1. INTRODUCTION

In computer science and electrical engineering, speech recognition (SR) is the translation of spoken words into text. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT).

Some SR systems use "training" (also called "enrolment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker independent"[1] systems. Systems that use training are called "speaker dependent".

Speech recognition applications include voice user interfaces such as voice dialling (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domotic appliance control, search (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).

The term voice recognition[2][3][4] or speaker identification[5][6] refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process.

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the world-wide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems.

These speech industry players include Microsoft, Google, IBM, Baidu (China), Apple, Amazon, Nuance, IflyTek (China), many of which have publicized the core technology in their speech recognition systems being based on deep learning.

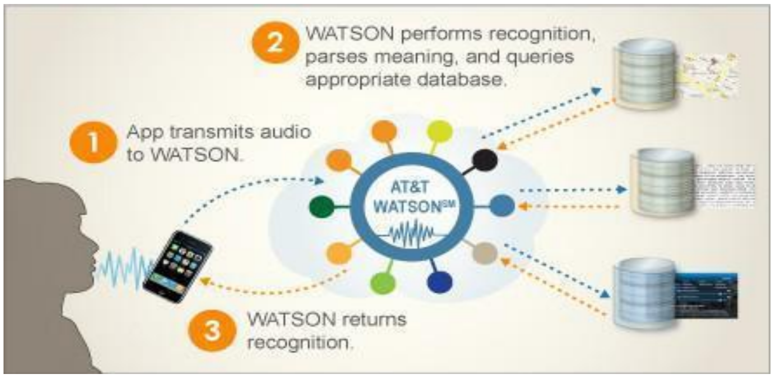

2 WATSON performs recognition, parses meaning, and queries appropriate database.
1 App transmits audio to WATSON.
AT&T WATSON℠
3 WATSON returns recognition.

Fig 1.WATSON block diagram

Now the rapid rise of powerful mobile devices is making voice interfaces even more useful and pervasive.

Jim Glass, a senior research scientist at MIT who has been working on speech interfaces since the 1980s, says today’s smart phones pack as much processing power as the laboratory machines he worked with in the ’90s. Smart phones also have high-bandwidth data connections to the cloud, where servers can do the heavy lifting involved with both voice recognition and understanding spoken queries. “The combination of more data and more computing power means you can do things today that you just couldn’t do before,” says Glass. “You can use more sophisticated statistical models.”

The most prominent example of a mobile voice interface is, of course, Siri, the voice-activated personal assistant that comes built into the latest iPhone. But voice functionality is built into Android, the Windows Phone platform, and most other mobile systems, as well as many apps. While these interfaces still have considerable limitations (see Social Intelligence), we are inching closer to machine interfaces we can actually talk to.

I have developed an **Artificial Intelligence Virtual Personal Assistance using Python**. This is desktop based software. It’s totally working on voice over commands. I will be teach numbers of commands to this software. I will be use API concepts. Like, Google API and Amazon API library files. This software fallow user’s commands.

An intelligent virtual assistant is an engineered entity residing in software that interfaces with humans in a human way. This technology incorporates elements of interactive voice response and other modern artificial intelligence projects to deliver full-fledged “virtual identities” that converse with users.[Web 1]

A **virtual assistant** or **intelligent personal 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). Some virtual assistants are able to interpret human speech and respond via synthesized voices. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal commands.[Web 2]

Artificial Intelligence personal assistants have become plentiful over the last few years. Applications such as Siri, Bixby, Ok Google and Cortana make mobile device users’ daily routines that much easier. You may be asking yourself how these functions. Well, the assistants receive external data (such as movement, voice, light, GPS readings, visually defined markers, etc.) via the hardware’s sensors for further processing - and take it from there to function accordingly.

I will making a Virtual Personal Assistance using Python. This software totally voice over commands. I will teach numbers of commands to this software. I will use API concepts. Like, Google API and Amazon API library files.

Not too long ago, building an AI assistant was a small component of developers’ capacities; however, nowadays, it is quite a realistic objective even for novice programmers. To create a simple personal AI assistant, one simply needs dedicated software and around an hour of working time. It would take much more time, though, to create something more advanced and conceptually innovative.

1.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Features of Python:

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

1.2 Python Modules:

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

1] The import Statement-

You can use any Python source file as a module by executing an import statement in some other Python source file. The **import** has the following syntax −

import module1[, module2[,... moduleN]

When the interpreter encounters an import statement, it imports the module if the module is present in the search path. A search path is a list of directories that the interpreter searches before importing a module.

2] The from...import Statement-

Python's **from** statement lets you import specific attributes from a module into the current namespace. The **from...import** has the following syntax −

from mod name import name1[, name2[, ... nameN]]

*3] The from...import * Statement*

It is also possible to import all the names from a module into the current namespace by using the following import statement −

from mod name import *

This provides an easy way to import all the items from a module into the current namespace; however, this statement should be used sparingly.

1.3 Installing Python:

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

If the binary code for your platform is not available, you need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

Here is a quick overview of installing Python on various platforms −

1] Unix and Linux Installation:

Here are the simple steps to install Python on Unix/Linux machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link to download zipped source code available for Unix/Linux.
- Download and extract files.
- Editing the *Modules/Setup* file if you want to customize some options.
- run ./configure script
- make
- make install

This installs Python at standard location */usr/local/bin* and its libraries at */usr/local/lib/pythonXX* where XX is the version of Python.

2] Windows Installation:

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer *python-XYZ.msi* file where XYZ is the version you need to install.
- To use this installer *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

2. Artificial Intelligence

AI was coined by John McCarthy, an American computer scientist, in 1956 at The Dartmouth Conference where the discipline was born. Today, it is an umbrella term that encompasses everything from robotic process automation to actual robotics. It has gained prominence recently due, in part, to big data, or the increase in speed, size and variety of data businesses are now collecting. AI can perform tasks such as identifying patterns in the data more efficiently than humans, enabling businesses to gain more insight out of their data.

2.1 Types of artificial intelligence

AI can be categorized in any number of ways, but here are two examples.

The first classifies AI systems as either weak AI or strong AI. Weak AI, also known as narrow AI, is an AI system that is designed and trained for a particular task. Virtual personal assistants, such as Apple's Siri, are a form of weak AI.

Strong AI, also known as artificial general intelligence, is an AI system with generalized human cognitive abilities so that when presented with an unfamiliar task, it has enough intelligence to find a solution. The Turing Test, developed by mathematician Alan Turing in 1950, is a method used to determine if a computer can actually think like a human, although the method is controversial.

The second example is from Arend Hintze, an assistant professor of integrative biology and computer science and engineering at Michigan State University. He categorizes AI into four types, from the kind of AI systems that exist today to sentient systems, which do not yet exist. His categories are as follows:

Type 1: Reactive machines.

An example is Deep Blue, the IBM chess program that beat Garry Kasparov in the 1990s. Deep Blue can identify pieces on the chess board and make predictions, but it has no memory and cannot use past experiences to inform future ones. It analyzes possible moves its own and its opponent -- and chooses the most strategic move. Deep Blue and Google's AlphaGO were designed for narrow purposes and cannot easily be applied to another situation.

Type 2: Limited memory.

These AI systems can use past experiences to inform future decisions. Some of the decision-making functions in autonomous vehicles have been designed this way. Observations used to inform actions happening in the not-so-distant future, such as a car that has changed lanes. These observations are not stored permanently.

Type 3: Theory of mind.

This is a psychology term. It refers to the understanding that others have their own beliefs, desires and intentions that impact the decisions they make. This kind of AI does not yet exist.

Type 4: Self-awareness. In this category,

AI systems have a sense of self, have consciousness. Machines with self-awareness understand their current state and can use the information to infer what others are feeling. This type of AI does not yet exist.

Examples of AI technology

Automation is the process of making a system or process function automatically. Robotic process automation, for example, can be programmed to perform high-volume, repeatable tasks normally performed by humans. RPA is different from IT automation in that it can adapt to changing circumstances.

Machine learning is the science of getting a computer to act without programming. Deep learning is a subset of machine learning that, in very simple terms, can be thought of as the automation of predictive analytics. There are three types of machine learning algorithms: supervised learning, in which data sets are labeled so that patterns can be detected and used to label new data sets; unsupervised learning, in which data sets aren't labeled and are sorted according to similarities or differences; and reinforcement learning, in which data sets aren't labeled but, after performing an action or several actions, the AI system is given feedback. Machine vision is the science of making computers see. Machine vision captures and analyzes visual information using a camera, analog-to-digital conversion and digital signal processing. It is often compared to human eyesight, but machine vision isn't bound by biology and can be programmed to see through walls, for example. It is used in a range of applications from signature identification to medical image analysis.

Computer vision, which is focused on machine-based image processing, is often conflated with machine vision.

Natural language processing (NLP) is the processing of human -- and not computer -- language by a computer program. One of the older and best known examples of NLP is spam detection, which looks at the subject line and the text of an email and decides if it's junk. Current approaches to NLP are based on machine learning. NLP tasks include text translation, sentiment analysis and speech recognition.

Pattern recognition is a branch of machine learning that focuses on identifying patterns in data. The term, today, is dated. Robotics is a field of engineering focused on the design and manufacturing of robots. Robots are often used to perform tasks that are difficult for humans to perform or perform consistently. They are used in assembly lines for car production or by NASA to move large objects in space. More recently, researchers are using machine learning to build robots that can interact in social settings.

2.2 AI Applications

AI in healthcare.

The biggest bets are on improving patient outcomes and reducing costs. Companies are applying machine learning to make better and faster diagnoses than humans. One of the best known healthcare technologies is IBM Watson. It understands natural language and is capable of responding to questions asked of it. The system mines patient data and other available data sources to form a hypothesis, which it then presents with a confidence scoring schema. Other AI applications include chatbots, a computer program used online to answer questions and assist customers, to help schedule follow-up appointments or aiding patients through the billing process, and virtual health assistants that provide basic medical feedback.

AI in business.

Robotic process automation is being applied to highly repetitive tasks normally performed by humans. Machine learning algorithms are being integrated into analytics and CRM platforms to uncover information on how to better serve customers. Chatbots have been incorporated into websites to provide immediate service to customers. Automation of job positions has also become a talking point among academics and IT consultancies such as Gartner and Forrester.

AI in education.

AI can automate grading, giving educators more time. AI can assess students and adapt to their needs, helping them work at their own pace. AI tutors can provide additional support to students, ensuring they stay on track. AI could change where and how students learn, perhaps even replacing some teachers.

AI in finance.

AI applied to personal finance applications, such as Mint or Turbo Tax, is upending financial institutions. Applications such as these could collect personal data and provide financial advice. Other programs, IBM Watson being one, have been applied to the process of buying a home. Today, software performs much of the trading on Wall Street.

AI in law.

The discovery process, sifting through of documents, in law is often overwhelming for humans. Automating this process is a better use of time and a more efficient process. Startups are also building question-and-answer computer assistants that can sift programmed-to-answer questions by examining the taxonomy and ontology associated with a database.

AI in manufacturing.

This is an area that has been at the forefront of incorporating robots into the workflow. Industrial robots used to perform single tasks and were separated from human workers, but as the technology advanced that changed.

3. OBJECTIVE

Providing information and entertainment, to otherwise solitary people, hence acts as a personal assistant.

People with disabilities can benefit from speech recognition programs. For individuals that are Deaf or Hard of Hearing, speech recognition software is used to automatically generate a closed-captioning of conversations such as discussions in conference rooms, classroom lectures, and/or religious services.[4]

Speech recognition is also very useful for people who have difficulty using their hands, ranging from mild repetitive stress injuries to involved disabilities that preclude using conventional computer input devices. In fact, people who used the keyboard a lot and developed RSI became an urgent early market for speech recognition.[6] Speech recognition is used in deaf telephony, such as voicemail to text, relay services, and captioned telephone. Individuals with learning disabilities who have problems with thought-to-paper communication (essentially they think of an idea but it is processed incorrectly causing it to end up differently on paper) can possibly benefit from the software but the technology is not bug proof.[7] Also the whole idea of speak to text can be hard for intellectually disabled person's due to the fact that it is rare that anyone tries to learn the technology to teach the person with the disability.[8]

Being bedridden can be very difficult for many patients to adjust to and it can also cause other health problems as well. It is important for family caregivers to know what to expect so that they can manage or avoid the health risks that bedridden patients are prone to. In this article we would like to offer some information about common health risks of the bedridden patient and some tips for family caregivers to follow in order to try and prevent those health risks.

Depression is also a very common health risk for those that are bedridden because they are unable to care for themselves and maintain the social life that they used to have. Many seniors begin to feel hopeless when they become bedridden but this can be prevented with proper care.

Family caregivers should make sure that they are caring for their loved one’s social and emotional needs as well as their physical needs. Many family caregivers focus only on the physical needs of their loved ones and forget that they have emotional and social needs as well. Family caregivers can help their loved ones by providing them with regular social activities and arranging times for friends and other family members to come over so that they will not feel

lonely and forgotten. Family caregivers can also remind their loved ones that being bedridden does not necessarily mean that they have to give up everything they used to enjoy.[10]

But since family members wont always be available at home, the above mentioned problems are still prevalent in these patients, hence our interactive system will provide them with entertainment (music, movies),and voice responses to general questions. Therefore it behaves as an electronic companion.

4. Proposed System

I have developed a desktop based virtual personal assistance software. When laptop or PC will be start then it is start on voice command of user. It will be show PC performance details and weather forecast details in the dashboard. Beautiful user interface will provide for every type of user of this software's.

5. SYSTEM REQUIREMENTS

The project needs both hardware and software components. The hardware components includes, the Raspberry Pi model B ,keyboard, mouse earphones, microphone with sound card, ethernet cable, HDMI screen and HDMI cable. Software components are Rasbian OS on SD card, Python compiler and the online resources Google speech API and Wolfram alpha .They are described in detail below

5.1 HARDWARE COMPONENTS

SOUND CARD WITH MICROPHONE

A sound card (also known as an audio card) is an internal computer expansion card that facilitates economical input and output of audio signals to and from a computer under control of computer programs. The term sound card is also applied to external audio interfaces that use software to generate sound, as opposed to using hardware inside the PC. Typical uses of sound cards include providing the audio component for multimedia applications such as music composition, editing video or audio, presentation, education and entertainment (games) and video projection.

Sound functionality can also be integrated onto the motherboard, using components similar to plug-in cards. The best plug-in cards, which use better and more expensive components, can achieve higher quality than integrated sound. The integrated sound system is often still referred to as a "sound card". Sound processing hardware is also present on modern video cards with HDMI to output sound along with the video using that connector; previously they used a SPDIF connection to the motherboard or sound card.

Headphone with mice

A microphone, colloquially nicknamed mic or mike (/ˈmark/),[1] is an acoustic-to-electric transducer or sensor that converts sound into an electrical signal. Electromagnetic transducers facilitate the conversion of acoustic signals into electrical signals.[2] Microphones are used in many applications such as telephones, hearing aids, public address systems for concert halls and public events, motion picture production, live and recorded audio engineering, two-way radios, megaphones, radio and television broadcasting, and in computers for recording voice, speech recognition, VoIP, and for non-acoustic purposes such as ultrasonic checking or knock sensors.

Most microphones today use electromagnetic induction (dynamic microphones), capacitance change (condenser microphones) or piezoelectricity (piezoelectric microphones) to produce an electrical signal from air pressure variations. Microphones typically need to be connected to a preamplifier before the signal can be amplified with an audio power amplifier and a speaker or recorded.

Fig 2.Sound Card Fig 3. Collar Mic

- Speakers
- 2GB RAM or
- Intel i3 and AMD Radeon Processor

5.2 Software Requirements

- Python 2.7

  Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.[3][4] Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java.[5][6] The language provides constructs intended to enable clear programs on both a small and large scale.[
- BOTO3

  Boto3 is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python, which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2. You can find the latest, most up to date, documentation at our doc site, including a list of services that are supported.
- Amazon AWS

  Amazon Web Services (**AWS**) offers a broad set of global compute, storage, database, analytics, application, and deployment **services** that help organizations move faster, lower IT costs, and scale applications. Learn more about **AWS** Products available in the Cloud.
- Windows 8 and later versions
- PyCharm IDE
- Google API
- Dialog Flow
- API .AI

6. IMPLEMENTATION

Instructions to setup proper environment for Project MAYA v.1

1) Install Python v. 2.7 from Folder “**Resources**” or download from :- https://www.python.org/downloads/release/python-2716/

2) Now Add python location into Environment variables Path :-

   a) Go to your python folder located in “C:\python27\” .
   b) Copy the same folder path i.e “C:\python27\” .
   c) Now right click on “**My computer**” or “**This PC**” icon in your desktop and then click on “**Properties**” menu.
   d) Now in Computer Properties screen, Click on “**Advanced System Settings**” on Top left side of the screen.
   e) Now click on “**Environment Variables**” button.
   f) Now up in **User Variables**, click on “**New**” to add an entry.
   g) Fill the Text boxes as :- [**variable name** – python , **variable value** – C:\python27\ ]
   h) Now down in **System Variables** find file name “**Path**” and click on “**Edit**” and then insert a semicolon at ending and then copy the python folder location (**;C:\python27\**)
   i) Now click “**Ok**” and once again “**Ok**” to save and exit .
   j) Done! You have added Python in System PATH.
3) Now Locate the Folder in ” **C:\python27\Scripts**” and follow the above guide (2) to add this path also to System PATH. After both Locations have been added to System PATH variables.
4) Press key “ **Windows + R** “ to open Run, here type “**cmd**” and click “**ok**” to open Command prompt.
5) Now go to the ‘**Resources’** folder path where “**get-pip.py**” is located, using CD command for e.g - “ cd D:/Resources/” and then Run “**python get-pip.py**” to install pip into your system.
6) Now As PIP is installed lets install some Python Modules for our Project :-
   a) Google-Search-API
   b) Boto3
   c) SpeechRecognition
   d) Google
   e) Playsound
   f) Pyaudio

   Command to install these modules :- “**pip install <module name>**” .

   Install one by one and for installing these modules you must be connected to internet.
7) Now as Python, pip and all python modules are installed and configured properly, you can double click on file “**MAYA.py**” to run the project.
8) Done!

7. ALGORITHMS

Both acoustic modeling and language modeling are important parts of modern statistically-based speech recognition algorithms. Hidden Markov models (HMMs) are widely used in many systems. Language modeling is also used in many other natural language processing applications such as document classification or statistical machine translation.

7.1 HMM (Hidden Markov models)

Modern general-purpose speech recognition systems are based on Hidden Markov Models. These are statistical models that output a sequence of symbols or quantities. HMMs are used in speech recognition because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal. In a short time-scale (e.g., 10 milliseconds), speech can be approximated as a stationary process. Speech can be thought of as a Markov model for many stochastic purposes.

Another reason why HMMs are popular is because they can be trained automatically and are simple and computationally feasible to use. In speech recognition, the hidden Markov model would output a sequence of n-dimensional real-valued vectors (with n being a small integer, such as 10), outputting one of these every 10 milliseconds. The vectors would consist of cepstral coefficients, which are obtained by taking a Fourier transform of a short time window of speech and de correlating the spectrum using a cosine transform, then taking the first (most significant) coefficients. The hidden Markov model will tend to have in each state a statistical distribution that is a mixture of diagonal covariance Gaussians, which will give a likelihood for each observed vector. Each word, or (for more general speech recognition systems), each phoneme, will have a different output distribution; a hidden Markov model for a sequence of words or phonemes is made by concatenating the individual trained hidden Markov models for the separate words and phonemes.

Described above are the core elements of the most common, HMM-based approach to speech recognition. Modern speech recognition systems use various combinations of a number of standard techniques in order to improve results over the basic approach described above. A typical large-vocabulary system would need context dependency for the phonemes (so phonemes with different left and right context have different realizations as HMM states); it would use cepstral normalization to normalize for different speaker and recording conditions; for further speaker normalization it might use vocal tract length normalization (VTLN) for male-female normalization and maximum likelihood linear regression (MLLR) for more general speaker adaptation. The features would have so-called delta and delta-delta coefficients to capture speech dynamics and in addition might use heteroscedastic linear discriminant analysis (HLDA); or might skip the delta and delta-delta coefficients and use splicing and an LDA-based projection followed perhaps by heteroscedastic linear discriminant analysis or a global semi-tied co variance transform (also known as maximum likelihood linear transform, or MLLT). Many systems use so-called discriminative training techniques that dispense with a purely statistical approach to HMM parameter estimation and instead optimize some classification-related measure of the training data. Examples are maximum mutual information (MMI), minimum classification error (MCE) and minimum phone error (MPE).

Decoding of the speech (the term for what happens when the system is presented with a new utterance and must compute the most likely source sentence) would probably use the Viterbi algorithm to find the best path, and here there is a choice between dynamically creating a combination hidden Markov model, which includes both the acoustic and language model information, and combining it statically beforehand (the finite state transducer, or FST, approach).

A possible improvement to decoding is to keep a set of good candidates instead of just keeping the best candidate, and to use a better scoring function (re scoring) to rate these good candidates so that we may pick the best one according to this refined score. The set of candidates can be kept either as a list (the N-best list approach) or as a subset of the models (a lattice). Re scoring is usually done by trying to minimize the Bayes risk[7] (or an approximation thereof): Instead of taking the source sentence with maximal probability, we try to take the sentence that minimizes the expectancy of a given loss function with regards to all possible transcriptions (i.e., we take the sentence that minimizes the average distance to other possible sentences weighted by their estimated probability).

The loss function is usually the Levenshtein distance, though it can be different distances for specific tasks; the set of possible transcriptions is, of course, pruned to maintain tractability. Efficient algorithms have been devised to re score lattices represented as weighted finite state transducers with edit distances represented themselves as a finite state transducer verifying certain assumptions.[8]

7.2 DEEP NEURAL NETWORK

A deep neural network (DNN) is an artificial neural network with multiple hidden layers of units between the input and output layers.[6] Similar to shallow neural networks, DNNs can model complex non-linear relationships. DNN architectures generate compositional models, where extra layers enable composition of features from lower layers, giving a huge learning capacity and thus the potential of modeling complex patterns of speech data.[6] The DNN is the most popular type of deep learning architectures successfully used as an acoustic model for speech recognition since 2010.

The success of DNNs in large vocabulary speech recognition occurred in 2010 by industrial researchers, in collaboration with academic researchers, where large output layers of the DNN based on context dependent HMM states constructed by decision trees were adopted.[7][8] [9]

One fundamental principle of deep learning is to do away with hand-crafted feature engineering and to use raw features. This principle was first explored successfully in the architecture of deep autoencoder on the "raw" spectrogram or linear filter-bank features,[2] showing its superiority over the Mel-Cepstral features which contain a few stages of fixed transformation from spectrograms. The true "raw" features of speech, waveforms, have more recently been shown to produce excellent larger-scale speech recognition results.[3]

Since the initial successful debut of DNNs for speech recognition around 2009-2011, there have been huge new progresses made. This progress (as well as future directions) has been summarized into the following eight major areas:[8]

Scaling up/out and speedup DNN training and decoding;
S1equence discriminative training of DNNs;

Feature processing by deep models with solid understanding of the underlying mechanisms;

Adaptation of DNNs and of related deep models; Multi-task and
transfer learning by DNNs and related deep models;

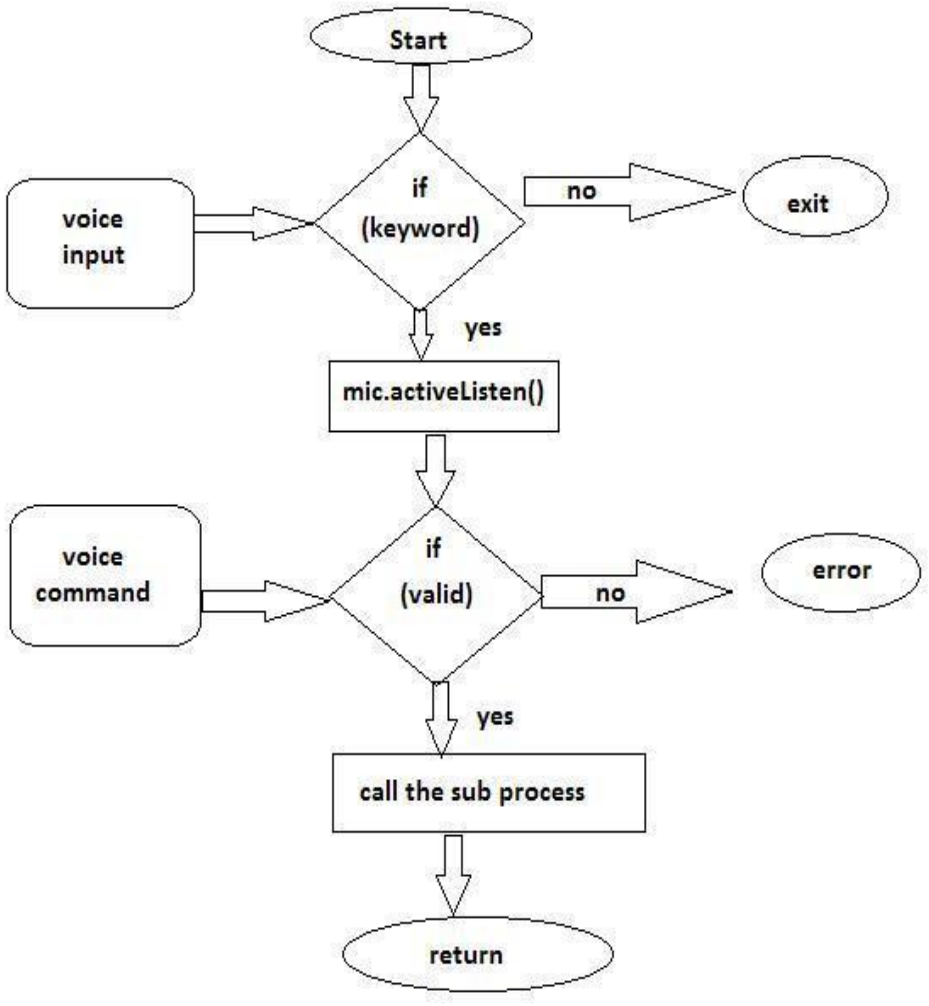
Convolution neural networks and how to design them to best exploit
domain knowledge of speech;

Recurrent neural network and its rich LSTM variants;

Other types of deep models including tensor-based models and integrated deep generative/discriminative models.

Large-scale automatic speech recognition is the first and the most convincing successful case of deep learning in the recent history, embraced by both industry and academic across the board. Between 2010 and 2014, the two major conferences on signal processing and speech recognition, IEEE-ICASSP and Interspeech, have seen near exponential growth in the numbers of accepted papers in their respective annual conference papers on the topic of deep learning for speech recognition. More importantly, all major commercial speech recognition systems (e.g., Microsoft Cortana, Xbox, Skype Translator, Google Now, Apple Siri, Baidu and iFlyTek voice search, and a range of Nuance speech products, etc.) nowadays are based on deep learning methods.[5]

8. FLOW CHART OF PROGRAM



9. BLOCK DIAGRAM

Here we are using CMU Sphinx with jasper-client brain which implements deep learning algorithm.

Python modules are written for various functions. First the keyword which is configured is said, we will hear a high beep, which means jasper is listening.

Now the command is given ,which is decoded and searched by the pocketshinx dictionary by HMM computation.

Match is found to mentioned words in modules and the appropriate function is executed. Which can be playing a song or video or reading a book or changing TV channel or playing a quiz game.

The song and video database can have any regional language songs as well. The output of the system is then heard through the speakers or earphones.

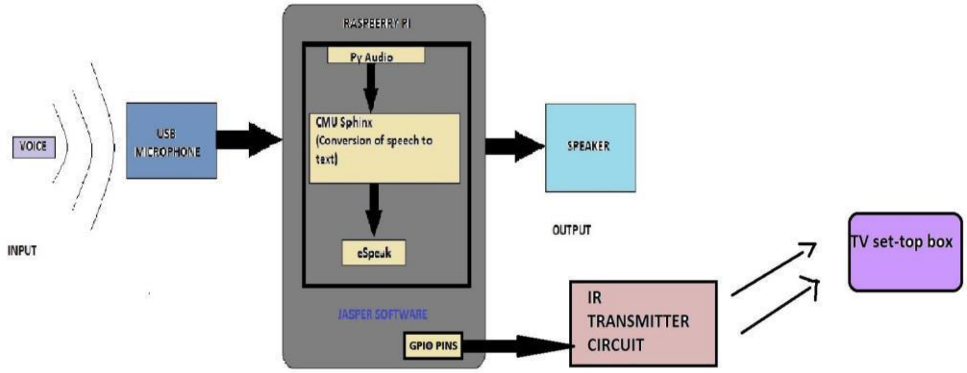


Fig 4.Block Diagram of System

10. COMMANDS OF MAYA

MAYA Artificial Intellgience Commands List

1) Hello Maya.
2) Hi Maya
3) Hey Maya
4) Open Program files
5) Open Windows
6) Open Facebook
7) Open Twitter
8) Open YouTube
9) Open Google
10) Launch Google
11) Launch YouTube
12) Search Video
13) Launch Twitter
14) Launch Facebook
15) Launch Windows
16) Launch Program Files
17) Thank you Maya
18) Shutdown the Pc
19) Cancel Shutdown
20) Restart the PC
21) Reboot the PC
22) Lock the Screen
23) Open Notepad
24) Open Word
25) Open Excel
26) Open PowerPoint
27) Launch word
28) Launch excel
29) Launch PowerPoint
30) Launch Notepad
31) Question statements like (what, who, when , why, which, where, how)

    For e.g – a) what is todays whether like?
    b) Who is narendra modi?
    c) when is avengers movie releasing?
    d) why is Earth Circle?
    e) how is Fevicol Made?
    f) how is time ?

31) Tell me about yourself?

32) Play Music

33) Locate <city or place name> for e.g – locate pune

34) Good bye Maya

35) Bye Maya

11. FURTHER ENHANCEMENTS

11.1. RECOGNITION WITHOUT INTERNET ACCESS

We are well aware that there is no availability of internet access throughout our country. Currently, India is nowhere near meeting the target for a service which is considered almost a basic necessity in many developed countries.

In such cases this project may not function, therefore we have enhancing this project to work even without internet using recognition toolkits such as Python library.

11.2. HOME AUTOMATION

With the right level of ingenuity, the sky's the limit on things you can automate in your home, but here are a few basic categories of tasks that you can pursue:

Automate your lights to turn on and of on a schedule, remotely, or when certain conditions are triggered.

Set your air conditioner to keep the house temperate when you're home and save energy while you're away.

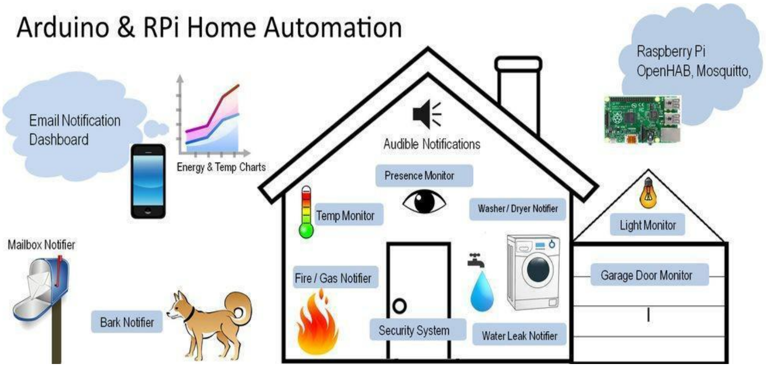


Fig.5 Home Automation Possibilities

12. APPLICATIONS

Usage in education and daily life

For language learning, speech recognition can be useful for learning a second language. It can teach proper pronunciation, in addition to helping a person develop fluency with their speaking skills.[6]

Students who are blind (see Blindness and education) or have very low vision can benefit from using the technology to convey words and then hear the computer recite them, as well as use a computer by commanding with their voice, instead of having to look at the screen and keyboard. [6]

Aerospace (e.g. space exploration, spacecraft, etc.) NASA’s Mars Polar Lander used

speech recognition from technology Sensory, Inc. in the Mars Microphone on the Lander[7]

Automatic subtitling with speech recognition[7]

Automatic translation

Court reporting (Realtime Speech Writing)

Telephony and other domains

ASR in the field of telephony is now commonplace and in the field of computer gaming and simulation is becoming more widespread. Despite the high level of integration with word processing in general personal computing. However, ASR in the field of document production has not seen the expected[by whom?] increases in use.

predefined or custom speech commands. Leading software vendors in this field are: Google, Microsoft Corporation (Microsoft Voice Command), Digital Syphon (Sonic Extractor), LumenVox, Nuance Communications (Nuance Voice Control), VoiceBox

Technology, Speech Technology Center, Vito Technologies (VITO Voice2Go), Speereo Software (Speereo Voice Translator), Verbyx VRX and SVOX.

In Car systems

Typically a manual control input, for example by means of a finger control on the steering-wheel, enables the speech recognition system and this is signalled to the driver by an audio prompt. Following the audio prompt, the system has a "listening window" during which it may accept a speech input for recognition.

Simple voice commands may be used to initiate phone calls, select radio stations or play music from a compatible smartphone, MP3 player or music-loaded flash drive. Voice recognition capabilities vary between car make and model. Some of the most recent car models offer natural-language speech recognition in place of a fixed set of commands. allowing the driver to use full sentences and common phrases. With such systems there is, therefore, no need for the user to memorize a set of fixed command words.

Fig 6.Car Automation

Helicopters

The problems of achieving high recognition accuracy under stress and noise pertain strongly to the helicopter environment as well as to the jet fighter environment. The acoustic noise problem is actually more severe in the helicopter environment, not only because of the high noise levels but also because the helicopter pilot, in general, does not wear a facemask, which would reduce acoustic noise in the microphone. Substantial test and evaluation programs have been carried out in the past decade in speech recognition systems applications in helicopters, notably by the U.S. Army Avionics Research and Development Activity (AVRADA) and by the Royal Aerospace Establishment (RAE) in the UK. Work in France has included speech recognition in the Puma helicopter. There has also been much useful work in Canada. Results have been encouraging, and voice applications have included: control of communication radios, setting of navigation systems, and control of an automated target handover system.

As in fighter applications, the overriding issue for voice in helicopters is the impact on pilot effectiveness. Encouraging results are reported for the AVRADA tests, although these represent only a feasibility demonstration in a test environment. Much remains to be done both in speech recognition and in overall speech technology in order to consistently achieve performance improvements in operational settings.

High-performance fighter aircraft

Substantial efforts have been devoted in the last decade to the test and evaluation of speech recognition in fighter aircraft. Of particular note is the U.S. program in speech recognition for the Advanced Fighter Technology Integration (AFTI)/F-16 aircraft (F-16 VISTA), and a program in France installing speech recognition systems on Mirage aircraft, and also programs in the UK dealing with a variety of aircraft platforms. In these programs, speech recognizers have been operated successfully in fighter aircraft, with applications including: setting radio frequencies, commanding an autopilot system, setting steer-point coordinates and weapons release parameters, and controlling flight display.

13. LITERATURE REVIEW

- Suma Swamy1 and K.V Ramakrishnan[1],in this paper states that the present batch learning strategy, based on the preliminary collection of the entire set of labeled data from experts, allowed an accurate experimental evaluation but needs to be evolved to include new operational requirements.

- Veton Këpuska and Gamal Bohouta [2] ,in this article, One of the goals of Artificial intelligence (AI) is the realization of natural dialogue between humans and machines. in recent years, the dialogue systems, also known as interactive conversational systems are the fastest growing area in AI. Many companies have used the dialogue systems technology to establish various kinds of Virtual Personal Assistants(VPAs) based on their applications and areas, such as Microsoft's Cortana, Apple's Siri, Amazon Alexa, Google Assistant, and Facebook's M. However, in this proposal, we have used the multi-modal dialogue systems which process two or more combined user input modes, such as speech, image, video, touch, manual gestures, gaze, and head and body movement in order to design the Next Generation of VPAs model.

- John Makhoul BBN Technologies [3].This survey of Bolt Beranek and Newman's (BBN) speech processing activities covers a period that began around 1971. Areas of importance—technical as well as historical—include speech recognition and understanding, speech coding, speaker recognition, and speech modification. A number of today's best-regarded techniques in speech and language processing stem from BBN's early work.

14. REFERENCES

14.1 Websites

- https://www.techopedia.com/definition/31383/intelligent-virtual-assistant
- https://en.wikipedia.org/wiki/Virtual_assistant
- https://dialogflow.com/
- https://www.researchgate.net/publication/264001644_Virtual_Personal_Assistant
- https://www.brainasoft.com/braina/
- Amazon. Amazon Lex is a service for building conversational interfaces. https://aws.amazon.com.
- Microsoft. Cortana Intelligence. https://azure.microsoft.com.
- Google Assistant https://ok-google.io/

14.2 BOOK

- Learning Python : Object Oriented Programming by Mark Lutz

14.3 RESEARCH PAPERS

[1]D. Yu and L. Deng"*Automatic Speech Recognition: A Deep Learning Approach*" (Publisher: Springer) published near the end of 2014,

[2]Claudio Becchetti, Klucio Prina Ricotti.*"Speech Recognition: Theory and C++ Implementation "* 2008 edition

[3]Reynolds, Douglas; Rose, Richard (January 1995). "Robust text-independent speaker identification using Gaussian mixture speaker models" (PDF). IEEE Transactions on Speech and Audio Processing (IEEE) 3 (1): 72–83. doi:10.1109/89.365379. ISSN 1063-6676. OCLC 26108901. Retrieved 21 February 2014.

[4]Waibel, Hanazawa, Hinton, Shikano, Lang. (1989) "*Phoneme recognition using time-delay neural networks. IEEE Transactions on Acoustics, Speech and Signal Processing.*

[5]Microsoft Research. "*Speaker Identification (WhisperID)*". Microsoft. Retrieved 21 February 2014.

[6]]'*Low Cost Home Automation Using Offline Speech Recognition*', International Journal of Signal Processing Systems, vol. 2, no. 2, pp. 96-101, 2014.

[7]Juang, B. H.; Rabiner, Lawrence R. "*Automatic speech recognition–a brief history of the technology development*" (PDF). p. 6. Retrieved 17 January 2015.

[8] Deng, L.; Li, Xiao (2013). "Machine Learning Paradigms for Speech Recognition: An Overview". IEEE Transactions on Audio, Speech, and Language Processing.

[9]P. V. Hajar and A. Andurkar, *"Facial Recognition and Speech Recognition using Raspberry Pi*', International Journal of Advanced Research in Computer and CommunicationReview Paper on System for Voice and F Engineering, vol. 4, no. 4, pp. 232-234, 2015.

[10]Common Health Risks of the Bedridden Patient Posted on October 24, 2013 by Carefect Blog Team.

1. Suma Swamy1 and K.V Ramakrishnan in "An efficient speech recognition system" Computer Science & Engineering: An International Journal (CSEIJ), Vol. 3, No. 4, August 2013

2. Veton Këpuska and Gamal Bohouta [2] in"Next-Generation of Virtual Personal

Assistants"

3. John Makhoul BBN Technologies[3] in "Speech Processing at BBN"

4. Ronald M. Baecker, "Readings in human-computer interaction: toward the year

2000", 1995.

5. Melanie Pinola, “Speech Recognition Through the Decades: How We Ended Up With Siri”, PCWorld.

6. Ganesh Tiwari, “Text Prompted Remote Speaker Authentication : Joint Speech and Speaker Recognition/Verification System”.

7. Dr.Ravi Sankar, Tanmoy Islam, Srikanth Mangayyagari, “Robust Speech/Speaker Recognition Systems”.