Voice Recognition Scheme For Coding Console Using Natural Language Processing

Prof. Mohan C.Nikam[1], Pooja Dighe[2] , Ruta Patel[3] , Lalita Wagh[4], Varsha Yadav[5]
[1, 2, 3, 4, 5] Dept of Information Technology
[1, 2, 3, 4, 5] Sandip Institute of Technology and Research Centre, Mahiravani, Trimbak road, Nashik 422 213.

Abstract- *Nowadays, voice recognition is used as an alternative for traditional way of typing on a keyboard. Voice or speech recognition is the capability of a system to receive and clarify dictation or to interpret and execute spoken instruction. A voice response system (VRS) is a system interface which acknowledge to speech instructions instead of acknowledge to inputs from a mouse or a keystroke. However on the other hand, Natural Language Processing (NLP) is primarily concerned with language communications between the user and the system. Using NLP computer can detect and understand human speech in real time. NLP has increasing opportunities in research and applications. Adopting voice recognition with NLP, in this project we offer a coding console for software programmer. Where programmer can dictate any code of C programming language and it will be written on editor along with compiler. This project is different approach to the NLP application and replacing the traditional code typing method with smart console recognizing the programmer's speech.*

Keywords- Keyword Extraction, Keyword Mapping, Machine Learning, Voice Recognition

I. INTRODUCTION

Voice recognition willingly referred to as speech recognition. It is a computer hardware appliance or software program with the capacity to decrypt the human speech. Voice detection is generally used to achieve that device execute commands or write beyond having to use a keyboard, mouse or press any buttons. Now a days, this is succeeded on a computer using automatic voice/speech detection (ASR) software programs. Number of ASR systems needs the user to train the ASR program so as to detect their speech so that it enables to convert the speech to text more precisely [13]. For example, you could say open Internet and the computer would open the Internet browser. The first ASR appliance was used in 1952 to detect single digits said by the user (it was not computer guided). Today ASR programs are used in many corporation along with health care, military, telecommunications and personal computing(i.e. hands free estimate).The different types of voice detection systems first is automatic speech detection is individual instance of voice recognition. Following are given some examples of the voice detection systems: 1) Speaker/user dependent system: Any speech recognition system requires some training before it is used which needs you to read a range of words and phrases. 2) Speaker independent system: The speech recognition software detects most user's speech with no training. 3) Distinct speech detection: The user must foible or slipup between each word so that the speech recognition can determine each word separately [14]. 4) Regular speech recognition-The speech recognition can also realize a normal rate of user's speaking. 5) Natural language- The voice detection not only can realize the speech but also return solution to investigation or other queries that are being asked. When speech is converted into text or computer instruction at that time computer need several intricate steps. When user speaks at time wave is created in the environment. Analog to digital converter (ADC) is used to convert the analog signal into the digital data, so that system can easily understand. For doing this it samples/digitizes the speech or voice by taking the exact assessments of the vibrations at constant intervals. The system penetrates the digitized sound to eliminate redundant noise and sometimes to divide it into dissimilar bands of frequency. Wavelength of the sound waves is the frequency, that frequency is noted by user as change in level it also distribute the voice or arrange it to a regular volume level. People don't consistently speak at the same speed so the voice must be arranged to map the speed of the template voice samples [16].

Next the signal splits into small sectors as little as a few hundredths of a seconds or even thousandths in the case of consonant sounds, the consonant stops produced by obstructing airflow in the verbal tract like p or t. The program then matches these sectors to known morphemes in the proper language. A phoneme is the little element of a language, an illustration of the sounds we make and put together to form concise terms. There are around 40 morphemes in the English language (distinct linguists have different opinions on the perfect number) while other languages have more or fewer. Then program check morphemes in the context of the other morphemes around them. It runs the provisional morphemes plot throughout the complicated statistical model and matches them with a large library of known or pre-specified words, phrases and sentences. The program then detects and decides what the user was saying and whether to outputs it as text or issues it as a computer instructions. NLP is an approach to investigate, realize and determine meaning from human language in a good and helpful way. By utilizing NLP, developers can construct and structure knowledge to execute tasks such as automated confession, explanation, named entity, translation, Named entity detection, relationship abstraction, tendency analysis, voice recognition and topic reduction. Apart from common word processor operations that treat text like a mere sequence of symbols, NLP considers the hierarchical structure of language: several words make a phrase, certain slogans make a sentence and finally sentences transmit ideas. NLP is used to analyse the text. It helps computer to understand how humans speak. Hence, Natural Language Processing also enables real world application by using human-computer interactions in the efficient way. Automatic text summarization and parts-of-speech-tagging are most used applications. The applications include name entity recognition, relationship extraction, topic extraction, sentiment analysis and more. In other areas now NLP is commonly used for automated question answering, language translation and text mining [17].

II. LITERATURE SURVEY

The paper titled keyboard less visual programming using voice, handwriting and gesture, presented by Jennifer L. Leopold and Allen L. Ambler, in this paper authors present programming interface which give response to user communication activities. They propose a multimodal user interface design. The majority of these type of systems have concentrated their efforts on speech recognition and gestures (from other sources), and have excluded handwriting. The objective behind developing a multimodal user interface is to improve the capacity to view edit and interact with visually oriented programs using speech and pen instead of the mouse and keyboard [1].

Another paper titled Vocabulary matching for information extraction language, introduced by Peter Bednr, in this paper author extends the previous specifications of the language with the extensions give grant to arrange words in sequence and corresponding metadata annotations in the vocabulary. The paper describes syntax extensions and internal structure of the vocabulary designed for efficient matching and low storage requirements. The presented paper describes an overview the declarative domain-specific language designed for the text processing in various NLP tasks [2].

Mrd. Ing. Alexandra Trifan, Mrd. Ing. Marilena Anghelu, L. Dr.Ing.Rodica Constantinescu, introduced the paper titled Natural Language Processing Model Compiling Natural Language into Byte Code. This paper includes a way concerning the processing of human languages and that language convert into byte code which is easily understand for computer. The paper will treat the requirements needed for this to happen in the programming language known as Java, but the principles should be the same for any or all programming languages. It distributed nature and a different learning approach is expected to make faster progress. The system is faster because it reuses previous data to learn new information. Machine learning is applied not only at retrieval level, but also on generation phase as well. The collaborative nature of the design brings a great improvement to the current NLP modelThe [3].

Peter Bednr introduced another paper titled Unified parsing and information extraction language, wherein they describes the declarative language for specification of parsing and Data abstraction rules that are used in the natural language processing work. The concepts of tokenization is unified using the regular expressions over the input text data abstraction design which is specified using the regular expressions over the annotations and graph mapping designs used for syntactic or co reference connection between words or phrases. The proposed annotation language unifies concept of tokenization (i.e. division of the input text into the sequence of tokens), annotation rules are based on the regular expression over the annotations and graph relations in annotation schema. Proposed language was primary designed for NLP tasks and processing of textual data. However, the language can be used for processing of general sequence or graph structures as well [4].

III. EXISTING SYSTEM

Coder uses text editors, notepad, IDEs for writing the program. On the other hand voice recognition is used in other fields. As voice recognition improves, it is being implemented in more places and it is very likely that we have already used it. Some examples are as follows, where we might encounter speech recognition. 1) Automated Phone Systems: Today many company use mobile phone due to that it help direct to the caller to accurate. 2) Google Voice - It is a service that allows us to search and ask any questions on the computer, tablet, and phones. 3) Siri - Today a good example of speech detection that helps to give answer to question on direct apple devices. 4) Car Bluetooth- Nowadays cars have Bluetooth phone pairing which can also use speech recognition to make instructions such as call my friend to make calls without touching your phone or taking your eyes off the road.

IV. PROPOSED SYSTEM

Proposed System is compatible on Windows platform. System is developed using Microsoft Visual Studio 2010.Voice of user is captured using windows speech recognition. API used for voice recognition is Speech-lib API. The voice of user (coder) is initial input to ADT (Analog to digital converter). Here, Microsoft Speech-lib is used to convert Speech to text. This text is further given to NLP framework as in-put. In NLP framework, two steps are follow in order to generate code: 1) Keyword/Stop-word Removal 2) Keyword Mapping. In stop word removal, text undergoes summarization and unwanted words are excluded. In keyword mapping, extracted words from stop-word removal step are mapped. Finally code is generated from the words listened and detected. SAPI 5.4, provides two types of recognizer i.e. shared process recognizer (SpSharedRecognizer) and in-process recognizer (SpinProcRecognizer). The workflow of the system is as follows:

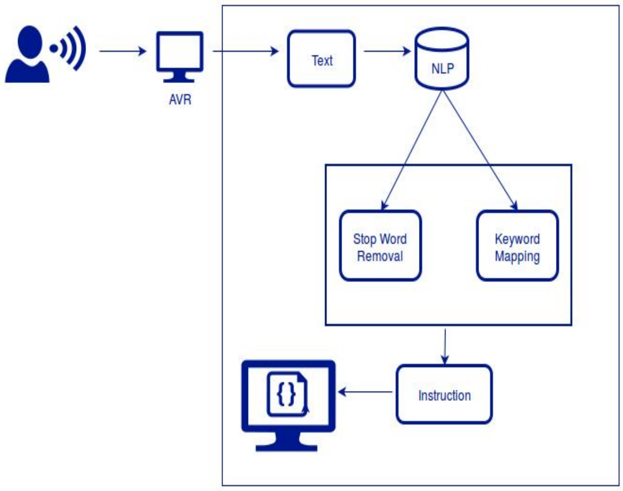


Figure 1. Proposed System Workflow Diagram

The inprocess recognizer is responsible for claiming the resources for applications. E.g. if an inprocess recognizer claims the microphone of the system, other application cannot to use it as it is not allowed. On the other hand, shared recognizer runs in a separate process from application. Hence it can be shared with other applications. In our system we are using shared process recognizer because shared recognizer grant an operation to play perfectly with other voice authorize applications on system. To manage the relationship between the recognition engine object (i.e. the recognizer) and the application, an object is used called as recognition context. When SpSharedRecognizer context is created SAPI create shared detected object. In recognition activity handler the ISPsprecoresult interface is used to retrieve the text which is detected by voice recognition, we used ISPRecoResut.

To retrieve element from text phrase we used ISPPhrase. NLP is main focus of proposed system. System will be time consuming and moderate without NLP. As we are developing a smart console we want our proposed system efficient. Suppose, the coder wants to include any header file, it is obvious that it must start with pre-processor. So, coder will not have to say header file. Even if he\she says, include stdio file or include stdio files, on the console stdio file will be written including symbol of pre-processor and triangular bracket i.e. along with the correct syntax. Here NLP plays important role as proposed system works on naturally spoken human language. Example: Input: take two integer a b, Output: int a, b;

V. MODULES

A module is a separate unit of software or hardware. Our Proposed system divide into three parts are as follows.

1. Graphical User Interface (GUI)
2. Voice to Text Converter
3. Natural Language Processing (NLP)
4. Code Generator

1. Graphical User Interface (GUI):- GUI basically gives the visual presentation of the virtual file system to the users. GUI color patterns design working and nature is quite same to windows. Windows XP style task pane arrange simple approach to normal operations and gives perfect look. Standard Tool bars, Pop-up menus and shortcut keys that make process of software simple for all type of peoples. Simple to Use, Simple accessibility to functions and nice look are the main features of GUI.

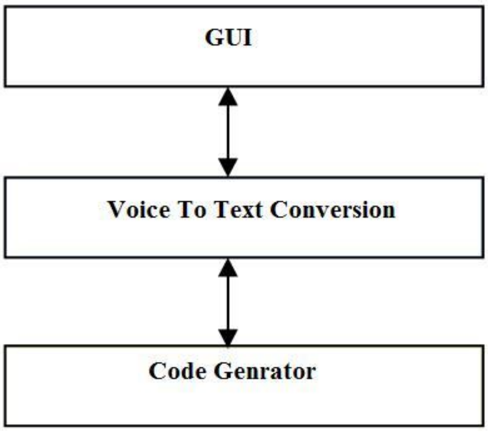


Figure 2. Proposed System Architecture Diagram

2. Voice to Text Converter: - The basis of proposed system is voice to text conversion. In this system, the windows speech recognition WSR is used, which is created on top of SAPI for

grabbing the speech and Microsoft Speechlib API is used to translate recognized speech to text. SAPI version 5.4 is carried with windows 7 and holds two specific types of speech recognition; dictation and command and control. In the proposed system we used Voice recognition technology, in this scheme first system takes the users speech then converts it into text. The speed of CPU and available memory of the system is directly responsible for certainty of dictation. More the resources available, more the context that can be considered in a fair amount of time the more likely the resulting recognition will be correct.

A single recognizer can be used by many circumstances. For example, if a speech enable application has 3 forms then it will have a separate context of each form with a single engine occurrence. When one form is in focus context of only that form is active and context of others are disabled. Hence by this way appropriate instructions for only one form are identified by the engine. In this way the commands fit to the one form are recognized by the engine. A individual recognize

can be utilized by many context. Sapi is good enough to develop the shared detected object for us naturally when the spsharedrecocontext is developed. In this scheme, dictation type of speech recognition is used. For achieving this, the grammar object is created and the grammar is loaded with SLOStatic value to set the dictation top of grammar as static. For setting this grammar object to be used dictation type of speech recognition, the Speech Rule State is initialized to state the property of grammar object to SGDSActive. In recognition event handler the ISpRecoResult interface is used by our application to restore data about the SR engines hypotheses recognition, and false recognition.

3. Natural Language Processing (NLP):- Natural language processing (NLP) consists of two techniques which are as follows: 1) Natural Language Understanding (NLU). 2) Natural Language Generation. Once the natural language is received by the computer, the most difficult part for NLP is understanding and providing meaning to it. Firstly computer must take input of natural language and convert it into artificial language. This is what speech-to-text or speech recognition are supposed to do. Speech to text is first step of NLU. When data is in text form, at that time NLU can take place try to figure out the meaning of that text. Not only this, but also it will be processed and desired code will be generated. A language processor must carry out a number of different functions primarily based around syntax analysis and semantic analysis. The purpose of syntax analysis is twofold: First it checks if the string of words is well formed, the sentence is then divided into structure that shows the relationship between different words.

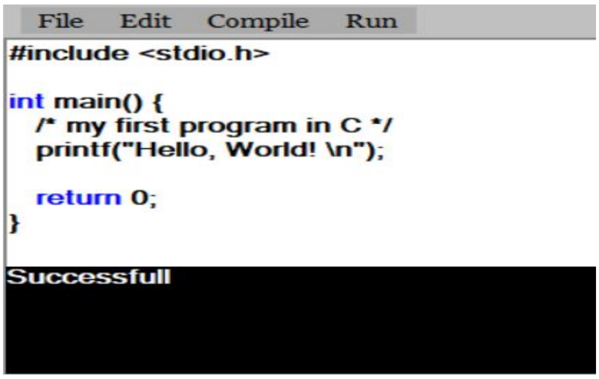
4. Code Generator: - The system takes the input through speech (voice) Commands and it processes it to generate the code i.e. the code is generated by analysing and processing the voice input. In first step, we find a list of reserved words of C; now for each reserved word we find words with same voice. For example we have following words with same voice: for, four, to two. After this, for each reserved word we developed a split list structure of words with same voice in C. When the user speaks some reserved word that is first converted into text and then this text will be mapped with the list of structure. If any match is found then this text is alter with reserved word, if not then is written as it is. Now if this text is wrong and user wants to delete that word user will speak “incorrect” or “delete”. A list of structure is also maintained for similar delivery of incorrect. If spoken word is matched with that same delivery then that word is deleted. At that time if match is found then reserved word has its own program construct then that Program construct generate simultaneously.

VI. ALGORITHM FOR STOP-WORD REMOVING

Step 1: First text is tokenized and specific words are reserve in array.
Step 2: A specific stop word is read from stop-word list.
Step 3: The stop word is correlated to text which is reserve in array using sequential search technique.
Step 4: If it maps the word in array is deleted, and the distinguished is stay till length of array.
Step 5: After deletion of stop-word finally, another stop-word is read from stop-word list and again algorithm follows step 2. The algorithm runs constantly until all the stop-words are distinguished.
Step 6: Resultant text devoid of stop-words is presented, also need statistics like stop-word deleted, no. of stop-words deleted from target text, total count of stop-words deleted from text, final count of words in text, resultant text and specific stop-word count in text is displayed.

VII. RESULTS

The voice input is given to system, the system then recognizes this and generates the required code. E.g.: Here the basic program is spoken by the user, these voice commands are then processed to generate the code. The command to compile the code can also be given through voice.

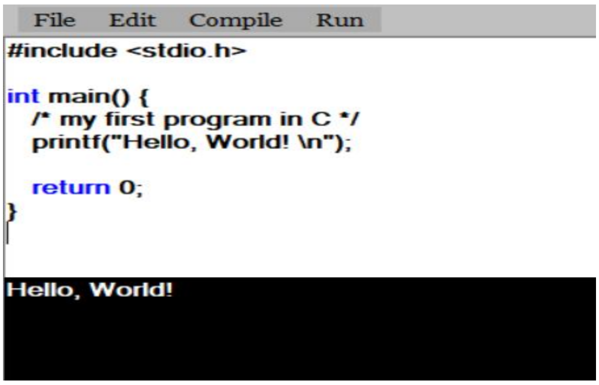


```
#include <stdio.h>

int main() {
   /* my first program in C */
   printf("Hello, World! \n");

   return 0;
}
```

```
Successfull
```

Figure 3. Implementation Example 1

It then compiles the code and shows if there exists any errors or if the compilation is successful.

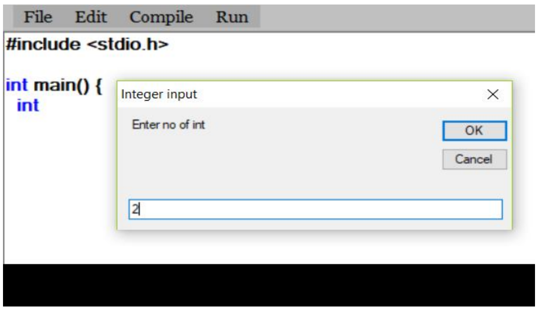


```
#include <stdio.h>

int main() {
   /* my first program in C */
   printf("Hello, World! \n");

   return 0;
}
```

```
Hello, World!
```

Figure 4. Implementation Example 2

The voice command can also be given to run the program and it gives the following output.



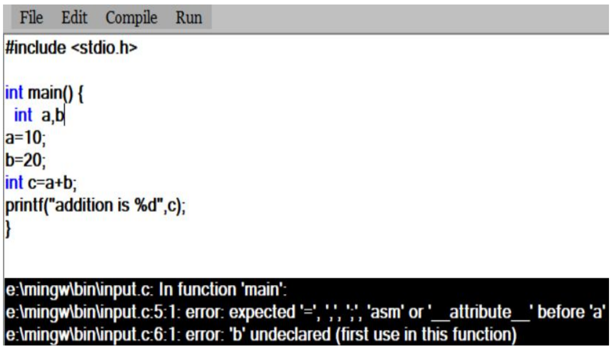
```
#include <stdio.h>

int main() {
  int
```

Enter no of int

2

Figure 5. Implementation Example 3

For placing the integers in the program, user or the programmer just need to speak “take integer” and it will ask you for the number of integers and the name of that integers.



```
#include <stdio.h>

int main() {
  int  a,b
a=10;
b=20;
int c=a+b;
printf("addition is %d",c);
}
```

```
e:\mingw\bin\input.c: In function 'main':
e:\mingw\bin\input.c:5:1: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'a'
e:\mingw\bin\input.c:6:1: error: 'b' undeclared (first use in this function)
```

Figure 5. Implementation Example 4

Here it shows the error after compilation, as semicolon is missing after int a.b.

VIII. CONCLUSION

Traditional way of software developing and programming is time consuming and tedious process, programmer wastes long hours in front of keyboard typing the entire codes. The voice based coding system provides efficient way to write the codes using voice commands. Implementation mainly consists of taking the voice commands from programmer, finding words that have similar sound to each reserved word of C programming language (e.g. for have similar sound as four), mapping them and then using Natural Language Processing (NLP) to generate the code accordingly. The input will be taken through voice and consequently the code will be written on editor i.e. instead of typing the entire structure of code, programmer just have to give a voice command for the structure they want and the code will be written accordingly along with the correct syntax on the editor. Special program constructs (e.g. function structure) are also developed for the ease of the user and thus user is free from the stress of recalling syntax and getting syntax errors. For future development this approach can be extended to all programming languages and also to visual programming languages. Future of voice programming can be extended to the level where user just needs to speak logic or algorithm and software will generate program accordingly. Proposed system is suitable for programmers, also for handicap people and peoples who suffering from constant strange injuries.

REFERENCES

[1] Jennifer L. Leopold and Allen L. Ambler ”key-boardless visual programming using voice, handwriting and gesture”,IEEE,10.1109/VL.1997.626555,**ISSN:**1049-2615. .

[2] Peter Bednr, “Vocabulary matching for information extraction language, IEEE 15th International

Symposium on Applied Machine Intelligence and Informatics,” Herlany, Slovakia, January 26-28, 2017.

[3] Mrd. Ing. Alexandru Trifan, Mrd. Ing. Marilena Anghelu, .L. Dr. Ing. Rodica Constantinescu “Natural Language Processing Model Compiling. Natural Language into Byte Code”.

[4] P. Bednr. , “Unified parsing and information extraction lan-guage. SAMI 2016 - IEEE 14th International Symposium on Applied Machine Intelligence and Informatics”, January 21-23,pp.131-1352016.

[5] Prof. Ahlam Ansari, Mirza Baig Shagufta titled “Constructing Test Cases Using Natural Language Process”.

[6] Jia Cui, Jonathan Mamou, Brian Kingsbury, and Bhuvana Ramabhadran, “Automatic keyword selection for keyword search development and tuning”, inProcICASSP.IEEE,. 2014,pp.78397843.

[7] Andrew Rosenberg, Kartik Audhkhasi, Abhinav Sethy, Bhuvana Ramabhadran, Michael Picheny, “End-To-End Speech Recognition And Keyword Search On Low-Resource Languages”, IEEE, 2017.

[8] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Phile- mon Brakel, and Yoshua Bengio, “End-to-end attention- based large vocabulary speech recognition”,
CoRR, vol. abs/1508.04395, 2015.

[9] Hagen Soltau, George Saon, and Brian Kingsbury,” The Ibm Attila Speech Recognition Toolkit”, IBM T. J. Watson Research Center,Yorktown Heights, NY 10598, USA.

[10] V. V. Digalakis, P. Monaco, and H. Murveit, “Genones: Gen- eralized mixture tying in continuous hidden Markov model- based speech recognizers”, IEEE Trans. Speech and Audio Processing, vol. 4, no. 4, pp. 281289, 1996.

[11] Junzo WATADA, IEEE, Hanayukia”Speech Recognition in a Multi-speakerEnvironment by using Hidden MarkovModel and Mel frequency Approach”.

[12] J Fiscus, J Ajot, and G Doddington,” The spoken term detection (std) 2006evaluation plan”, NIST USA, Sep, 2006.

[13] https://www.dmsitech.com/glossary-of-terms/voice reco-gnition/”, Refer from this article.

[14] https://www.computerhope.com/jargon/v/voicreco.htm Refer from this article

[15] http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C14_ADCdataAcquisition.htm.

[16] https://www.scribd.com/document/36605087/Voice-Rec-ognition-System-Report.

[17] https://medium.com/@mattkiser/an-introduction-to-nat-ural-language- procesing.