



VOICE ASSISTANTS

Submitted by

Candidate's Name :- DHAIRYA VARDHAN & DEEPANSHU VARUN

Roll No. – (2K19/EC/053) & (2K19/EC/052) resp.

Submitted to

PROF. ROHIT KUMAR

**Department of Electronics & Communication
Engineering Delhi Technological University, Delhi**

Delhi-110042 MAY

2021



Electronics & Communication Engg. Deptt.

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Bawana Road-Delhi-42

CANDIDATE'S DECLARATION

We, DEEPANSHU VARUN and DHAIRYA VARDHAN, Roll No(s). 2K19/EC/052 and 2K19/EC/053 student(s) of B. Tech. ELECTRONICS AND COMMUNICATION, hereby declare that the project Dissertation titled " **Project to develop an application that provides an intelligent voice assistant**" which is submitted by us to the Department of electronics and communication, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree Bachelor of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

DEEPANSHU VARUN AND DHAIRYA VARDHAN

DATE:28.05.2021

CERTIFICATE

I hereby certify that the Project Dissertation titled " **Project to develop an application that provides an intelligent voice assistant**" which is submitted by DEEPANSHU VARUN and DHAIRYA VARDHAN, Roll No(s). 2K19/EC/052 and 2K19/EC/053 of BTech. Electronic and communication, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of Bachelor of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

PROF. ROHIT KUMAR

Date: 28.05.2021

ACKNOWLEDGEMENT

We would like to express my special thanks of gratitude to my teacher Mr. ROHIT KUMAR, who gave us the golden opportunity to do this wonderful project of COMMUNICATION SYSTEM . Who also helped us in completing our project! we came to know about so many new things we are really thankful to him. Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

ABSTRACT

Artificial intelligence technologies are starting to be actively utilized in human life, this is often facilitated by the appearance and wide dissemination of Internet of Thing (IOT). Autonomous devices are getting smarter in their thanks to interact with both a person's and themselves. New capacities cause creation of varied systems for integration of smart things into Social Networks of the web of Things. One of the relevant trends in AI is that the technology of recognizing the tongue of a person's . New insights in this topic can lead to new means of natural human-machine interaction, in which the machine would learn how to understand human's language, adjusting and interacting in it. One of such tools is voice assistant, which can be integrated into many other intelligent systems. In this report, the principles of the functioning of the voice assistant are described, its main shortcomings and limitations are given. The method to creating a local voice assistant without using cloud services is as described, which allows to significantly expand the applicability of such devices in the future.

INDEX

- AIM/ OBJECTIVE OF THE PROJECT
- OVERVIEW
- INTRODUCTION
- THEORY
- METHODOLOGY
- CODING PART
- CONCLUSION
- REFERENCES

OBJECTIVE

The purpose of the project is to develop an application that provides an intelligent voice assistant with the functionalities as calling services, message, music play service, checking weather, searching engine (Google, Wikipedia) etc.

INTRODUCTION

Today the development of artificial intelligence (AI) systems that are able to organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behavior and trying to become his personalized assistant. The work on creating and improving such personalized assistants has been going on for a long time. These systems are constantly improving and improving, go beyond personal computers and have already firmly established themselves in various mobile devices and gadgets. One of the most popular voice assistants are Siri, from Apple, Amazon Echo, which responds to the name of Alex from Amazon, Cortana from Microsoft, Google Assistant from Google, and the recently appeared intelligent assistant under the name "AIVA".

THEORY

As we know Python is a suitable language for script writers and developers. Let's write a script for Voice Assistant using Python. The query for the assistant can be manipulated as per the user's need. Speech recognition is the process of converting audio into text. This is commonly used in voice assistants like Alexa, Siri, etc. Python provides an API called

SpeechRecognition to allow us to convert audio into text for further processing. In this article, we will look at converting large or long audio files into text using the SpeechRecognition API in python.

METHODOLOGY

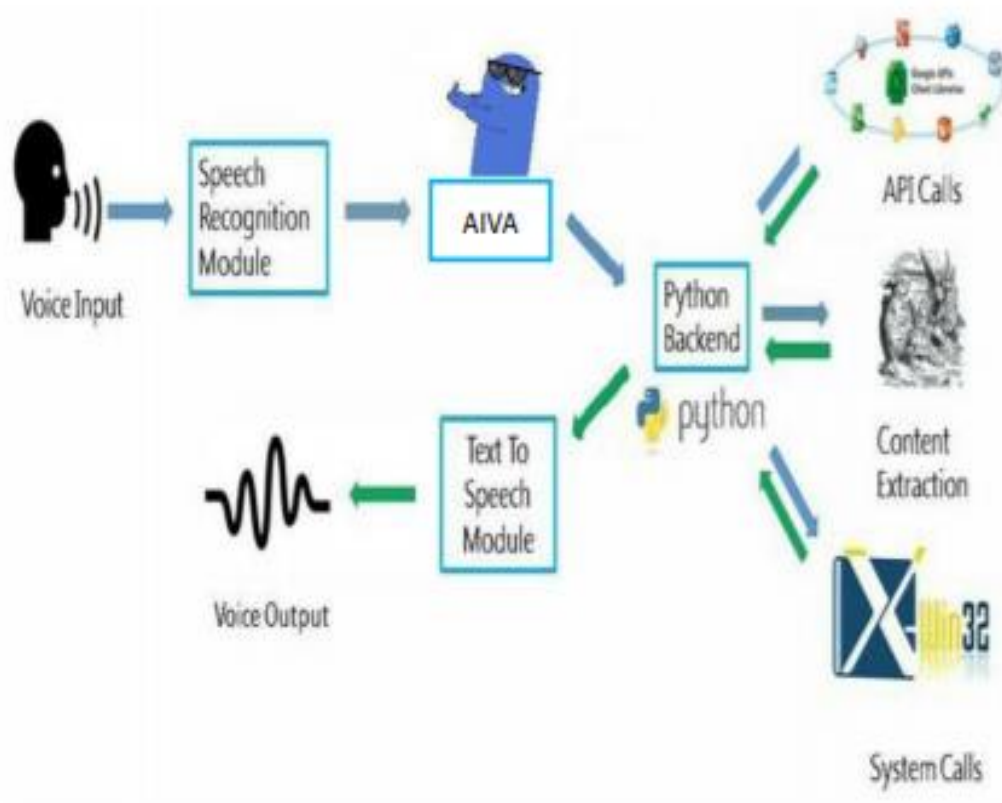


Fig 1

The work started with analyzing the audio commands given by the user through microphone. This can be anything like getting any information, operating computer's internal files, etc.

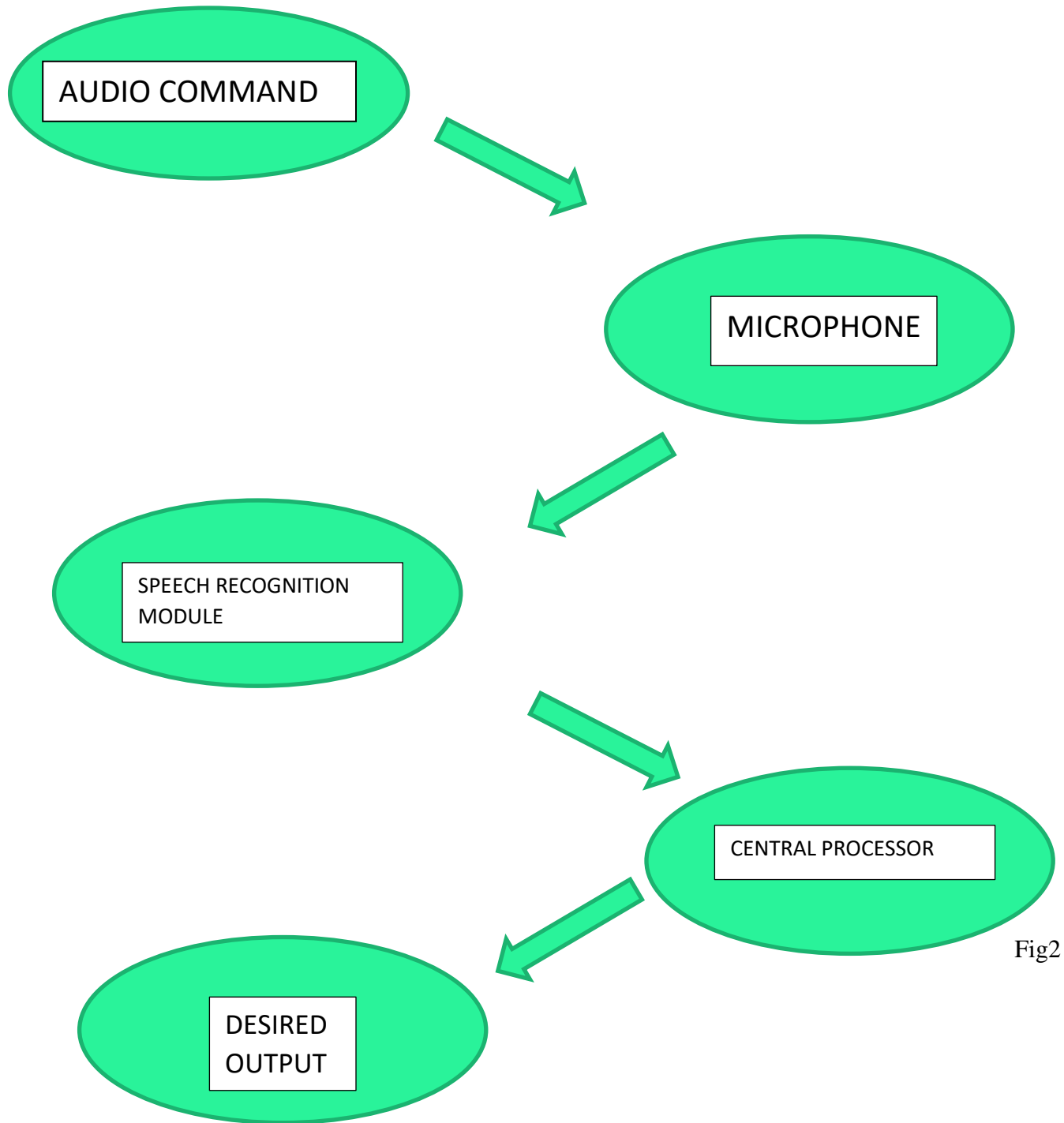


Fig2

Fig.2 shows the workflow of the basic process of the voice assistant. Speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relevant script for execution.

PROCEDURE

- We need to understand Python language in order to work on our project.
- We need to use to with interface for the coding.
- We need to understand all python modules used in the project such as:

```
import pyttsx3 as p
import speech_recognition as sr
from selenium_web import *
from YT_auto import *
import pyjokes
import datetime
from datetime import date
```

1. pyttsx3

It is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline. During construction, the engine initializes a `pyttsx3.driver.DriverProxy` object responsible for loading a speech engine driver implementation from the `pyttsx3.drivers` module.

2. Speech Recognition:-

Since we're building an Application of voice assistant, one of the most important things in this is that your assistant recognizes your voice (means what you want to say/ ask). To install this module type the below command in the terminal.

3. Datetime:-

Date and Time is used to showing Date and Time. This module comes built-int with Python.

4. Pyjokes

It is a python library that is used to create one-line jokes for programmers. Informally, it can also be referred as a fun python library which is pretty simple to use. Let us see how you can actually use it to perform the required task.

```
9 engine = p.init()
10 rate = engine.getProperty('rate')
11 engine.setProperty('rate', 150)
12 voices = engine.getProperty('voices')
13 engine.setProperty('voices', voices[1].id)
```

5. the built-in `__init__()` function.

All classes have a function called `__init__()`, which is always executed when the class is being initiated. Use the `__init__()` function to assign values to object properties, or other operations that are necessary to do when the object is being created

6. `property()`

It is a built-in function that creates and returns a property object. The syntax of this function is: `property(fget=None, fset=None, fdel=None, doc=None)` where, `fget` is function to get value of the attribute. `fset` is function to set value of the attribute. `fdel` is function to delete the attribute.

7. `get_property` method

It is used to get properties of an element, such as getting `text_length` property of anchor tag. This method will first try to return the value of a property with the given name.

```

22     speak("Hey,i am SHANKY How are you?")
23     engine.runAndWait()
24     with sr.Microphone() as source:
25         r.energy_threshold = 10000
26         r.adjust_for_ambient_noise(source,1.2)
27         print("Listening.....")
28         audio = r.listen(source)
29         text2= r.recognize_google(audio)
30         print(text2)
31         if "what " and "about" and "you" in text2:
32             speak("I am having a good day sir.")
33         speak("What can I do for you?")
34

```

8. adjust_for_ambient_noise function,

To handle the background noise, the recognizer class has a built-in function called `adjust_for_ambient_noise` function, which also takes a parameter of duration. Using this function the recognizer class listens to the audio for the specified duration seconds from the beginning of the aiod and then adjusts the energy threshold value so that the whole audio is more recognizable.

9. recognizer_instance.energy_threshold

The `recognizer_instance.energy_threshold` property is probably set to a value that is too high to start off with, and then being adjusted lower automatically by dynamic energy threshold adjustment. Before it is at a good level, the energy threshold is so high that speech is just considered ambient noise.

10. r.recognize_google

Google has a great Speech Recognition API. This API converts spoken text (microphone) into written text (Python strings), briefly Speech to Text. You can simply speak in a microphone and Google API will translate this into written text. The API has excellent results for English language.

11. r.listen

listen in Python calls the underlying listen syscall: listen() marks the socket referred to by sockfd as a passive socket, that is, as a socket that will be used to accept incoming connection requests using accept(2). A passive socket is the one you'd informally call the server.

12. datetime.datetime.now().strftime

The strftime() method returns a string representing date and time using date, time or datetime object. %Y, %m, %d etc. are format codes. The strftime() method takes one or more format codes as an argument and returns a formatted string based on it.

```
73 elif "time" in text2:
74     t=datetime.datetime.now().strftime('%I:%M %p')
75     print(t)
76     speak("Current time is "+t)
77
78
79 elif "date" in text2:
80     date = str(date.today())
81     print(date)
82     speak("Today's date is"+date)
```

%a

Abbreviated weekday name.

Sun, Mon,
...

%A

Full weekday name.

Sunday,
Monday,
...

%W

Weekday as a decimal number.

0, 1, ..., 6

%d

Day of the month as a zero-padded decimal.

01, 02, ...,
31

13. str ()

The str () function returns the string version of the given object. The syntax of str () is: str (object, encoding='utf-8', errors='strict')

14. pyjokes.get_joke()

Language and category are the two parameters of get_joke () and get_jokes () functions. Language specifies in which language you want the joke (s) to be displayed. By default, it is set to “en” that returns jokes in English.

15. driver = webdriver.Chrome

Python Chrome webdriver. In my code I used: to point the webdriver to the webdriver executable. Is there a way to point webdriver to the Chrome Browser binaries? In <https://sites.google.com/a/chromium.org/chromedriver/capabilities> they have the following (which I assume it what I'm looking for)

16. driver.get

The first line inside this method create a local reference to the driver object created in setUp method. The driver.get method will navigate to a page given by the URL. WebDriver will wait until the page has fully loaded (that is, the “onload” event has fired) before returning control to your test or script.

17. driver.find_element_by_xpath

This method returns a list with type of elements specified. XPath is the language used for locating nodes in an XML document. As HTML can be an implementation of XML (XHTML), Selenium users can leverage this powerful language to target elements in their web applications

18. .send_keys

send_keys method is used to send text to any field, such as input field of a form or even to anchor tag paragraph, etc. It replaces its contents on the webpage in your browser. It replaces its contents on the webpage in your browser.

19. click()

Click is a Python package for creating beautiful command line interfaces in a composable way with as little code as necessary.

CODE PHOTO

MAIN CODE

```
1  import pyttsx3 as p
2      import speech_recognition as sr
3      from selenium_web import *
4      from YT_auto import *
5      import pyjokes
6      import datetime
7      from datetime import date
8
9      engine = p.init()
10     rate = engine.getProperty('rate')
11     engine.setProperty('rate', 150)
12     voices = engine.getProperty('voices')
13     engine.setProperty('voices', voices[1].id)
14
15
16     def speak(text):
17         engine.say(text)
18         engine.runAndWait()
19
20     r = sr.Recognizer()
21
22     speak("Hey, i am SHANKY How are you?")
23     engine.runAndWait()
24     with sr.Microphone() as source:
25         r.energy_threshold = 10000
26         r.adjust_for_ambient_noise(source, 1.2)
27         print("Listening.....")
28         audio = r.listen(source)
29         text2= r.recognize_google(audio)
```

```

30     print(text2)
31     if "what" and "about" and "you" in text2:
32         speak("I am having a good day sir.")
33         speak("What can I do for you?")
34
35     with sr.Microphone() as source:
36         r.energy_threshold = 10000
37         r.adjust_for_ambient_noise(source, 1.2)
38         print("Listening.....")
39         audio = r.listen(source)
40         text2 = r.recognize_google(audio)
41
42
43
44     if "information" in text2:
45         speak("You need information related to which topic?")
46         with sr.Microphone() as source:
47             r.energy_threshold = 10000
48             r.adjust_for_ambient_noise(source, 1.2)
49             print("Listening.....")
50             audio = r.listen(source)
51             info = r.recognize_google(audio)
52
53         print("searching {} in wikipedia".format(info))
54         assist = infow()
55         assist.get_info(info)
56

```



```

59 elif "play" and "video" in text2 :
60     with sr.Microphone() as source:
61         r.energy_threshold = 10000
62         r.adjust_for_ambient_noise(source, 1.2)
63
64         audio = r.listen(source)
65         video = r.recognize_google(audio)
66         print("Listening.....")
67         print("Playing on youtube".format(video))
68
69     assist = music()
70     assist.play(video)
71
72
73 elif "time" in text2:
74     t=datetime.datetime.now().strftime('%I:%M %p')
75     print(t)
76     speak("Current time is "+t)
77
78
79 elif "date" in text2:
80     date = str(date.today())
81     print(date)
82     speak("Today's date is "+date)
83
84
85 elif "joke" or "jokes" in text2:
86     speak("Sure sir, get ready for some laugh")
87     speak(pviokes.get ioke())

```

<http://selenium web.py/>

```
1  from selenium import webdriver
2
3  class infow():
4      def __init__(self):
5          self.driver = webdriver.Chrome(executable_path='F:\DOWNLOADS\chromedriver.exe')
6
7
8      def get_info(self, query):
9          self.query = query
10         self.driver.get(url="https://www.wikipedia.org/")
11         search = self.driver.find_element_by_xpath('//*[@id="searchInput"]')
12         search.click()
13         search.send_keys(query)
14         enter = self.driver.find_element_by_xpath('//*[@id="search-form"]/fieldset/button')
15         enter.click()
```

YT_auto.py

```
1  from selenium import webdriver
2  '''from selenium.webdriver.common.desired_capabilities import DesiredCapabilities
3  cap = DesiredCapabilities.CHROME
4  cap = {'binary_location': "C:\Program Files\Google\Chrome"}
5  driver = webdriver.Chrome(desired_capabilities=cap, executable_path="C:\Program Files\Google\Chrome")
6  driver.get('http://google.com/')'''
7
8
9  class music():
10     def __init__(self):
11         self.driver = webdriver.Chrome(executable_path='F:\DOWNLOADS\chromedriver.exe')
12
13     def play(self, query):
14         self.query = query
15         self.driver.get(url="https://www.youtube.com/results?search_query="+query)
16
17         video = self.driver.find_element_by_xpath('//*[@id="video-title"]/yt-formatted-string')
18         video.click()
```

CONCLUSION

In this report, we discussed the design and implementation of a Digital Assistance. The project is built using open source software modules with PyCharm community backing which can accommodate any updates in the near future. The modular nature of this project makes it more flexible and easy to add additional features without disturbing current system functionalities.

It not only works on human commands but also give responses to the user on the basis of query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing each and every task. The entire system works on the verbal input rather than the text one.

RESULT

It worked very well and full fill our objective to make a voice assistants using python as mentioned above.

OUTCOME

The project is very useful and owns a large potential use in different industries. Although the program primary concerns more about how to do the personal assistant on Android phone using the voice, the concept of voice recognition can be applied in different industries as in many situations it will be more convenient, save a lot of time and helpful especially for those who have difficulty in working with manual operations. Thus, the concept is only for programming the Android application

REFERENCES

<https://www.bing.com/search?q=.click%28%29+meaning+in+python&qs=n&form=QBRE&sp=-1&pq=.click%28%29+meaning+in+python&sc=0-26&sk=&cvid=FB3DFC2DC5F84110AC0D297774A905B4>

<https://www.geeksforgeeks.org/python-script-to-create-random-jokes-using-pyjokes/>

<https://www.programiz.com/python->

https://www.youtube.com/watch?v=fjPuAKR_h4M&t=2714s

<https://stackoverflow.com/questions/33645534/pyttsx-loop-makes-runandwait-cut-off>

[1] G. Bohouta, V. Z. Këpuska, "Comparing Speech Recognition Systems (Microsoft API Google API And CMU Sphinx)",

Int. Journal of Engineering Research and Application 2017, 2017.

[2] Artificial intelligence (AI), sometimes called machine intelligence.
https://en.wikipedia.org/wiki/Artificial_intelligence

[3] B. Marr, The Amazing Ways Google Uses Deep Learning AI.

[4] Cortana Intelligence, Google Assistant, Apple Siri.

[5] Hill, J., Ford, W.R. and Farreras, I.G., 2015. Real conversations with artificial intelligence: A comparison between human–

human online conversations and human–chatbot conversations. Computers in Human Behavior, 49, pp.245-25