ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 3, March 2018

# Voice Recognition Scheme for Coding Console Using Natural Language Processing-A Survey

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

*Abstract* – **Recently, voice recognition is an alternative to 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. On the other hand, Natural Language Processing (NLP) concerns itself primarily with language communication between user and 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 a proposal of different approach to the NLP application and replacing the traditional code typing method with smart console recognizing the programmer's speech.**

**Keywords: Voice-Recognition,Machine-Learning, NLP:-Keyword Mapping, Keyword Extraction.**

## 1. 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. Today, this is succeeded on a computer with automatic voice detection ASR software programs. Many ASR programs need the user to train the ASR program to detect their speech so that it can more precisely convert the speech to text. 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 some examples of speech detection systems: 1) Speaker dependent system: The speech recognition requires training before it can be used which need 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 voice detection: The user must lapse between each word so that the speech recognition can determine each word separately. 4) Regular voice recognition-The speech recognition can realize a normal rate of 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) convert analog signal into digital data, so that system can easily understand. To do this it samples or digitizes the voice by taking 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.

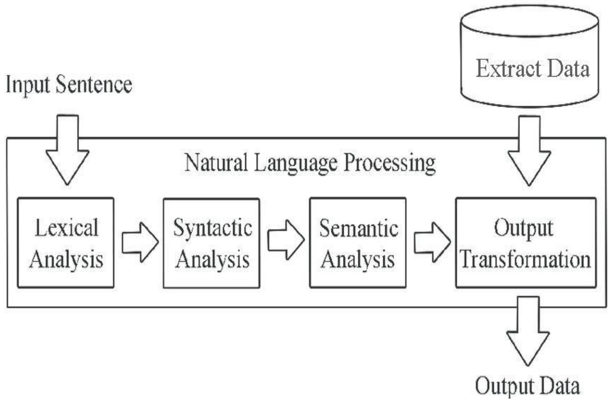

*Fig. 1. Natural Language Processing Diagram*

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 through a complicated statistical model and contrasts them to a large library of known words, phrases and sentences. The program then decides what the user was saying and either outputs it as text or issues 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, NLP enables real world application by utilizing human-computer interaction efficiently. 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

## II.MOTIVATION

Motivation of our project idea came from the real time problems faced by the coders while programming. Some of these problems includes following: Irritating syntax errors in huge pro-grams, time consuming typing speed for beginners, etc. So, to overcome this issues the proposed system brings an innovative idea of generating the code as per coder speaks. The motivation is to create platform where coder can invest their productivity in developing the logic instead of focusing on syntax of code. We are proposing the keyboard-less programming for smart console. The main focus of the proposed system in on Automatic syntax generation. Natural Language Processing (NLP) is focusing on natural language of coder which will automatically generate the code instead of typing manually.

## III. LITERATURE SURVEY

Mrd. Ing. Alexandra Trifan, Mrd. Ing. Marilena Anghelu, L. Dr. Ing. Rodica Constantinescu, titled Natural Language Pro-cessing Model Compiling Natural Language into Byte Code, they introduces a technical proposal of a different. 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 model [1].

Peter Bednr introduced a paper titled Vocabulary matching for information extraction language, in this paper author extends the previous specification 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].

Jennifer L. Leopold and Allen L. Ambler presented a paper titled keyboard less visual programming using voice, handwriting and gesture. In this paper authors present programming interface which give response to user communication activities. They pro-pose a multimodal user interface design. The majority of these systems have concentrated their efforts on voice recognition and gestures (from sources other than a pen), and have not included handwriting. The objective in developing a multimodal user interface to formulate is to improve the capacity to view edit and interact with visually oriented programs using voice and pen in-stead of the mouse and keyboard [3].

Peter Bednr introduced another paper titled Unified parsing and information extraction language, They describes declarative language for specification of parsing and Data abstraction rules used in the natural language processing work. The language unify concepts of tokenization using the regular expressions over the input text data abstraction design 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].

## IV. EXISTING SYSTEM

Coder uses text editors, notepad, IDEs for writing the program. On the other hand voice recognition is used in other fields. As speech detection improves, it is being executed in more places and its very likely 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 - Google voice is a service that allows you to search and ask questions on your computer, tablet, and phone.3) Siri - Today a good example of speech detection that help to give answer to question on direct apple devices. 4) Car Bluetooth- Today cars with Bluetooth phone pairing can use speech recognition to make instructions such as call my wife to make calls without taking your eyes off the road.

## V. 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 work flow of the system is as follows:

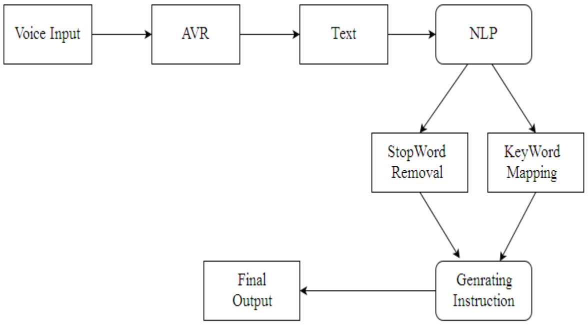

*Fig. 2. Proposed System Workflow Diagram*

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 listened words.

SAPI 5.4 provides two types of recognizer i.e. in-process recognizer (SpinProcRecognizer) and shared process recognizer (SpSharedRecognizer). The inprocess recognizer is responsible for claiming the resources for applications. E.g. if an inprocess recognizer claims the microphone of the system , no other application is allowed to use it. 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;

## VI. 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 gives the visual presentation of the virtual file system to the end user. GUI colour 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.

2. Voice to Text Converter: - The foundation of proposed system is speech to text conversion. In proposed system we used windows speech recognition WSR which is created on top of SAPI for grab speech and Microsoft Speechlib API to translate this 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 appopriate 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 recognizer 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 our scheme we are using dictation type of speech recognition. For this purpose we created a grammar object and load the grammar with SLOStatic value to set the dictation top of grammar as static. To set this grammar object to use dictation type of speech recognition we initialize Speech Rule State to state 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.

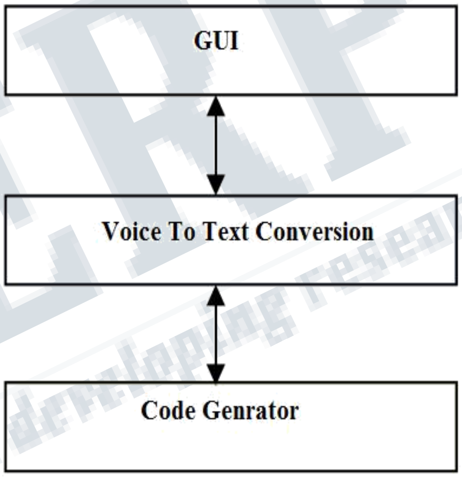

*Figure 3: Proposed System Architecture Diagram*

3. Natural Language Processing (NLP):- Natural language processing (NLP) consists of two methods are as follows: 1) Natural Language Understanding (NLU). 2) Natural Language Generation. Once the computer receives natural language 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 two-fold: 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 reserve 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.

## VII. ALGORITHM

Algorithm for removing stop-word. 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.

## VIII. CONCLUSION

Traditional way of software developing and programming is time consuming and tedious, programmers have to waste long hours in front of keyboard typing the entire codes.Disabled people or Proposed system makes it easier and time efficient by enabling programmers to use voice recognition for coding. Implementation mainly consists of taking the voice commands ,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 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 and the code will be written accordingly. 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. Proposed system is suitable for programmers, handicap people and peoples who suffering from constant strange injuries. 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.

## REFERENCES

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

[2] Peter Bednr, "Vocabulary matching for information extrac-tion language, EEE 15th International Symposium on Applied Machine Intelligence and Informatics," , Herlany, Slovakia, January 26-28, 2017.

[3] Jennifer L. Leopold and Allen L. Ambler "key-boardless visual programming using voice, handwriting and gesture".

[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.