Voice Assistant

Using machine learning and python

Submitted to:

Ass. pro. Randeep Kaur

Guided By:
Er. Shweta Agarwal *shweta*

Submitted By:

ADITYA

ARORA(18BCS6656)

SUMIT

KUMAR(18BCS6654)

BINIT KUMAR

(18BCS6639) SHIVAM

(18BCS3054)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Chandigarh University, Gharuan

Introduction

Who doesn't want to have the luxury to own an assistant who always listens for your call, anticipates your every need, and takes action when necessary? That luxury is now available thanks to artificial intelligence-based voice assistants.

Voice assistants come in somewhat small packages and can perform a variety of actions after hearing your command. They can turn on lights, answer questions, play music, place online orders and do all kinds of AI-based stuff.

Voice assistants are not to be confused with virtual assistants, which are people who work remotely and can, therefore, handle all kinds of tasks. Rather, voice assistants are technology based. As voice assistants become more robust, their utility in both the personal and business realms will grow as well.

This project includes an implementation of an intelligent voice recognition assistant for Android where functionality on current existing applications on other platforms is compared. Until this day, there has not been any good alternative for Android, so this project aims to implement a voice assistant for the Android platform while describing the difficulties and challenges in this task.

- **Technology used**

1.) Python 2.7, Spyder IDE, MacOS Mojave(version 10.14)

2.) Install all these python libraries :

3.) pip install SpeechRecognition

4.) pip install beautifulsoup4

5.) pip install vlc

6.) pip install youtube-dl

7.) pip install pyowm

8.) pip install wikipedia

- **Field of project**

1.) Machine learning

2.) Deep learning

3.) python

Feasibility Study

Why we need voice assistant :
A voice assistant or intelligent personal assistant is a software agent that can perform tasks or services for an individual based on verbal commands i.e. by interpreting 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, open or close any application etc with verbal commands.

Having said that, how cool it would be to build a simple voice-based desktop/laptop assistant that has the capability to:-

1. Open the subreddit in the browser.

2. Open any website in the browser.

3. Send an email to your contacts.

4. Launch any system application.

5. Tells you the current weather and temperature of almost any city

6. Tells you the current time.

7. Greetings

8. Play you a song on VLC media player(of course you need to have VLC media player installed in your laptop/desktop)

9. Change desktop wallpaper.

10. Tells you latest news feeds.

11. Tells you about almost anything you ask.

Risk Feasibility

1. Business Impact Risk - Number of customers who will use this project and the consistency of their needs relative to the product will increase the Business.

2. Customer-related risks
This project is a general type of project (not designed just for a single college). Before implementing the system in an educational institute, there will be some modifications areas.

Methodology/ Planning of work

There are three ways to build a virtual assistant:

Simple method

Using this method, you can integrate trusted technologies from famous companies — e.g. Apple, Microsoft, Amazon, etc. In this method, you can use ready-made solutions like an API from Siri, Cortana, or Google Assistant. It's a simple approach.

Advanced methods

This solution is based on open-source libraries and systems. Here are some cross-platform tools for developing assistant apps.

Mellissa- It's a virtual assistant for OS X, Windows, and Linux systems. She currently uses Google Chrome's speech-to-text engine, OS X's say a command, Linux's e-speak command.

DialogFlow- It incorporates Google's machine learning expertise and products such as Google Cloud Speech-to-Text and runs on Google Cloud Platform, letting you scale to hundreds of millions of users.

Wit.ai- It's so similar to Dialogflow. It requires two components to be set up: intents (users request) and entities (intent's characteristics). Wit.ai has a good long list of intents, so you don't have to build it yourself. And it is completely free for private and public versions.

Develop from Scratch

This requires some expertise in Deep Learning, Natural Language Processing, and Conversational systems. It requires knowledge of programming languages.

In particular, such languages as Lisp, Java, Prolog, and Python are used for the creation of AI-based apps. Python is used as a base for the most renowned AI-based software because of its flexibility, simplicity, and longstanding reputation.

It requires knowledge about different python libraries like NumPy, speech-recognition, gTTS, pygame, selenium, web browser, Wikipedia, etc.

In this project, we will develop our project from scratch using deep learning and artificial intelligence through python. We will try to develop a virtual assistant that and automate tasks such as search videos in YouTube and play them, send emails, open websites, search materials in Wikipedia and read them, inform weather forecast in your country, greetings and more.

Module & Team Member wise Distribution of work

There will be a total of 8 major tasks in this project and the work is equally distributed between both team members. The modules are as explained below:

- **Listening to the user's command**– Defining a function to record the audio of the user and printing it on the screen also.

- **Get a response from the assistant** – Defining a function to make the assistant respond by speaking back to the user.

- **Greeting the user** – Responding with greetings like hello, hi, etc. when the user greets the assistant.

- **The main function** – This function contains trigger words that will make the assistant perform the required task. The tasks are:

- **Searching the web –** If there is a search word in the user's statement the assistant will ask back what the user wants to search for and then opens the web browser and make the required query.

- **Searching location-** – If there is a search location word in the user's statement the assistant will ask back what is the location the user wants to search for and then opens the google maps to show the location.

- **Sending email** – If the user gives a command to send an email the assistant will ask for the subject and the body and then send the email to the required recipient.

- **Opening YouTube** – If the user asks for YouTube videos the assistant opens the YouTube and then searches the required video.

- **Telling about the weather**- If the user asks about the weather the assistant tells the weather condition.

1.)Aditya arora : 18BCS6656 : leader : coding part.
2.)Sumit kumar : 18BCS6654 : software handling part .
3.)Binit kumar : 18CS6639 : handling voice part of project.
4.)Shivam : 18BCS3054 : handling documentation of project.

Innovations in Project

Voice assistants began to emerge in 2011 with the introduction of Siri by Apple, no one could have predicted that this novelty would become a driver for tech innovation. The main driver of the shift towards voice user interfaces is the changing user demands. There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers.

Though VPA's are moderately efficient, it is not very helpful and is not used by the user due to its high amount of error. Though the error percentage of the upcoming VPAs is around 5 percent, it still is not quite up to the mark to where it becomes a basic part of the user's life. Thus, the project aims to build a VPA with speech recognition which has a very minimal error percentage.

We will also try to achieve a streamlined conversation, the assistants will no longer require the use of repeated "wake" words. Generally, assistants are dependent on a wake word to initiate a new line of conversation.

Software and Hardware Requirements

Software Required:

Python (version 3.8) - Python is a programming language that lets you work more quickly and integrate your systems more effectively. It includes libraries necessary for developing a machine learning project.

PyCharm - PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python Language. It is developed by the Czech company JetBrains. It provides code analysis, a graphic debugger, an integrated unit tester, integration with version control systems (VCSes). It is cross-platform with Windows, Linux, and macOS.

Microsoft Word- Word is a word processor published by Microsoft. It is one of the office productivity applications included in the Microsoft Office suite. Microsoft Word allows you to create professional-quality documents, reports, letters, and resumes. Unlike a plain text editor, Microsoft Word has features including spell check, grammar check, text and font formatting, and more. In this project, it will be used to do most of the Documentation Work.

Hardware Required:

Windows Laptop with Intel(R) core (TM) i5-8250U CPU @ 1.60GHz 1.80GHz Installed RAM 8 GB 64-bit operating system, x64 based processor

Mic – To command the virtual assistant

Speaker – To hear the response from a virtual assistant

Router – To have a fast internet connection so that functions like searching and emailing can be done with high speed.

Bibliography

https://www.google.com/
https://www.youtube.com/
en.wikipedia.org
www.researchgate.net
www.jetbrains.com
www.python.org
clearbridgemobile.com

towardsdatascience.com
softmedialab.com
avirtual.co.uk
www.geeksforgeeks.org