# DESIGN AND IMPLEMENTATION OF VOICE BASED PERSONAL ASSISTANT

*This project report is submitted to*

*Rashtrasant Tukadoji Maharaj Nagpur University*

*in the partial fulfillment of the requirement for the award of the degree*

*of*

**Bachelor of Engineering in Computer Technology**

*By*

**Ms. Mansi Joshi (CT15067)**

**Mr. Rishi Kumar (CT15113)**

**Mr. Bhupender Yadav (CT15115)**

**Mr. Aditya Gaurav (CT15117)**

*Under the guidance of*

**Mr. Manish Sharma**

Assistant Professor

**2017-18**

**DEPARTMENT OF COMPUTER TECHNOLOGY**

**KAVIKULGURU INSTITUTE OF TECHNOLOGY AND SCIENCE**

**RAMTEK – 441 106**

**DEPARTMENT OF COMPUTER TECHNOLOGY**

**KAVIKULGURU INSTITUTE OF TECHNOLOGY AND SCIENCE**

**RAMTEK – 441 106**

# CERTIFICATE

This is to certify that the project report entitled **“Design And Implementation of Voice Based Personal Assistant”** carried out by Ms. Mansi Joshi (CT15067), Mr. Rishi Kumar (CT15113), Mr. Bhupender Yadav(CT15115) and Mr. Aditya Gaurav (CT15117) of the B.E. third year of computer technology, during the academic year 2017-2018, in the partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering (Computer Technology)** offered by the **Rashtrasant Tukadoji Maharaj Nagpur University,** Nagpur.

Mr. Manish Sharma
**Project Guide**

Mr. V. P. Mahatme
**Head of the Department**

Dr. B. Ram Rathan Lal
**Principal**

Date:

Place: Ramtek

# DECLARATION

We declare that

a. The work contained in this project has been done by as under the supervision of our guide.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. We have followed the guidelines provided by the Institute in preparing the project report.

d. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever we have used material (data, theoretical analysis, figures and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

**Project-mates**

# ACKNOWLEDGEMENT

We are grateful to our respected guide **Mr. Manish Sharma,** for his kind, disciplined and invaluable guidance which inspired us to solve all the difficulties that came across during completion of project.

We express our special thanks to **Mr. V. P. Mahatme,** Head of the Department, for his kind support, valuable suggestion and allowing us to use all facilities that are available in the Department during this project.

Our sincere thanks to **Dr. B. Ram Rathan Lal**, Principal, for extending all the possible help and allowing us to use all resources that are available in the Institute.

We are also thankful to our **Parents** and **Friends** for their valuable cooperation and standing with us in all difficult conditions.

**Project-mates**

# ABSTRACT

New technologies in the field of Artificial Intelligence could be harnessed to create an intelligent Virtual Personal Assistant (VPA) with a focus on user-based data. It is the need of the hour to design intelligent programs with capabilities to provide the user with the accurate information and deliver the services with performance and minimal efforts. In the domain of natural language processing numerous software are available in different programming language, with potential to prove usefulness in giving services to human as a VPA.

This engages the ability to communicate socially through natural language processing, holding and analysing data within the context of the user. It is suggested that new technologies may soon make the idea of virtual personal assistants a reality. Experiments conducted on this system, combined with user testing, have provided evidence that a basic program with natural language processing algorithms in the form of a VPA, with basic natural language processing and the ability to function without the need for other type of human input (or programming) is viable.

**KEYWORDS:** Virtual Personal Assistant, Natural Language Processing, Artificial Intelligence, Tokenization, Speech Recognition

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

CHAPTER 1

# INTRODUCTION

A personal assistant also referred to as personal aide (PA) or personal secretary (PS) is a job title describing a person who assists a specific person with their daily business or personal tasks.

Speech is the vocalized form of communication used by humans and some animals, which is based upon the syntactic combination of items drawn from the lexicon. Each spoken word is created out of the phonetic combination of a limited set of vowel and consonant speech sound units (phonemes). These vocabularies, the syntax that structures them and their sets of speech sound units differ, creating many thousands of different, and mutually unintelligible, human languages. The vocal abilities that enable humans to produce speech also enable them to sing.

One of the most notable advantages of speech recognition technology includes the dictation ability it provides. With the help of the technology users can easily control devices and create documents by speaking. Speech recognition can allow documents to be created faster because the software generally produces words as fast as they are spoken, which is generally much faster than a person can type. Dictation solutions are not only used by individuals but also by organizations which require heavy transcription tasks such as healthcare and legal.

Speech recognition technology also makes invaluable contributions to organizations. Businesses which provide customer services benefit from the technology in order to improve self-service in a way that enriches customer experience and reduces organizational costs. With the help of the voice recognition technology callers can input information such as name, account number, the reason of their call etc. without interacting with a live agent. Instead of having callers remain idly on hold while agents are busy, organizations can engage their callers without live customer representatives. That is why speech recognition technology contributes to cost savings by minimizing or even eliminating the need of live agents while improving customer experience.

Call centers which are continually challenged to balance customer satisfaction with cost containment apply voice recognition technology in order to benefit from invaluable advantages of the technology. Speech recognition technology:

- delivers a great customer experience while improving self-service system's containment rate
- encourages natural, human-like conversations that create more satisfying self-service interactions with customers
- automates what touchtone cannot by collecting dynamic data such as names and addresses
- enables organizations save agents for more important tasks

Speech Recognition technology act as an effective customer services automation tool for many organizations, notably call centers. Speech Recognition which differentiates with has a high recognition accuracy it provides as a result of its state-of-the-art acoustic model:

- transforms complicated menus to easy-to-use systems
- is compatible with all major operating systems and speech standards
- improves security by eliminating agents from data processing
- provides an alternative for the customers who prefer to skip touch-tone menus and connect to agents
- guarantees better customer service experience, increased customer satisfaction and decreased operational costs

This report summarizes the voice based personal assistant using the speech recognition technique .Speech recognition is an interesting and successful application for natural language processing, and has wide scope in future technologies.

## 1.1 Motivation

This project gets the motivation from the inability of the blind and disabled people for not using the computer even it's the basic task.

The virtual personal assistant will play a vital role in Interaction of disabled people with the computer system for their basic task as well as entertainment purpose.

## 1.2 Project Objective

- To understand speech recognition and its fundamentals
- To understand concept of Artificial Intelligence
- To understand the Natural Language Processing
- Development of software that can be mainly used for
    1. Playing Music
    2. Playing videos
    3. Internet surfing
    4. Notepad application
    5. Basic conversation with user

## 1.3 Project Scope

This project has the speech recognizing and speech synthesizing capabilities it can interact easily with the user, understand users commands and respond to it. This software also can open software's such as Notepad, Media Player, Browser and more.

## 1.4 Organization of Report

Chapter one contains the introduction of project which describes the voice recognition and features of software.

Chapter two includes the literature review which refers the study design and implementation of proposed work.

Chapter three explains proposed approach of system architecture which includes the detail of how the system has been developed with different components and modules.

Chapter four contains the results, testing and the snapshots of the software application while performing different functions.

Chapter five includes the various snapshots of the result generated from the project maintaining every necessary output need to describe precisely.

# CHAPTER 2
# LITERATURE REVIEW

Voice Controlled Personal Assistant System will use the Natural language processing and can be integrated with artificial intelligence techniques to achieve a smart assistant that can control IoT applications and even solve user queries using web searches. It can be designed to minimize the human efforts to interact with many other subsystems, which would otherwise have to be performed manually. By achieving this, the system will make human life comfortable. More specifically, this system is designed to interact with other subsystems intelligently and control these devices, this includes IoT devices or getting news from Internet, providing other information, getting personalized data saved previously on the system, etc. The android application should let the user add data such as calendar entries, set alarm, or even reminders.

## 2.1 Virtual Personal Assistant Using Raspberry Pi

**Asst. Prof. Emad S. Othman (2017)** proposed a system on Speech recognition and conversion to text. Speech has not been used much in the field of electronics and computers due to the complexity and variety of speech signals and sounds. However, with modern processors, complex algorithms and methods we can process speech signals to convert to text. Thus, proposed application deals with display of text from speech on a monitor using Android mobile, Bluetooth and Raspberry Pi. This application is quite useful in classrooms and presentations. A speech-to-text conversion and display can also improve system accessibility by providing data entry options for blind, deaf, or physically handicapped users. The code for application program in raspberry pi is written using Python programming language.

## 2.2 Natural Language Processing

**Steven Bird, Ewan Klien and Edward Looper (2017)** researches NLP which aim to gather knowledge on how human beings tend to understand and use the language so that appropriate tools and techniques can be developed to make computer systems understand

and manipulate natural languages to perform the desired Phonological rules are captured through machine learning on training sets. Pronunciation dictionaries are also used for both text-to-speech and automatic speech recognition. Sounds as well as words can be predicted by using the conditional probability theory the input to a speech recognizer is a series of acoustic waves. The waves are then sampled, quantified and literally converted to spectral representation. The method of Conditional probability is then used to evaluate each vector of the spectral representation with a system of stored phonetic representation.

## 2.3 Personal Assistant with Voice Recognition Intelligence

**Dr. Kotrappa Sirbi, Mr. Abhijit J.Patankar and Dr. Kshama V.Kulhalli.(2017)** analyzed the Speech recognition with several waves of major innovations. Speech recognition for dictation, search, and voice commands has become a standard feature on smartphones and wearable devices. Design of a compact large vocabulary speech recognition system that can run efficiently on mobile devices, accurately and with low latency. This is achieved by using a CTC based LSTM acoustic model which predicts context independent phones and is compressed to a tenth of its original size using a combination of SVD-based compression and quantization. Quantized deep neural networks (DNNs) and on-the-fly language model rescoring to achieve real-time performance on modern smartphones. The ASR and Search components perform speech recognition and search tasks. In addition to ASR and Search, they also integrate a query parsing module between ASR and Search for a number of reasons. Set of techniques for improving the performance of automated voice search services intended for mobile users accessing these services over a range of portable devices. Voice search is implemented as a two stage search procedure where string candidates generated by automatic speech recognition (ASR) system are re-scored in order to identify the best matching entry from a potentially very large application specific database.

## 2.4 Voice Based System

A virtual assistant is a software agent that can perform tasks or services for an individual. Sometimes the term "chatbot" is used to refer to virtual assistants generally or specifically those accessed by online chat(or in some cases online chat programs that are for entertainment and not useful purposes).

As of 2017, the capabilities and usage of virtual assistants are expanding rapidly, with new products entering the market. An online poll in May 2017 found the most widely used in the US were Apple's Siri (34%), Google Assistant (19%), Amazon Alexa (6%), and Microsoft Cortana (4%).Apple and Google have large installed bases of users on Smartphone’s. Microsoft has a large installed base of Windows-based personal computers; Smartphone’s and smart speakers. Alexa has a large install base for smart speakers.

## 2.5 An Overview of Speech Recognition

Speech recognition is a technology that able a computer to capture the words spoken by a human with a help of microphone. These words are later on recognized by speech recognizer, and in the end, system outputs the recognized words. The process of speech recognition consists of different steps that will be discussed in the following sections one by one. An ideal situation in the process of speech recognition is that, a speech recognition engine recognizes all words uttered by a human but, practically the performance of a speech recognition engine depends on number of factors. Vocabularies, multiple users and noisy environment are the major factors that are counted in as the depending factors for a speech recognition engine.

## 2.6 History of Speech Recognition

The concept of speech recognition started somewhere in 1940s, practically the first speech recognition program was appeared in 1952 at the bell labs, that was about recognition of a digit in a noise free environment. 1940s and 1950s consider as the foundational period of the speech recognition technology, in this period work was done on the foundational paradigms of the speech recognition that is automation and information theoretic models . In the 1960’s we were able to recognize small vocabularies (order of 10-100 words) of isolated words, based on simple acoustic phonetic properties of speech sounds. The key technologies that were developed during this decade were, filter banks and time normalization methods in 1970s the medium vocabularies (order of 100-1000 words) using simple template-based, pattern recognition methods were recognized. In 1980s large vocabularies (1000-unlimited) were used and speech recognition problems based on statistical, with a large range of networks for handling language structures were addressed. The key invention of this era was hidden Markova model (HMM) and the stochastic language model, which together Continuous speech recognition enabled powerful new methods for handling problem efficiently and with high performance. In 1990s the key technologies developed during this period were the methods for stochastic language understanding, statistical learning of acoustic and language models, and the methods for implementation of large vocabulary speech understanding systems. After the five decades of research, the speech recognition technology has finally entered marketplace, benefiting the users in variety of ways. The challenge of designing a machine that truly functions like an intelligent human is still a major one going forward.

## 2.7 Types of Speech Recognition

Speech recognition systems can be divided into the number of classes based on their ability to recognize that words and list of words they have. A few classes of speech recognition are classified as under:

### 2.7.1 Isolated Speech

Isolated words usually involve a pause between two utterances; it doesn't mean that it only accepts a single word but instead it requires one utterance at a time .

A feature extraction method for isolate speech recognition is proposed, which is based on a time frequency analysis using a critical band concept similar to that performed in the inner ear model; which emulates the inner ear behavior by performing signal decomposition, similar to carried out by the basilar membrane. Evaluation results show that the proposed method performs better than other previously proposed feature extraction methods, when it is used to characterize normal as well as esophageal speech signal.

### 2.7.2 Connected Speech

Connected words or connected speech is similar to isolated speech but allow separate utterances with minimal pause between them. Recognition is a technique that is sometimes used in continuous-speech applications. In this technique, the sentence is decoded by patching together models built from discrete words and matching the complete utterance to these concatenated models. The system usually does not attempt to model word boundary allophonic effects, nor sloppy intra or inter-word articulation. There is an implicit assumption that, while distinct boundaries cannot be located among words, the words are reasonably well articulated. The accuracy of the system could be increased when probabilistic relationships among words (syntax) are known.

### 2.7.3 Continuous Speech

Continuous speech allow the user to speak almost naturally, it is also called the computer dictation. Speech is most common mode of communication between human. Human are trying to develop systems which can understand and accept the command via speech. Speech is the most commonly and widely used form of communication between humans. There are various spoken languages which are used throughout the world. The communication among the human being is mostly done by vocally, therefore it is natural for people to expect speech interfaces with computer. Since early 1960's researchers are trying to develop system which can record, interpret and understand human speech. The use of speech for interacting with the computer may help the developing nations as the language technologies can be implemented for the e-governance system. Speech recognition (SR) means translation of spoken words the text or commands. Development of Speech recognition systems has attained new heights but robustness and noise tolerant recognition systems are few of the problems which make speech recognition systems inconvenient to use . Many Research projects have been completed and currently in progress around the world for the development of robust speech recognition systems. The paper presents the review of the continuous speech recognition systems.

### 2.7.4 Spontaneous Speech

At a basic level, it can be thought of as speech that is natural sounding and not rehearsed. An ASR system with spontaneous speech ability should be able to handle a variety of natural speech features such as words being run together, "ums" and "ahs", and even slight stutters.

## 2.8 Speech Recognition Process

### 2.8.1 Components of Speech Recognition System

**Voice Input**

With the help of microphone audio is input to the system, the pc sound card produces the equivalent digital representation of received audio.

**Digitization**

The process of converting the analog signal into a digital form is known as digitization, it Involves the both sampling and quantization processes. Sampling is converting a continuous Signal into disc-rete signal, while the process of approximating a continuous range of values is known as quantization.

**Acoustic Model**

An acoustic model is created by taking audio recordings of speech, and their text transcriptions, and using software to create statistical representations of the sounds that make up each word. It is used by a speech recognition engine to recognize speech. The software acoustic model breaks the words into the phonemes.

**Language Model**

Language modeling is used in many natural language processing applications such as speech recognition tries to capture the properties of a language and to predict the next word in the speech sequence. The software language model compares the phonemes to words in its built in dictionary.

**Speech engine**

The job of speech recognition engine is to convert the input audio into text; to accomplish this it uses all sorts of data, software algorithms and statistics. Its first operation is digitization as discussed earlier, that is to convert it into a suitable format for further processing. Once audio signal is in proper format it then searches the best match for it. It does this by considering the words it knows, once the signal is recognized it returns its corresponding text string.

## 2.9 Uses of Speech Recognition Programs

Basically speech recognition is used for two main purposes. First and foremost dictation that is in the context of speech recognition is translation of spoken words into text, and second controlling the computer, that is to develop such software that probably would be capable enough to authorize a user to operate different application by voice . Writing by voice let a person to write 150 words per minute or more if indeed he/she can speak that much quickly. This perspective of speech recognition programs create an easy way for composing text and help the people in that industry to compose millions of words digitally in short time rather than writing them one by one, and this way they can save their time and effort. Speech recognition is an alternative of keyboard. If you are unable to write or just don’t want to type then programs of speech recognition helps you to do almost anything that you used to do with keyboard.

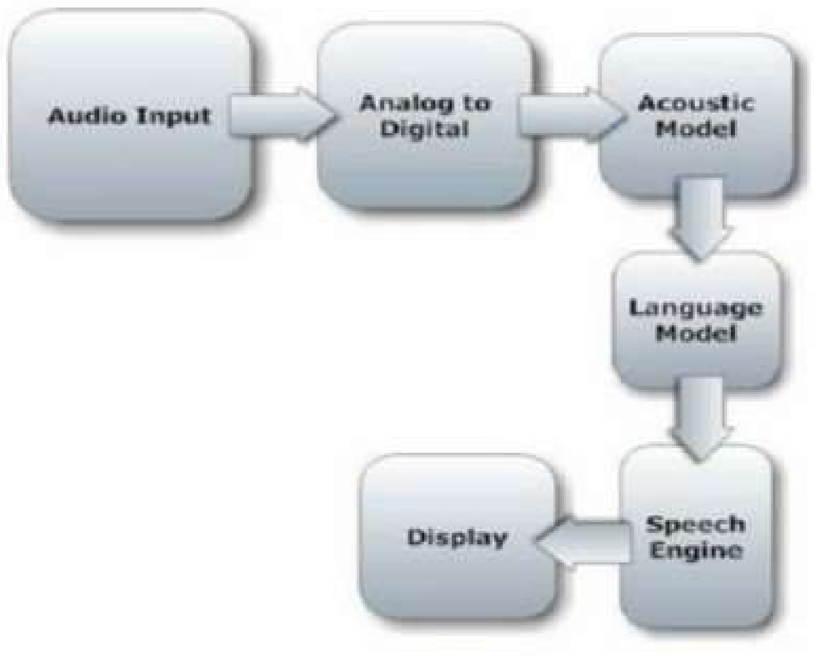


Fig: 2.1 Speech recognition process

## 2.10 Applications

### 2.10.1 From Medical Perspective

People with disabilities can benefit from speech recognition programs. Speech recognition is especially useful for people who have difficulty using their hands, in such cases speech recognition programs are much beneficial and they can use for operating computers. Speech recognition is used in deaf telephony, such as voicemail to text.

### 2.10.2 From Military Perspective

Speech recognition programs are important from military perspective; in Air Force speech recognition has definite potential for reducing pilot workload. Beside the Air force such Programs can also be trained to be used in helicopters.

## 2.11 Speech Recognition Weakness and Flaws

Besides all these advantages and benefits, yet a hundred percent perfect speech recognition system is unable to be developed. There are number of factors that can reduce the accuracy and performance of a speech recognition program.

Speech recognition process is easy for a human but it is a difficult task for a machine, comparing with a human mind speech recognition programs seems less intelligent, this is due to that fact that a human mind is God gifted thing and the capability of thinking,

**Few factors those are considerable in this regard**

Understanding and reacting is natural, while for a computer program it is a complicated task, first it need to understand the spoken words with respect to their meanings, and it has to create a sufficient balance between the words, noise and spaces. A human has a built in capability of filtering the noise from a speech while a machine requires training, computer requires help for separating the speech sound from the other sounds.

**Homonyms**

Are the words that are differently spelled and have the different meaning but acquires the same meaning, for example "there" "their" "be" and "bee". This is a challenge for computer machine to distinguish between such types of phrases that sound alike.

**Overlapping speeches**

A second challenge in the process, is to understand the speech uttered by different users, current systems have a difficulty to separate simultaneous speeches from multiple users.

**Noise factor**

The program requires hearing the words uttered by a human distinctly and clearly. Any extra sound can create interference, first you need to place system away from noisy environments and then speak clearly else the machine will confuse and will mix up the words

## 2.12 Introduction to Natural Language Processing

NLP is a branch of data science that consists of systematic processes for analyzing, understanding, and deriving information from the text data in a smart and efficient manner. By utilizing NLP and its components, one can organize the massive chunks of text data, perform numerous automated tasks and solve a wide range of problems such as – automatic summarization, machine translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation etc. Before moving further, I would like to explain some terms that are used in the article:

- Tokenization – process of converting a text into tokens
- Tokens – words or entities present in the text
- Text object – a sentence or a phrase or a word or an article

## 2.13 Text Preprocessing

Since, text is the most unstructured form of all the available data, various types of noise are present in it and the data is not readily analyzable without any pre-processing. The entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text preprocessing. It is predominantly comprised of three steps:

- Noise Removal
- Lexicon Normalization
- Object Standardization

The following image shows the architecture of text preprocessing pipeline.

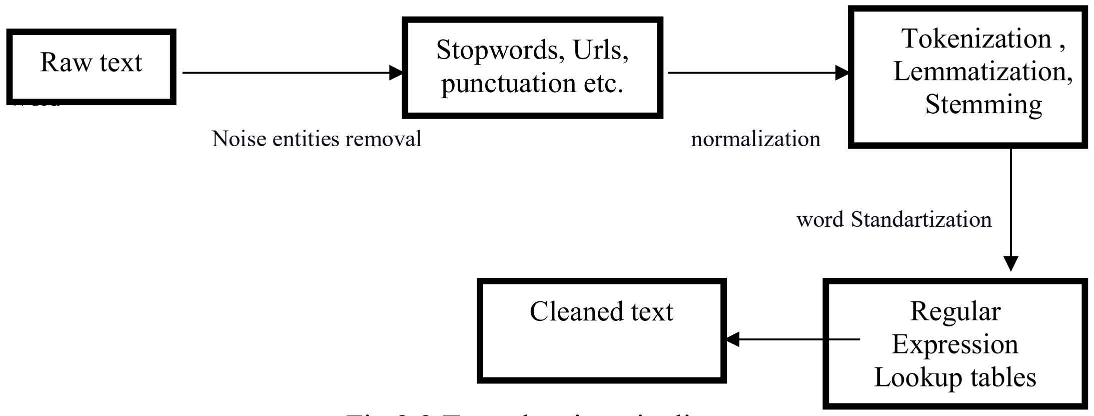

Raw text
Noise entities removal
Stopwords, Urls, punctuation etc.
normalization
Tokenization , Lemmatization, Stemming
word Standartization
Regular Expression Lookup tables
Cleaned text

Fig 2.2 Text cleaning pipeline

### 2.13.1 Noise Removal

Any piece of text which is not relevant to the context of the data and the end-output can be specified as the noise. For example – language stopwords (commonly used words of a language – is, am, the, of, in etc), URLs or links, social media entities (mentions, hashtags), punctuations and industry specific words. This step deals with removal of all types of noisy entities present in the text. A general approach for noise removal is to prepare a dictionary of noisy entities, and iterate the text object by tokens (or by words), eliminating those tokens which are present in the noise dictionary.

### 2.13.2 Lexicon Normalization

Another type of textual noise is about the multiple representations exhibited by single word. For example – "play", "player", "played", "plays" and "playing" are the different variations of the word – "play", Though they mean different but contextually all are similar. The step converts all the disparities of a word into their normalized form (also known as lemma). Normalization is a pivotal step for feature engineering with text as it converts the high dimensional features (N different features) to the low dimensional space (1 feature), which is an ideal ask for any ML model. The most common lexicon normalization practices are:

- **Stemming:** Stemming is a rudimentary rule-based process of stripping the suffixes ("ing", "ly", "es", "s" etc) from a word.
- **Lemmatization:** Lemmatization, on the other hand, is an organized & step by step procedure of obtaining the root form of the word, it makes use of vocabulary (dictionary importance of words) and morphological analysis (word structure and grammar relations).

### 2.13.3 Object Standardization

Text data often contains words or phrases which are not present in any standard lexical dictionaries. These pieces are not recognized by search engines and models. Some of the examples are – acronyms, hash tags with attached words, and colloquial slangs. With the help of regular expressions and manually prepared data dictionaries, this type of noise can be fixed, the code below uses a dictionary lookup method to replace social media slangs from a text.

## 2.14 NLP problems / tasks

- **Text Summarization** – Given a text article or paragraph, summarize it automatically to produce most important and relevant sentences in order.
- **Machine Translation** – Automatically translate text from one human language to another by taking care of grammar, semantics and information about the real world, etc.
- **Natural Language Generation and Understanding** – Convert information from computer databases or semantic intents into readable human language is called language generation. Converting chunks of text into more logical structures that are easier for computer programs to manipulate is called language understanding.
- **Optical Character Recognition** – Given an image representing printed text, determine the corresponding text.
- **Document to Information** – This involves parsing of textual data present in documents (websites, files, pdfs and images) to analyzable and clean format.

## 2.15 Important Libraries for NLP (python)

- Scikit-learn: Machine learning in Python
- Natural Language Toolkit (NLTK): The complete toolkit for all NLP techniques.
- Pattern – A web mining module for the with tools for NLP and machine learning.
- TextBlob – Easy to use nlp tools API, built on top of NLTK and Pattern.
- spaCy – Industrial strength NLP with Python and Cython.
- Gensim – Topic Modelling for Humans
- Stanford Core NLP – NLP services and packages by Stanford NLP Group.

## 2.16 GTTS API(Google –text-to –speech)

Google Text-to-Speech is a screen reader application developed by Google for its Android operating system. It powers applications to read aloud (speak) the text on the screen which support many languages. Text-to-Speech may be used by apps such as Google Play Books for insight to the pronunciation of words, by Google Talkback and other spoken feedback accessibility-based applications, as well as by third-party apps. Users must install voice data for each language.

### 2.16.1 Supported Languages

Currently, languages supported by Google Text-to-Speech include, Bangla (Bangladesh), Bangla (India), Cantonese (Hong Kong), Czech, Danish, Dutch, English (Australia), English (India), English (United Kingdom), English (United States), Estonian, Filipino, Finnish, French, German, Greek, Hindi, Hungarian, Indonesian, Italian, Japanese, Khmer, Korean, Mandarin (China), Mandarin (Taiwan), Nepali, Norwegian, Polish, Portuguese (Brazil), Romanian, Russian, Sinhala, Slovak, Spanish (Spain), Spanish (United States), Swedish, Tamil,Thai, Turkish, Ukrainian and Vietnamese.

### 2.16.2 Evolution

Some app developers have started adapting and tweaking their Android Auto apps to include Text to Speech, such as Hyundai in 2015. Apps such as textPlus and WhatsApp_use Text to Speech to read notifications aloud and provide voice-reply functionality.

## 2.17 Python (Programming Language)

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.
Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

### 2.17.1 History of Python Language

Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system. Van Rossum
remains Python's principal author. His continuing central role in Python's development is reflected in the title given to him by the Python community: Benevolent Dictator For Life (BDFL).On the origins of Python, Van Rossum wrote in 1996:Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release, the development process became more transparent and community-backed. Python 3.0 (initially called Python 3000 or py3k) was released on 3 December 2008 after a long testing period. It is a major revision of the language that is not completely backward-compatible with previous versions. However, many of its major features have been back ported to the Python 2.6.x and 2.7.x version series, and releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.Python 3.6 had changes regarding UTF-8 (in Windows, PEP 528 and PEP 529) and Python 3.7.0b1 (PEP 540) adds a new "UTF-8 Mode" (and overrides POSIX locale).In January 2017, Google announced work on a Python 2.7 to Go transcompiler to improve performance under concurrent workloads.

### 2.17.2 Syntax and semantics

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

### 2.17.3 Indentation

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. This feature is also sometimes termed the off-side rule.

### 2.17.4 Statements and Control Flow

- The **with** statement, from Python 2.5 released on September 2006, which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior and replaces a common try/finally idiom.
- The **pass** statement, which serves as a NOP. It is syntactically needed to create an empty code block.
- The **assert** statement, used during debugging to check for conditions that ought to apply.
- The **yield** statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.
- The **import** statement, which is used to import modules whose functions or variables can be used in the current program. There are four ways of using import: import <module name> or from <module name> import * or import numpy as np or from numpy import pi as Pie.
- The **print** statement was changed to the print() function in Python 3.

### 2.17.5 Libraries

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing. Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wagered follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations. As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Documentation
- System administration
- Scientific computing
- Text processing
- Image processing

## 2.18 CMU Sphinx

CMU Sphinx, also called Sphinx in short, is the general term to describe a group of speech recognition systems developed at Carnegie Mellon University. These include a series of speech recognizers (Sphinx 2 - 4) and an acoustic model trainer (Sphinx Train).In 2000, the Sphinx group at Carnegie Mellon committed to open source several speech recognizer components, including Sphinx 2 and later Sphinx 3 (in 2001). The speech decoders come with acoustic models and sample applications. The available resources include in addition software for acoustic model training, Language model compilation and a public domain pronunciation dictionary, cmudict.

### 2.18.1 Sphinx

Sphinx is a continuous-speech, speaker-independent recognition system making use of hidden Markov acoustic models (HMMs) and an n-gram statistical language model. It was developed by Kai-Fu Lee. Sphinx featured feasibility of continuous-speech, speaker-independent large-vocabulary recognition, the possibility of which was in dispute at the time (1986). Sphinx is of historical interest only; it has been superseded in performance by subsequent versions.

### 2.18.2 Sphinx 2

A fast performance-oriented recognizer originally developed by Xuedong Huang at Carnegie Mellon and released as Open source with a BSD-style license on SourceForge by Kevin Lenzo at Linux World in 2000. Sphinx 2 focuses on real-time recognition suitable for spoken language applications. As such it incorporates functionality such as end-pointing, partial hypothesis generation, dynamic language model switching and so on. It is used in dialog systems and language learning systems. It can be used in computer based PBX systems such as Asterisk. Sphinx 2 code has also been incorporated into a number of commercial products. It is no longer under active development (other than for routine maintenance). Current real-time decoder development is taking place in the Pocket Sphinx project. An archival article describes the system.

### 2.18.3 Sphinx 3

Sphinx 2 used a semi-continuous representation for acoustic modeling (i.e., a single set of Gaussians is used for all models, with individual models represented as a weight vector over these Gaussians). Sphinx 3 adopted the prevalent continuous HMM representation and has been used primarily for high-accuracy, non-real-time recognition. Recent developments (in algorithms and in hardware) have made Sphinx 3 "near" real-time, although not yet suitable for critical interactive applications. Sphinx 3 is under active development and in conjunction with SphinxTrain provides access to a number of modern modeling techniques, such as LDA/MLLT, MLLR and VTLN, that improve recognition accuracy (see the article on Speech Recognition for descriptions of these techniques).

### 2.18.4 Sphinx 4

Sphinx 4 is a complete re-write of the Sphinx engine with the goal of providing a more flexible framework for research in speech recognition, written entirely in the Java programming language. Sun Microsystems supported the development of Sphinx and contributed software engineering expertise to the project. Participants included individuals at MERL, MIT and CMU. Current development goals include:

- Developing a new (acoustic model) trainer
- Implementing speaker adaptation (e.g. MLLR)
- Improving configuration management
- Creating a graph-based UI for graphical system design

### 2.18.5 PocketSphinx

A version of Sphinx that can be used in embedded systems (e.g., based on an ARM processor). PocketSphinx is under active development and incorporates features such as fixed-point arithmetic and efficient algorithms for GMM computation.

# CHAPTER 3
# PROPOSED APPROACH AND SYSTEM ARCHITECTURE

## 3.1 System architecture and requirements

The system architecture of voice based system is described below:

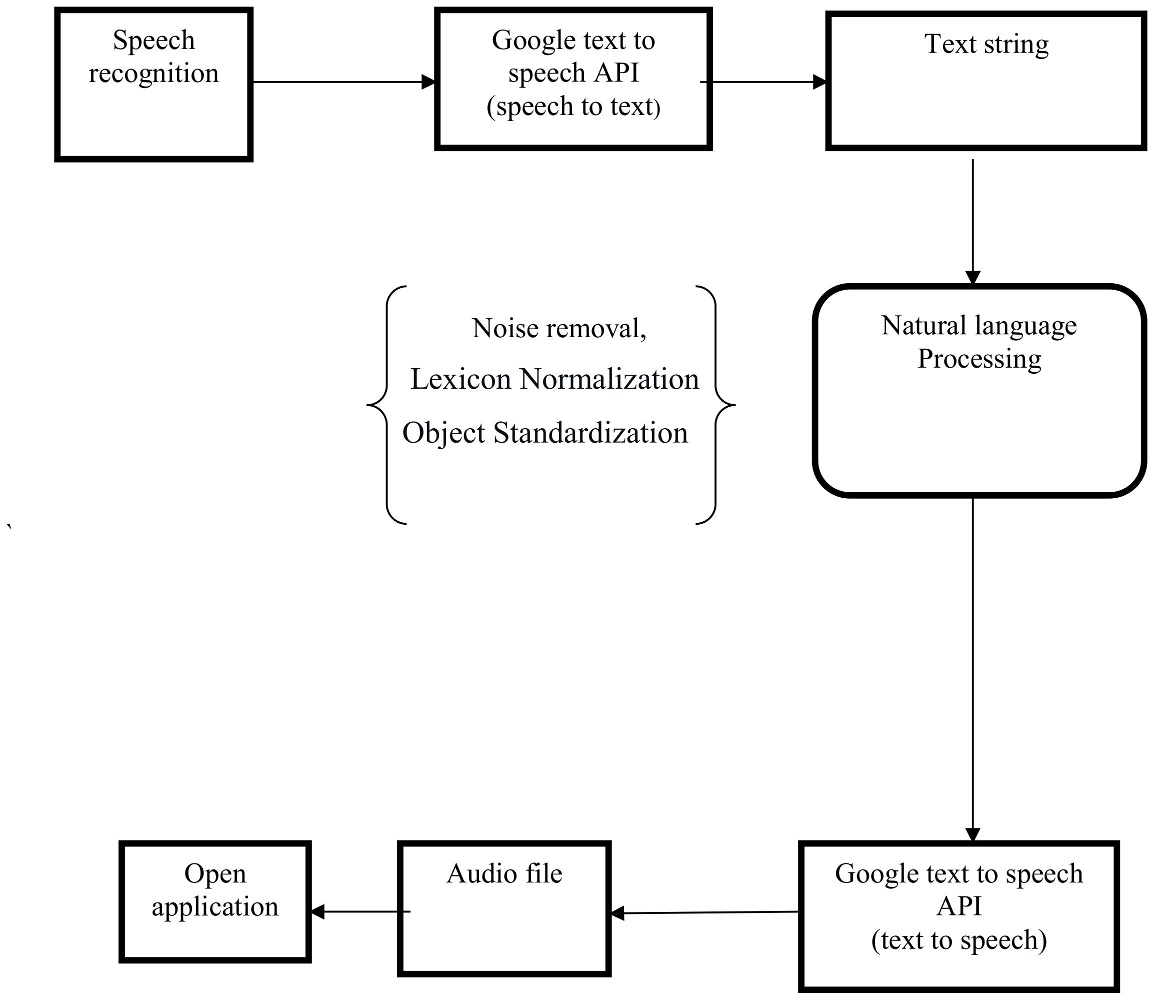


Fig 3.1 System architecture of voice based system

- **Speech**

  Speech is the vocalized form of communication used by humans and some animals, which is based upon the syntactic combination of items drawn from the lexicon. Each spoken word is created out of the phonetic combination of a limited set of vowel and consonant speech sound units (phonemes). These vocabularies, the syntax that structures them, and their sets of speech sound units differ, creating many thousands of different, and mutually unintelligible, human languages. The vocal abilities that enable humans to produce speech also enable them to sing.

  **Functions and libraries**

  **r.listen( )**

  This function is used to take input as speech

  **sr.Microphone()**

  This function is used to detect microphone while the input is taken.

  **sr.Recognizer()**

  This function is used to recognize audio input.

- **Speech to text:**

  Speech to text conversion is the process of converting spoken words into written texts. This process is also often called speech recognition. Although these terms are almost synonymous, Speech recognition is sometimes used to describe the wider process of extracting meaning from speech, i.e. speech understanding. The term voice recognition should be avoided as it is often associated to the process of identifying a person from their voice, i.e. speaker recognition.

  All speech-to-text systems rely on at least two models: an acoustic model and a language model. In addition large vocabulary systems use a pronunciation model. It is important to understand that there is no such thing as a universal speech recognizer. To get the best transcription quality, all of these models can be specialized for a given language, dialect, application domain, type of speech, and communication channel.

  Like any other pattern recognition technology, speech recognition cannot be error free. The speech transcript accuracy is highly dependent on the speaker, the style of speech and the environmental conditions. Speech recognition is a harder process than what

people commonly think, even for a human being. Humans are used to understanding speech, not to transcribing it, and only speech that is well formulated can be transcribed without ambiguity.
From the user's point of view, a speech-to-text system can be categorized based in its use: command and control, dialog system, text dictation, audio document transcription, etc. Each use has specific requirements in terms of latency, memory constraints, vocabulary size, and adaptive features.

**Functions and libraries**

**GTTS API**

Google text to speech API converts the speech to text

- **Natural language processing**

It is a computer activity in which computers are entailed to analyze, understand, alter, or generate natural language. This includes the automation of any or all linguistic forms, activities, or methods of communication, such as conversation, correspondence, reading, written composition, dictation, publishing, translation, lip reading, and so on. Natural language processing is also the name of the branch of computer science, artificial intelligence, and linguistics concerned with enabling computers to engage in communication using natural language(s) in all forms, including but not limited to speech, print, writing, and signing.

**Functions and libraries**

**NLTK (Natural language toolkit):**

This library is used to perform text processing operations ex. Tokenization ,object standardization etc.

- **Text to speech**

It is a form of speech synthesis that converts text into voice output. Text-To-Speech software basically takes the text you write and turns it into speech files that you can use.

A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read any text aloud, whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system. Let us try to be clear. There is a fundamental difference between the system we are about to discuss here and any other talking machine (as a cassette-player for example) in the sense that we are interested in the automatic production of new sentences. This definition still needs some refinements. Systems that simply concatenate isolated words or parts of sentences, denoted as Voice Response Systems, are only applicable when a limited vocabulary is required (typically a few one hundreds of words), and when the sentences to be pronounced respect a very restricted structure, as is the case for the announcement of arrivals in train stations for instance. In the context of TTS synthesis, it is impossible (and luckily useless) to record and store all the words of the language. It is thus more suitable to define Text-To-Speech as the automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter

**Functions and libraries**

**GTTS API**

Google text to speech API is used for conversion of text to speech

- **Open application**

  After the processing of the speech desired application opens using function calls

  **Functions and libraries**

  **process.kill()**

  This function is used to kill the ongoing process.

### 3.1.1 Minimum requirements

- Pentium 200 MHz processor
- 512Mb of RAM
- Microphone
- Sound card

### 3.1.2 Best requirements

- 1.6 GHz Processor
- 1 GB or more of RAM
- Sound cards with very clear signals

- High quality microphones

## 3.2 Hardware Requirements

- **Sound cards**

  Speech requires relatively low bandwidth, high quality 16 bit sound card will be better enough to work. Sound must be enabled, and proper driver should be installed. Sound cards with the 'cleanest' A/D (analog to digital) conversions are recommended, but most often the clarity of the digital sample is more dependent on the microphone quality and even more dependent on the environmental noise. Some speech recognition systems might require specific sound card.

- **Microphones**

  A quality microphone is key when utilizing the speech recognition system. Desktop microphones are not suitable to continue with speech recognition system, because they have tendency to pick up more ambient noise. The best choice, and most common is the headset style. It allows the ambient noise to be minimized, while allowing you to have the microphone at the tip of your tongue all the time. Headsets are available without earphones and with earphones (mono or stereo).

- **Computer/ Processors**

  Speech recognition applications can be heavily dependent on processing speed. This is because a large amount of digital filtering and signal processing can take place in ASR.

## 3.3 Python Libraries Used

### 3.3.1 psutil

psutil (process and system utilities) is a cross-platform library for retrieving information on running processes and system utilization (CPU, memory, disks, network, sensors) in Python. It is useful mainly for system monitoring, profiling and limiting process resources and management of running processes. It implements much functionality offered by UNIX command line tools such as: ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap. psutil currently supports the following platforms:

- Linux
- Windows
- OSX,
- FreeBSD, OpenBSD, NetBSD
- Sun Solaris
- AIX

### 3.3.2 Wikiapi

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the Media Wiki API so you can focus on using Wikipedia data, not getting it.

### 3.3.3 Subprocess

The subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions. The recommended approach to invoking subprocesses is to use the following convenience functions for all use cases they can handle. For more advanced use cases, the underlying Popen interface can be used directly.

subprocess.**call**(*args*, *, *stdin=None*, *stdout=None*, *stderr=None*, *shell=False*, *timeout= Non*)

### 3.3.4 OS

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module.

Notes on the availability of these functions:

- The design of all built-in operating system dependent modules of Python is such that as long as the same functionality is available, it uses the same interface; for example, the function os.stat(path) returns stat information about *path* in the same format (which happens to have originated with the POSIX interface).
- Extensions peculiar to a particular operating system are also available through the os module, but using them is of course a threat to portability.
- All functions accepting path or file names accept both bytes and string objects, and result in an object of the same type, if a path or file name is returned.
- An "Availability: UNIX" note means that this function is commonly found on Unix systems. It does not make any claims about its existence on a specific operating system.
- If not separately noted, all functions that claim "Availability: Unix" are supported on Mac OS X, which builds on a Unix core.

### 3.3.5 Webbrowser

The webbrowser module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the open() function from this module will do the right thing. Under Unix, graphical browsers are preferred under X11, but text-mode browsers will be used if graphical browsers are not available or an X11 display isn't available. If text-mode browsers are used, the calling process will block until the user exits the browser. If the environment variable BROWSER exists, it is interpreted as the os.pathsep-separated list of browsers to try ahead of the platform defaults. When the value of a list part contains the string %s, then it is interpreted as a literal browser command line to be used with the argument URL substituted for %s; if the part does not contain %s, it is simply interpreted as the name of the browser to launch. For non-Unix platforms, or when a remote browser is available on Unix, the controlling process will not wait for the user to finish with the browser, but allow the remote browser to maintain its own windows on the display. If remote browsers are not available on Unix, the controlling process will launch a new browser and wait. The script webbrowser can be used as a command-line interface for the module. It accepts an URL as the argument. It accepts the following optional parameters: -n opens the URL in a new browser window, if possible; -t opens the URL in a new browser page ("tab"). The options are, naturally, mutually exclusive.

### 3.3.6 NLTK

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as Word Net, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project. NLTK has been called "a wonderful tool for teaching, and working in, computational linguistics using Python," and "an amazing library to play with natural language."Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more

### 3.3.7 Threading

This module constructs higher-level threading interfaces on top of the lower level _thread module. See also the queue module. The dummy_threading module is provided for situations where threading cannot be used because _thread is missing.

This module defines the following functions:

- **threading.active_count():** Return the number of Thread objects currently alive. The returned count is equal to the length of the list returned by enumerate ().
- **threading.current_thread():** Return the current Thread object, corresponding to the caller's thread of control. If the caller's thread of control was not created through the threading module, a dummy thread object with limited functionality is returned.

- **threading.get_ident():** Return the ‘thread identifier’ of the current thread. This is a nonzero integer. Its value has no direct meaning; it is intended as a magic cookie to be used e.g. to index a dictionary of thread-specific data. Thread identifiers may be recycled when a thread exits and another thread is created.

## 3.4 User Interface

### Tkinter

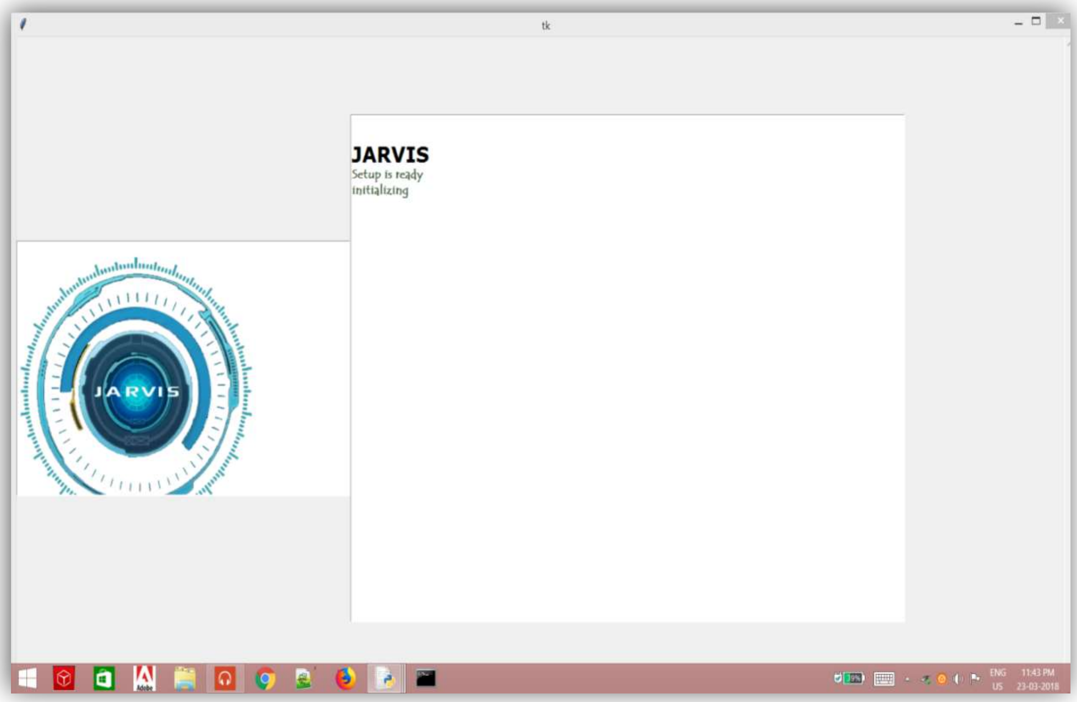
Standard builds of Python include an object-oriented interface to the Tcl/Tk widget set, called tkinter. This is probably the easiest to install (since it comes included with most binary distributions of Python) and use. For more info about Tk, including pointers to the source, see the Tcl/Tk home page. Tcl/Tk is fully portable to the Mac OS X, Windows, and UNIX platforms.

# CHAPTER 4

# RESULT AND DISCUSSION

## 4.1 Initialization

The below Fig. 4.1 shows that the software is being initialized a message is displayed on screen “setup is ready” and “initializing”. Software is preparing its self for the user.

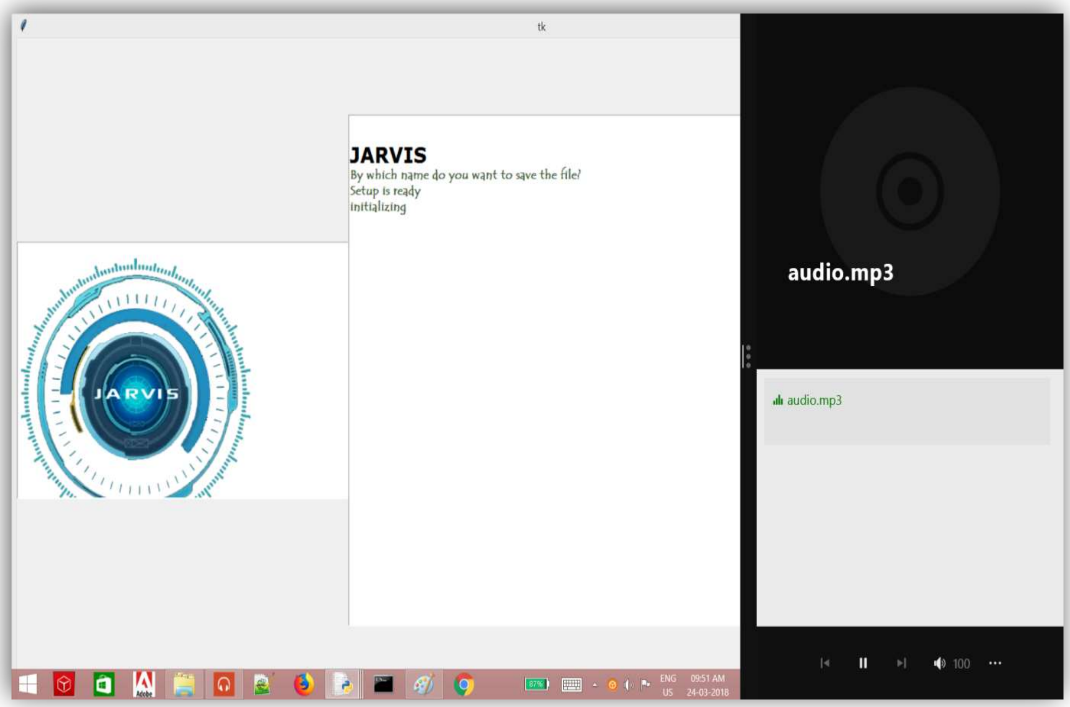


JARVIS
Setup is ready
initializing

Fig. 4.1 Initialization

## 4.2 Creating File

The below Fig. 4.2 shows the creation of new document file, it will ask the user that "by which name do you want to save the file?", when the user gives the file name it will create a blank document and ask the user to feed the data in document and save the file .

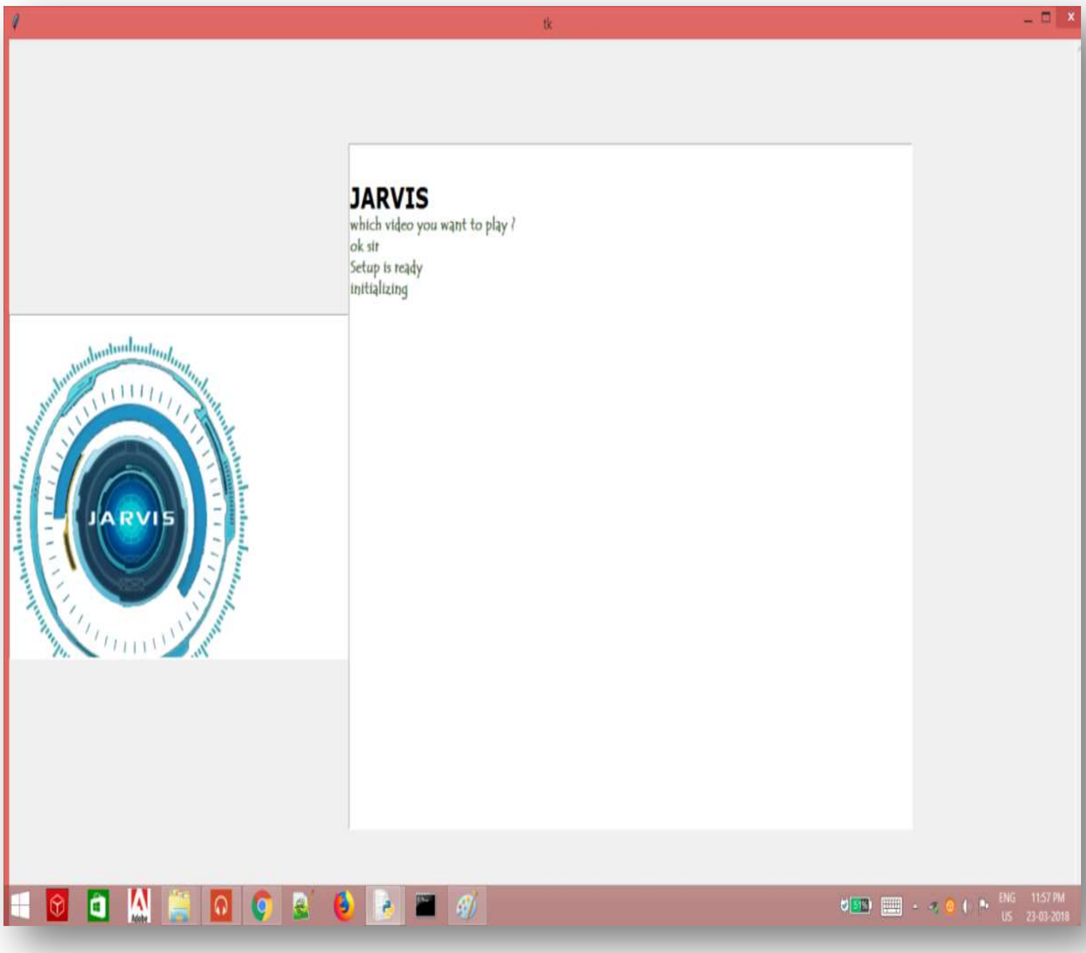


```
JARVIS
By which name do you want to save the file?
Setup is ready
initializing
```

Fig. 4.2 Creating file

## 4.3 Playing Video File

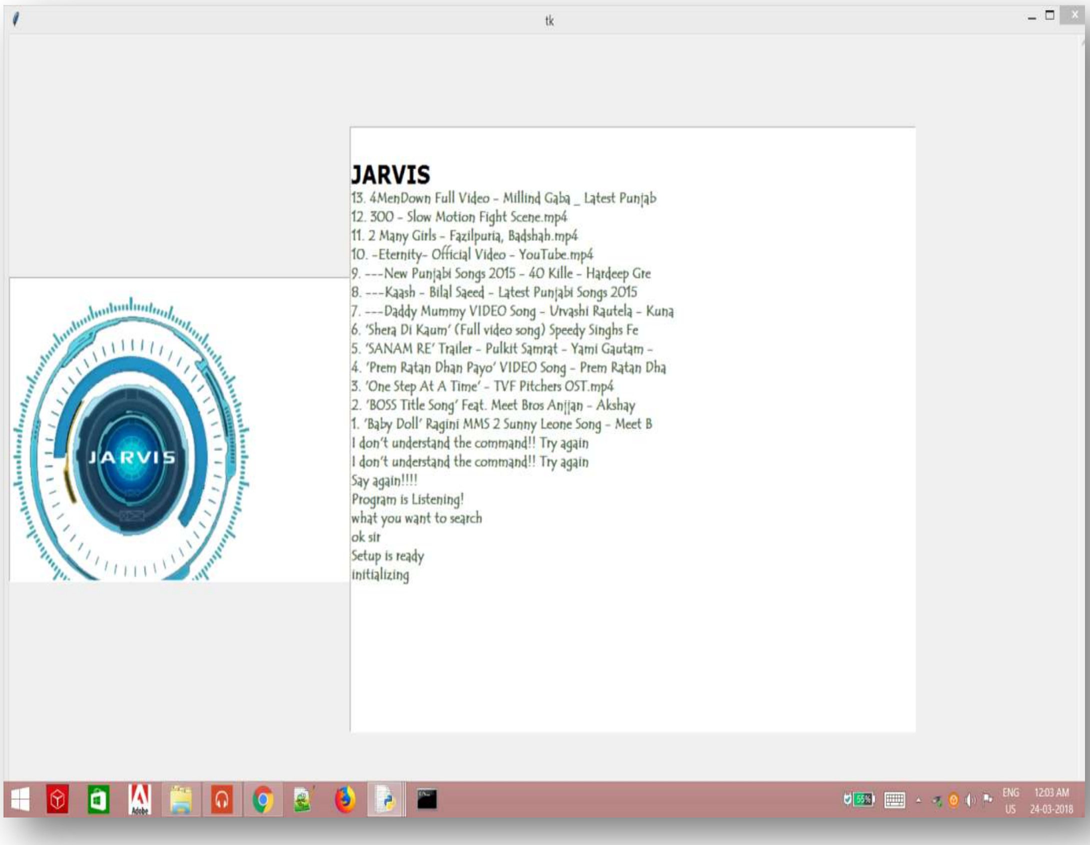
The below Fig 4.3 shows the playing of video by software; it will show the list of video in directory and ask the users "which song you want to play?"When user selects one video from that it will play that video.



```
JARVIS
which video you want to play ?
ok sir
Setup is ready
initializing
```

Fig 4.3 Playing video

The below Fig. 4.4 shows the list of videos after the play video command is detected the software will display all the video files stored in the system to the user. It will give the list of all videos stored in that folder.

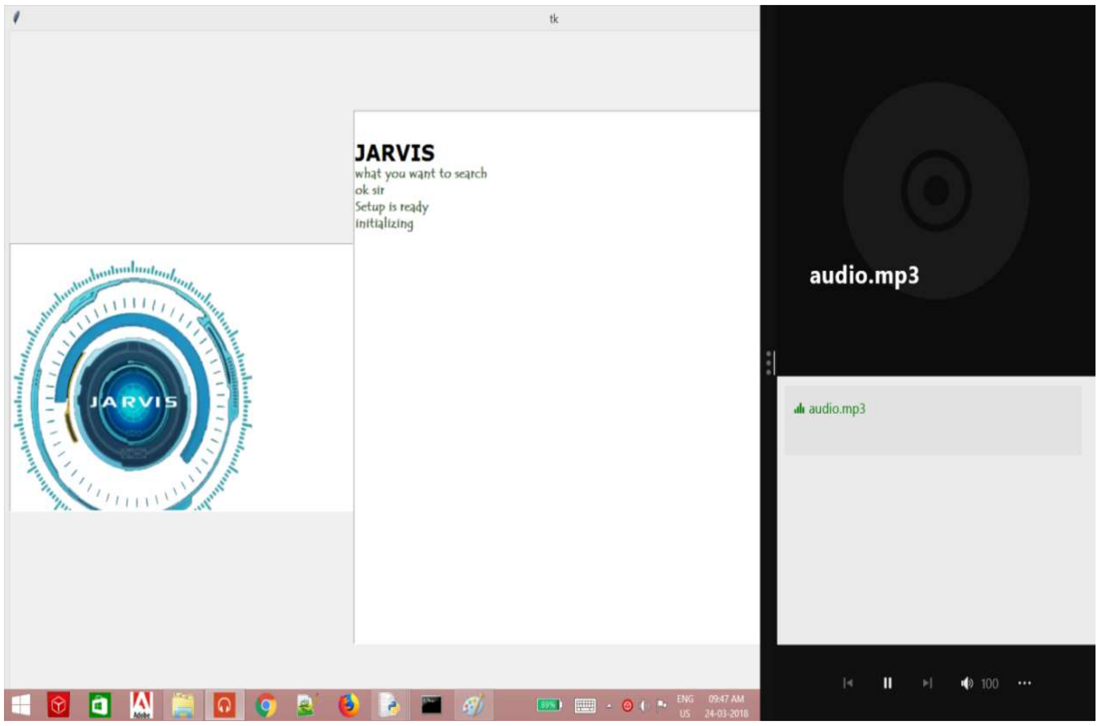


```
JARVIS
13. 4MenDown Full Video - Millind Gaba _ Latest Punjab
12. 300 - Slow Motion Fight Scene.mp4
11. 2 Many Girls - Fazilpuria, Badshah.mp4
10. -Eternity- Official Video - YouTube.mp4
9. ---New Punjabi Songs 2015 - 40 Kille - Hardeep Gre
8. ---Kaash - Bilal Saeed - Latest Punjabi Songs 2015
7. ---Daddy Mummy VIDEO Song - Urvashi Rautela - Kuna
6. 'Shera Di Kaum' (Full video song) Speedy Singhs Fe
5. 'SANAM RE' Trailer - Pulkit Samrat - Yami Gautam -
4. 'Prem Ratan Dhan Payo' VIDEO Song - Prem Ratan Dha
3. 'One Step At A Time' - TVF Pitchers OST.mp4
2. 'BOSS Title Song' Feat. Meet Bros Anjjan - Akshay
1. 'Baby Doll' Ragini MMS 2 Sunny Leone Song - Meet B
I don't understand the command!! Try again
I don't understand the command!! Try again
Say again!!!!
Program is Listening!
what you want to search
ok sir
Setup is ready
initializing
```

Fig. 4.4 List of videos

## 4.4 Surfing Browser

The below Fig. 4.5 shows the surfing of the web browser by the software. The software will ask the user that “what you want to search?” When the user gives the information then the desire result will be displayed by the software. It can also open specific videos in "YouTube" for the user on user’s request.



JARVIS
what you want to search
ok sir
Setup is ready
initializing

Fig. 4.5 Surfing browser

## 4.5 Initial Test and Result

The table below shows the testing results of the software

| S no. | Test | Input | Result | Discussion |
|---|---|---|---|---|
| 1. | Running notepad | Voice command | Notepad file opened successfully | While providing voice input to the software it recognized the spoken words in few attempts, this is due to noisy environment, variation in thevoice and multiple user factor |
| 2. | Running video player | Voice command | Video file opened | By providing voice input to the software for running system commands it worked fine and result in expectations, without repeating the commands twice or thrice. |
| 3. | Surfing browser | Voice command | Browser surf successful | We provided voice input to software and it successfully searched the desired item |

Table 4.1 Initial tests, results and discussions

# CHAPTER 5
# CONCLUSION

This project report details out the work of speech recognition completed by implementing this project. The project consists of both the gui and the development part implemented. Various functionalities implemented in the project are creation of file document, playing video, normal interaction with user, google search and Wikipedia. Various units in the project are implemented by using the google APIs for speech recognition and the overall text processing is done using NLP libraries in Python.

## 5.1 Limitation of the Study

This project will only work online with internet connection. This project can also be implemented using the offline libraries but its efficiency was very low. Program was facing problem in detecting the words. Pocket sphinx is the offline library provided by Python. So there is a problem in using this application offline.

Also the current application is facing some problem in detecting the words correctly but works fine mostly.

## 5.2 Future Scope of Work

This work can be taken into more detail and more work can be done on the project in order to bring modifications and additional features. The current software doesn't support a large vocabulary, the work will be done in order to accumulate more number of samples and increase the efficiency of the software. The current version of the software supports only few areas of the notepad but more areas can be covered and effort will be made in this regard. The data generated from the application can also be used to profile the user and run the analysis on user habits of using the technology.

## References

1. Steven Bird, Ewan Klien and Edward Looper. (2017). "Natural Language Processing", International Journal of Scientific & Engineering Research, 8.
2. Asst. Prof. Emad S. Othman. (2017). "Voice Controlled Personal Assistant Using Raspberry Pi", International Journal of Scientific & Engineering Research, 8.
3. Dr. Kotrappa Sirbi, Mr. Abhijit J.Patankar and Dr. Kshama V.Kulhalli. (2017). " Personal Assistant with Voice Recognition Intelligence", International Journal of Engineering Research and Technology,10.
4. Zed Shaw ,"Learn Python the Hard Way" , Third Edition, 2014.
5. www.tutorialpoint.com/python accessed on dated 01-jan-2018 at 6:00 pm referred syntax for list, tuples, loops and function calls.
6. https://docs.python.org/3/installing/index.html accessed on dated 28-dec-2017 at 7:00 pm referred libraries for python.
7. https://www.analyticsvidhya.com/blog/2017/01/ultimate-guide-to-understand-implement-natural-language-processing-codes-in-python/ accessed on dated 14-jan-2018 at 9:00 pm referred natural language processing.
8. http://www.nltk.org/ accessed on dated 23-jan-2018 at 8:00 pm referred natural language toolkit.
9. https://stackoverflow.com/questions/10158552/how-to-use-an-image-for-the-background-in-tkinter accessed on dated 1-mar-2018 at 8:00 pm referred gui for python.
10. https://www.youtube.com/watch?v=w36-U-ccajM&t=1s accessed on dated 2-Feb-2018 at 9:00 pm referred video for creating a file and then read it.