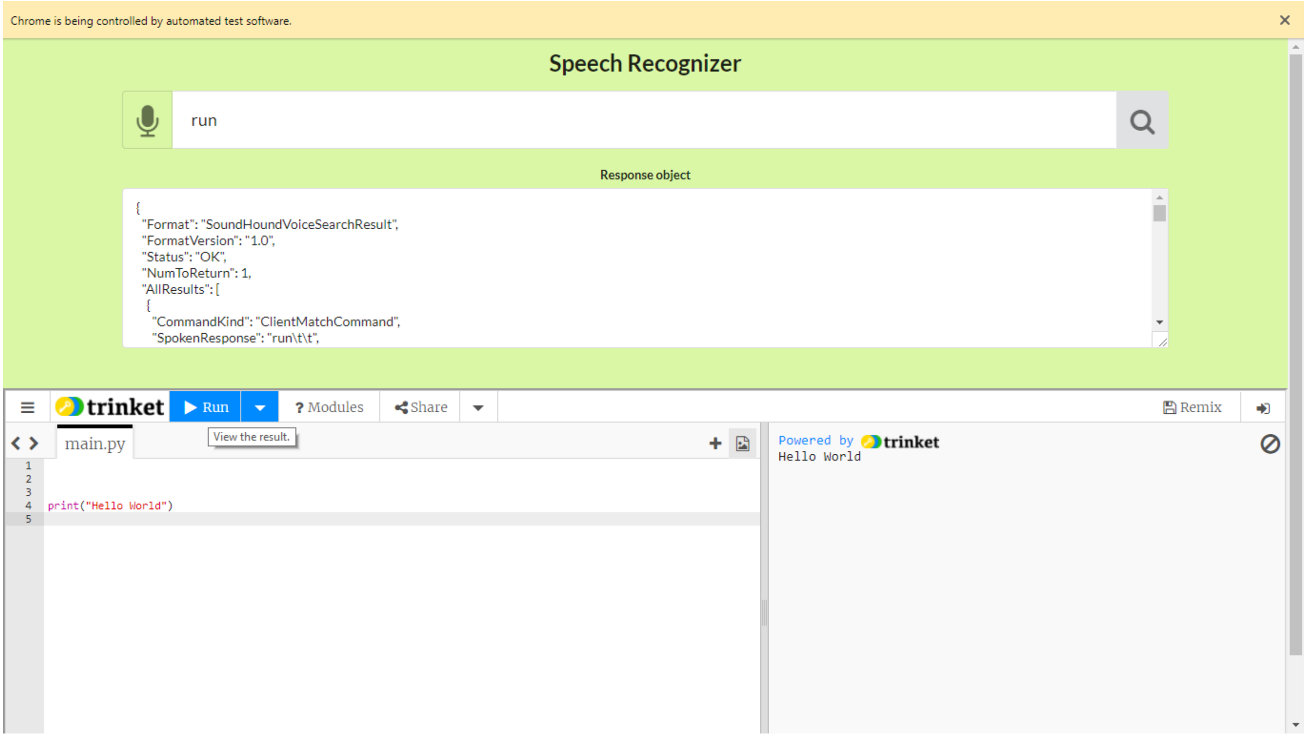


Fig.5.1. Screenshot Of UI/UX



Chrome is being controlled by automated test software.

Speech Recognizer

run

Response object

```
{
 "Format": "SoundHoundVoiceSearchResult",
 "FormatVersion": "1.0",
 "Status": "OK",
 "NumToReturn": 1,
 "AllResults": [
  {
   "CommandKind": "ClientMatchCommand",
   "SpokenResponse": "run\t\t",
```

```
print("Hello World")
```

```
Hello World
```

Fig.5.2 screenshot of Response

the Fig.5.2 shows the implemented model. It uses pyautogui to control screen automations. The response which is obtained in JSON format is then processed using selenium to get the spoken response of the user which is later converted into equivalent python syntax.

TABLE I. KEYWORDS

S.no	Input (Obtained voice)	Output	Success /Failure
1.	Print	Print(" ")	Success
2.	While	While( ):	Success
3.	Print Hello World	Print("Hello world")	Success
4.	Run	Run's the code	Success
5.	Terminate	" "	Failed
6.	New Function	def fun_name():	Success
7.	If	If():	Success
8.	Print sriram	" "	Failed
9.	Variable power	" "	Failed
10.	Variable index	Var index	Success

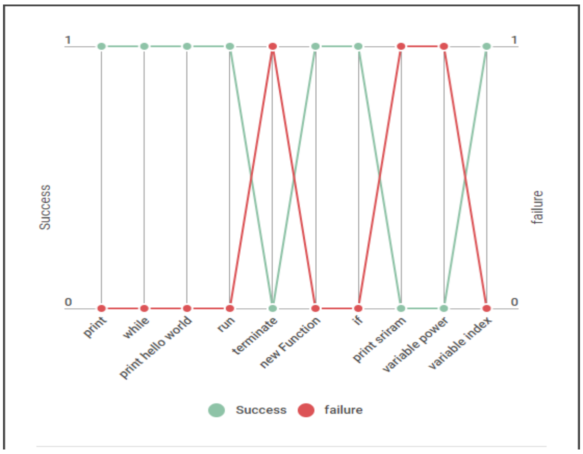


Fig.5.4. Success-Failure Rate

The fig.5.3 shows the keywords we used in this model. The input and its equivalent output is given with the success/failure status. The fig.5.4 shows the result of this model according to the success/failure status. Green line indicates the success and red line indicates failure.

VI CONCLUSION AND FUTURE WORK

Our proposed model is deployed in the python environment. The results show that the model is 70% accurate in matching the syntax with spoken word. This accuracy is in acceptable range. In our future work we plan to improve this accuracy by proper implementation. From this model it is clear that everyone can code without any trouble using this model. We are also planning to release this model for other languages like C,C++ , etc.. in the near future.

REFERENCE

[1] Farhan Ali Surahio, Awais Khan Jumani, Sawan Talpur(2016) "An approach to accept input in Text Editor through voice and its Analysis, designing, development, implementation using speech recognition" IJCSNS, vol.16 No.3

[2] Hongbing Hu, Stephen A.Zahorian (2010) "Dimensionally Reduction Method for HMM Phonetic Recognition" ICASSP Vol 1.

[3] R. Bellman and R. Kalaba, \On adaptive control processes,"Automatic Control, IRE Transactions on, vol. 4,no. 2, pp. 1{9,1959.

[4] S. A. Zahorian, A. M. Zimmer, and F. Meng, (2002)"Vowel Classification for Computer based Visual Feedback for Speech Training for the Hearing Impaired," in ICSLP2002.

[5] Sakoe, H. and Chiba, S. (1978). Dynamic Programming Algorithm Optimization for Spoken Word Recognition.IEEE Trans. on Acoustics, Speech, and Signal Processing,26(1):43-49, February 1978. Reprinted in Waibel and Lee(1990).

[6] V. Niennattrakul and C. A. Ratanamahatana,"On clusteringmultimedia time series data using k-means and dynamictime warping," in Multimedia and Ubiquitous Engineering,2007. MUE '07. International Conference on, 2007, pp.733{738.

[7] Vintsyuk, T. (1971). Element-Wise Recognition of Continuous Speech Composed of Words from a Specified Dictionary. Kibernetika 7:133-143, March-April 1971.

[8] Nadeem.A.K, Habibullah.U.A, A.Ghafoor.M, Mujeeb-U-Rehman.M and Kamran.T.P. "Speech Recognition in Context of predefined words, Phrases and Sentences stored in database and its analysis, designing, development and implementation in an Application". International Journal of Advance in Computer Science and Technology, vol.2, No12, pp .256-266, December 2013.

[9] Vit Niennattrakul, Chotirat Ann Ratanamahatana "on clustering Multimedia Time Series Data Using K-Means and Dynamic Time Warping."(2007) International conference on Multimedia and Ubiquitous Engineering, 0-7695-2777-9/07.

[10] A.waibel, T.Hanazawa, G.Hinton, K.Shikano, and K.J.Lang, "phoneme recognition using time-delay neural networks."(1989) IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37, pp.328-339.

[11] J.T.Geiger, J.F.Gemmeke, B.Schuller, and G.Rigoll, "Investigating NMF Speech Enhancement for neural networks bsed acoustic models," in proceedings Interspeech. ISCA, 2014, pp. 2405-2409.