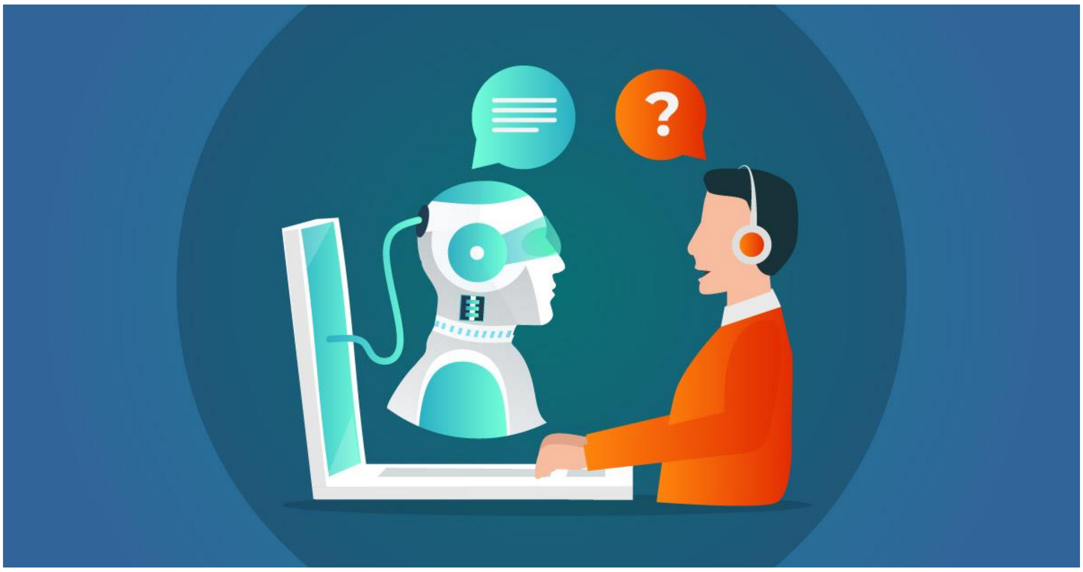
HACKATHON STUDY REPORT

ON

VIRTUAL ASSISTANT



Bachelor of Computer Application

DAV College Amritsar, Punjab

Submitted by Team No. (Batch ####)

RN	Name
7372	Piyush Pandey
7374	Aditya

Submitted to
Department of Computer Science
DAV College Amritsar, Punjab

Guide & Mentor: ###########

TABLE OF CONTENTS

Introduction to Artificial Intelligence

1.1 What is Artificial Intelligence (AI)?

According to the father of Artificial Intelligence **John McCarthy**, AI is the

"The science and engineering of making intelligent machines, especially intelligent computer programs".

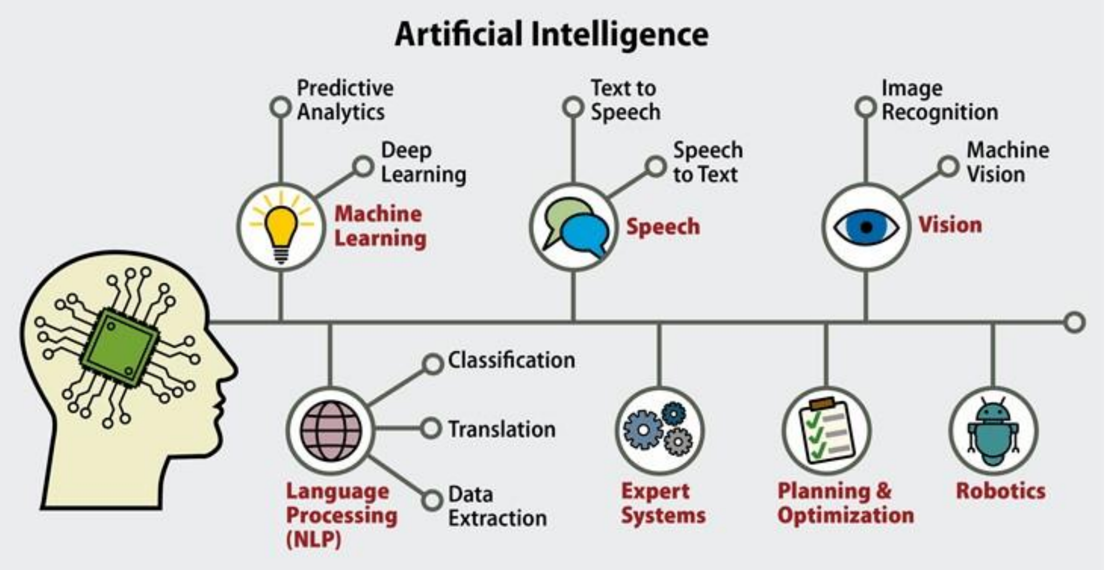


Figure 1: Artificial intelligence

Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction. Particular applications of AI include expert systems, speech recognition and machine vision.

Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think.

Artificial Intelligence (AI) and Virtual Assistants are two closely related technologies that have gained widespread popularity in recent years. AI refers to the ability of machines to perform tasks that would typically require human intelligence, such as problem-solving, reasoning, and decision-making. Virtual Assistants, on the other hand, are AI-powered software applications designed to interact with humans through natural language interfaces.

Virtual Assistants leverage AI technologies such as Natural Language Processing (NLP) and Machine Learning (ML) to understand human speech and respond appropriately. They can perform a wide range of tasks, including answering questions, scheduling appointments, setting reminders, and even making phone calls or sending text messages.

One of the most popular Virtual Assistants is Siri, developed by Apple. Siri can perform a wide range of tasks such as setting alarms, playing music, and answering questions. Another popular Virtual Assistant is Amazon's Alexa, which can be found in smart home devices such as the Amazon Echo.

AI-powered Virtual Assistants are becoming increasingly sophisticated, with the ability to understand context, recognize different voices, and learn from user behavior. They are being used in various industries, including healthcare, finance, and customer service.

2. What is Natural language Processing?

Voice processing in AI refers to the technology that enables machines to understand and interpret human speech. It involves a combination of natural language processing (NLP) and machine learning techniques that enable computers to process, analyze, and respond to spoken language.

In voice processing, the AI system captures audio data from a microphone and converts it into a digital signal. This signal is then analyzed using NLP algorithms, which help the system understand the meaning of the spoken words and phrases. The system may use machine learning techniques to improve its accuracy and learn from previous interactions.

Voice processing in AI has many applications, including virtual assistants like Siri and Alexa, speech recognition software, and language translation services. It has revolutionized the way we interact with technology, allowing us to use natural language to communicate with machines and access information and services more easily.

However, there are still some limitations to voice processing in AI. For example, it may struggle with accents, dialects, and non-standard speech patterns. It may also have difficulty with complex commands or questions that require context or background knowledge. As AI continues to develop, voice processing technology is likely to improve and become even more accurate and effective in understanding and interpreting human speech.

2.1 Types of NLP:

Sentiment Analysis:

This technique involves analyzing text to determine the sentiment or emotion behind it. Sentiment analysis is often used to gauge public opinion on a particular topic or product.

Named Entity Recognition (NER):

This technique involves identifying and categorizing entities such as people, organizations, locations, and other named objects within a text.

Text Classification:

This technique involves categorizing text into predefined categories or topics. Text classification is often used in applications such as spam filtering, news classification, and sentiment analysis.

Language Translation:

This technique involves translating text from one language to another. Machine translation is often used in applications such as language learning and cross-lingual communication.

Information Extraction:

This technique involves identifying and extracting specific information from unstructured text data. This information can include things like dates, names, and other important details.

Question Answering:

This technique involves using NLP to automatically answer questions posed in natural language. Question answering is often used in virtual assistants and chatbots.

Natural Language Generation:

This technique involves using NLP to generate text in natural language. Natural language generation is often used in applications such as automated writing, customer service, and chatbots.

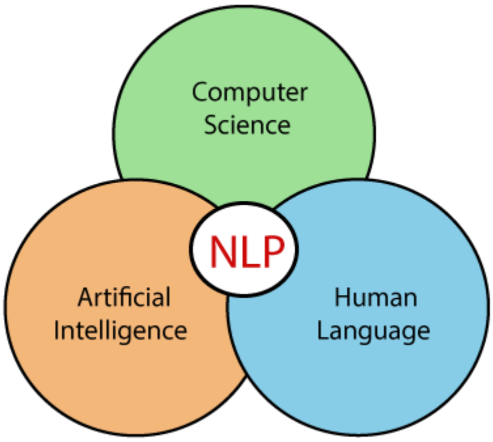


Figure 2: NPL

2.3 INPUTS

All AI Problems require some form of representation:

- Python
- Tkinter
- speech_recognition
- pygame
- tkVideoPlayer
- googletrans

Tkinter is a standard Python library for creating graphical user interfaces (GUIs). It is a built-in module that allows developers to create applications with a graphical interface for their users. Tkinter is a cross-platform GUI toolkit that works on Windows, macOS, and Linux operating systems.

Tkinter is based on the Tcl/Tk GUI toolkit and provides a set of tools for creating GUI applications, such as buttons, labels, text boxes, and more. It also provides a canvas widget that can be used to create complex graphics and animations.

The core of Tkinter is the Tk library, which is a toolkit for creating graphical user interfaces. Tkinter provides Python bindings for the Tk library, allowing developers to create GUI applications with Python code. Tkinter also provides a set of event-driven functions that allow developers to respond to user input and update the GUI accordingly.

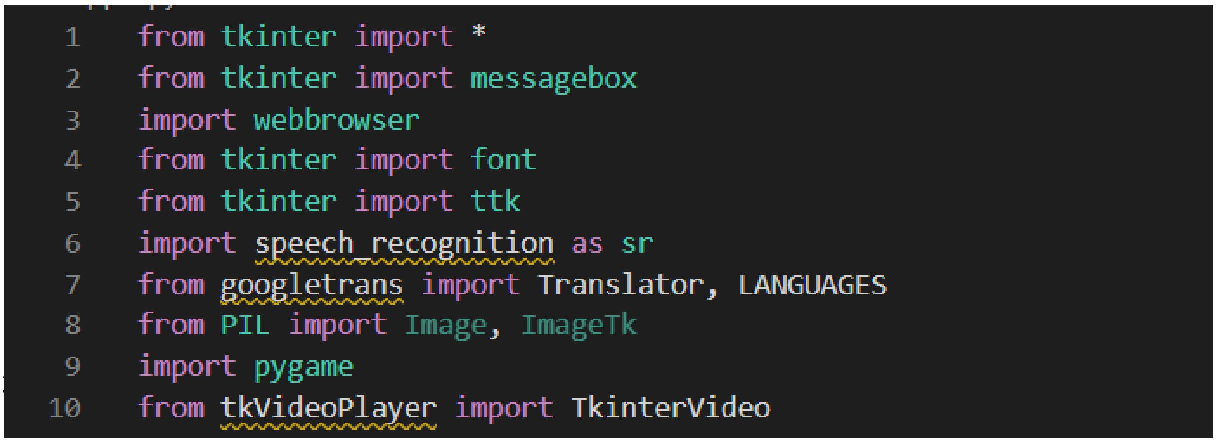
2.4. OUTPUT

The output action also can be complex

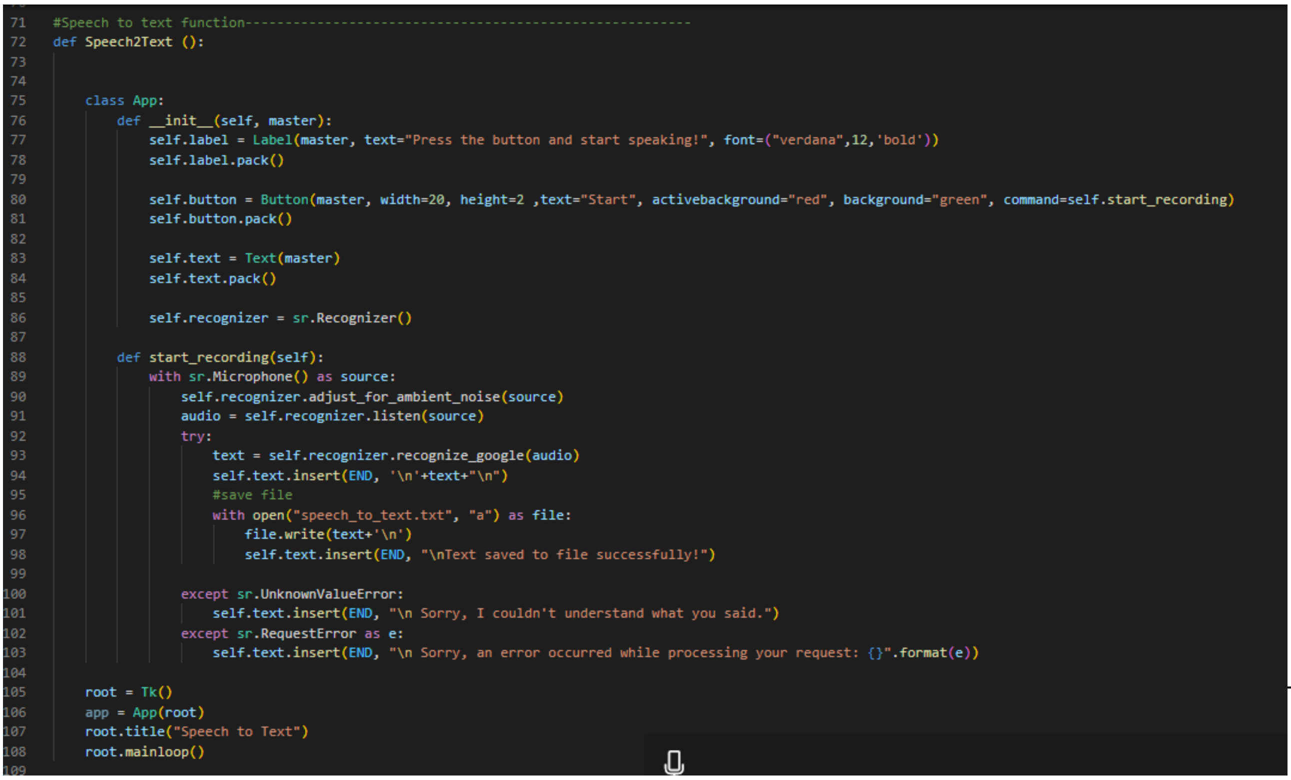
- Translation feature
- speech to text feature
- text to speech feature
- online search feature
- virtual command

There are several libraries in Python that provide speech-to-text functionality. One of the most popular ones is the Google Cloud Speech-to-Text API, which allows you to convert audio files to text using Google's machine learning technology

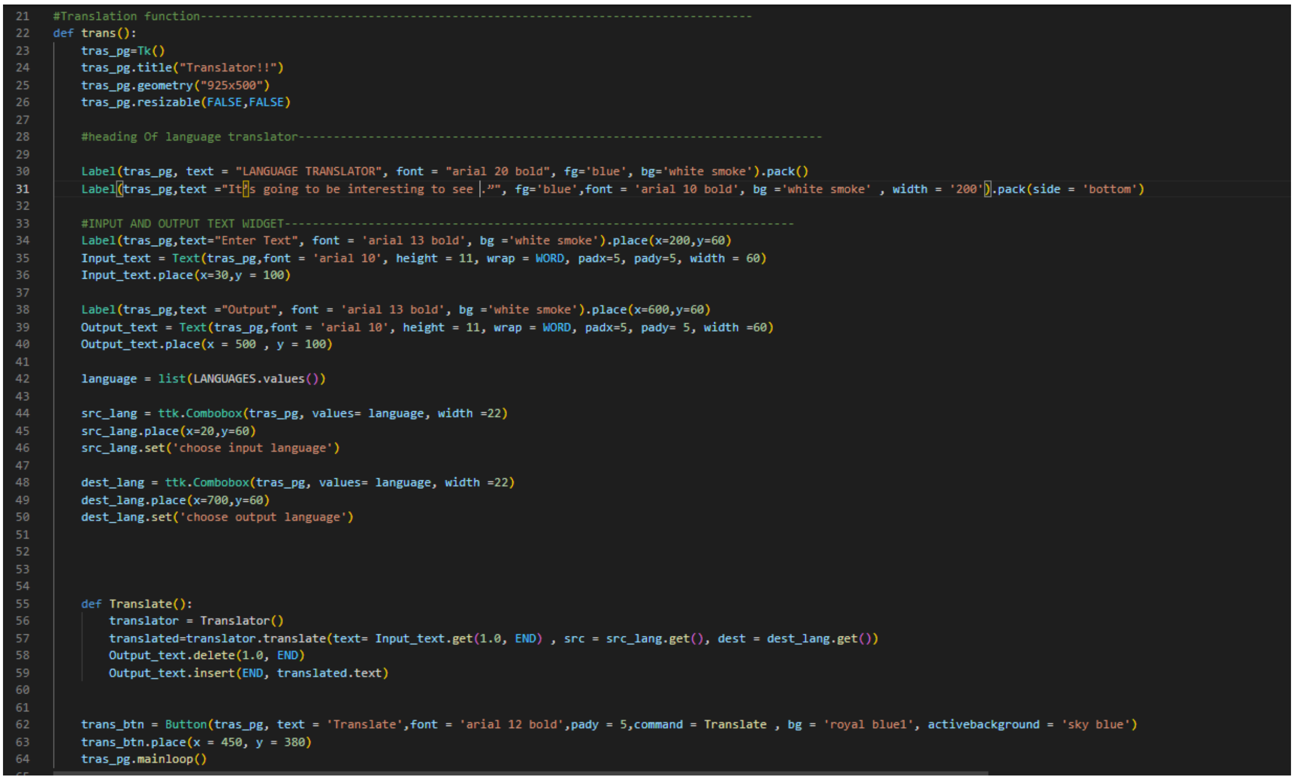
3. Source Code:



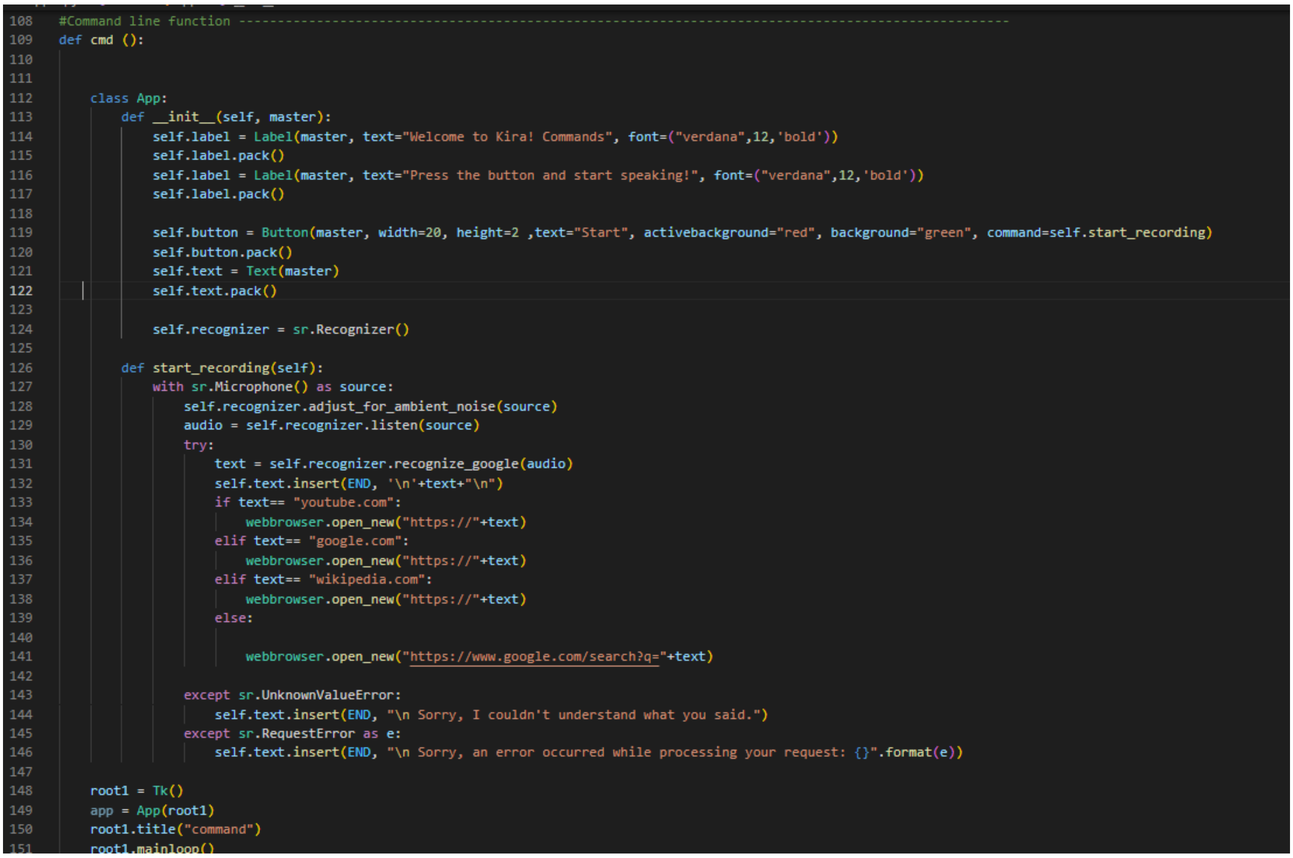
```python
from tkinter import *
from tkinter import messagebox
import webbrowser
from tkinter import font
from tkinter import ttk
import speech_recognition as sr
from googletrans import Translator, LANGUAGES
from PIL import Image, ImageTk
import pygame
from tkVideoPlayer import TkinterVideo
```



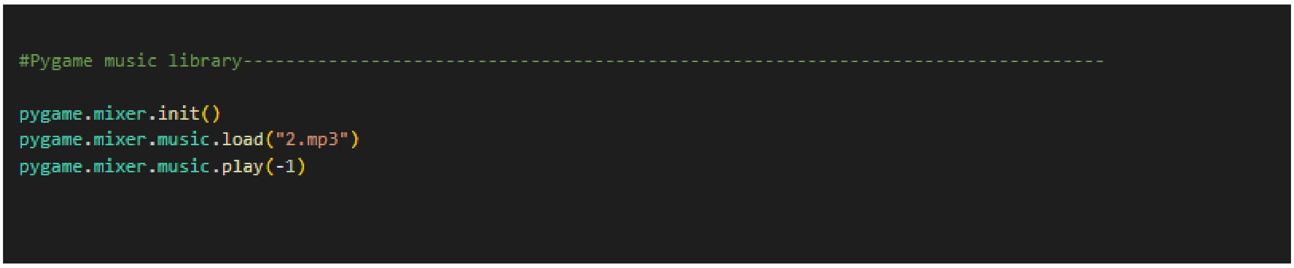
```python
#Speech to text function-----------------------------------------------------
def Speech2Text ():


    class App:
        def __init__(self, master):
            self.label = Label(master, text="Press the button and start speaking!", font=("verdana",12,'bold'))
            self.label.pack()

            self.button = Button(master, width=20, height=2 ,text="Start", activebackground="red", background="green", command=self.start_recording)
            self.button.pack()

            self.text = Text(master)
            self.text.pack()

            self.recognizer = sr.Recognizer()

        def start_recording(self):
            with sr.Microphone() as source:
                self.recognizer.adjust_for_ambient_noise(source)
                audio = self.recognizer.listen(source)
                try:
                    text = self.recognizer.recognize_google(audio)
                    self.text.insert(END, '\n'+text+"\n")
                    #save file
                    with open("speech_to_text.txt", "a") as file:
                        file.write(text+'\n')
                        self.text.insert(END, "\nText saved to file successfully!")

                except sr.UnknownValueError:
                    self.text.insert(END, "\n Sorry, I couldn't understand what you said.")
                except sr.RequestError as e:
                    self.text.insert(END, "\n Sorry, an error occurred while processing your request: {}".format(e))

    root = Tk()
    app = App(root)
    root.title("Speech to Text")
    root.mainloop()
```



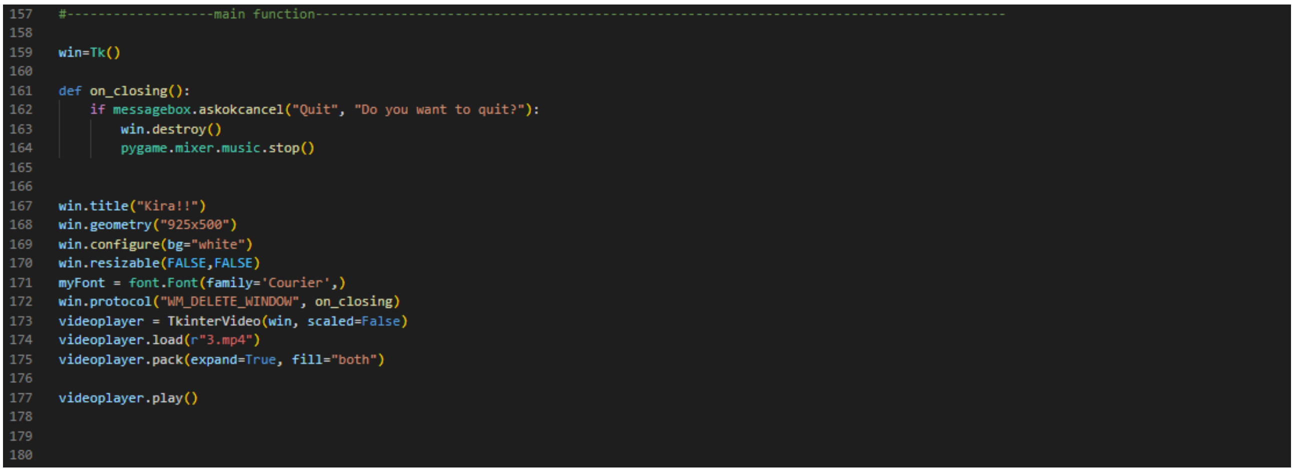
```python
#Translation function---------------------------------------------------------------------------
def trans():
    tras_pg=Tk()
    tras_pg.title("Translator!!")
    tras_pg.geometry("925x500")
    tras_pg.resizable(FALSE,FALSE)

    #heading Of language translator------------------------------------------------------------------

    Label(tras_pg, text = "LANGUAGE TRANSLATOR", font = "arial 20 bold", fg='blue', bg='white smoke').pack()
    Label(tras_pg,text ="It's going to be interesting to see .”", fg='blue',font = 'arial 10 bold', bg ='white smoke' , width = '200').pack(side = 'bottom')

    #INPUT AND OUTPUT TEXT WIDGET-------------------------------------------------------------------
    Label(tras_pg,text="Enter Text", font = 'arial 13 bold', bg ='white smoke').place(x=200,y=60)
    Input_text = Text(tras_pg,font = 'arial 10', height = 11, wrap = WORD, padx=5, pady=5, width = 60)
    Input_text.place(x=30,y = 100)

    Label(tras_pg,text ="Output", font = 'arial 13 bold', bg ='white smoke').place(x=600,y=60)
    Output_text = Text(tras_pg,font = 'arial 10', height = 11, wrap = WORD, padx=5, pady= 5, width =60)
    Output_text.place(x = 500 , y = 100)

    language = list(LANGUAGES.values())

    src_lang = ttk.Combobox(tras_pg, values= language, width =22)
    src_lang.place(x=20,y=60)
    src_lang.set('choose input language')

    dest_lang = ttk.Combobox(tras_pg, values= language, width =22)
    dest_lang.place(x=700,y=60)
    dest_lang.set('choose output language')



    def Translate():
        translator = Translator()
        translated=translator.translate(text= Input_text.get(1.0, END) , src = src_lang.get(), dest = dest_lang.get())
        Output_text.delete(1.0, END)
        Output_text.insert(END, translated.text)


    trans_btn = Button(tras_pg, text = 'Translate',font = 'arial 12 bold',pady = 5,command = Translate , bg = 'royal blue1', activebackground = 'sky blue')
    trans_btn.place(x = 450, y = 380)
    tras_pg.mainloop()
```



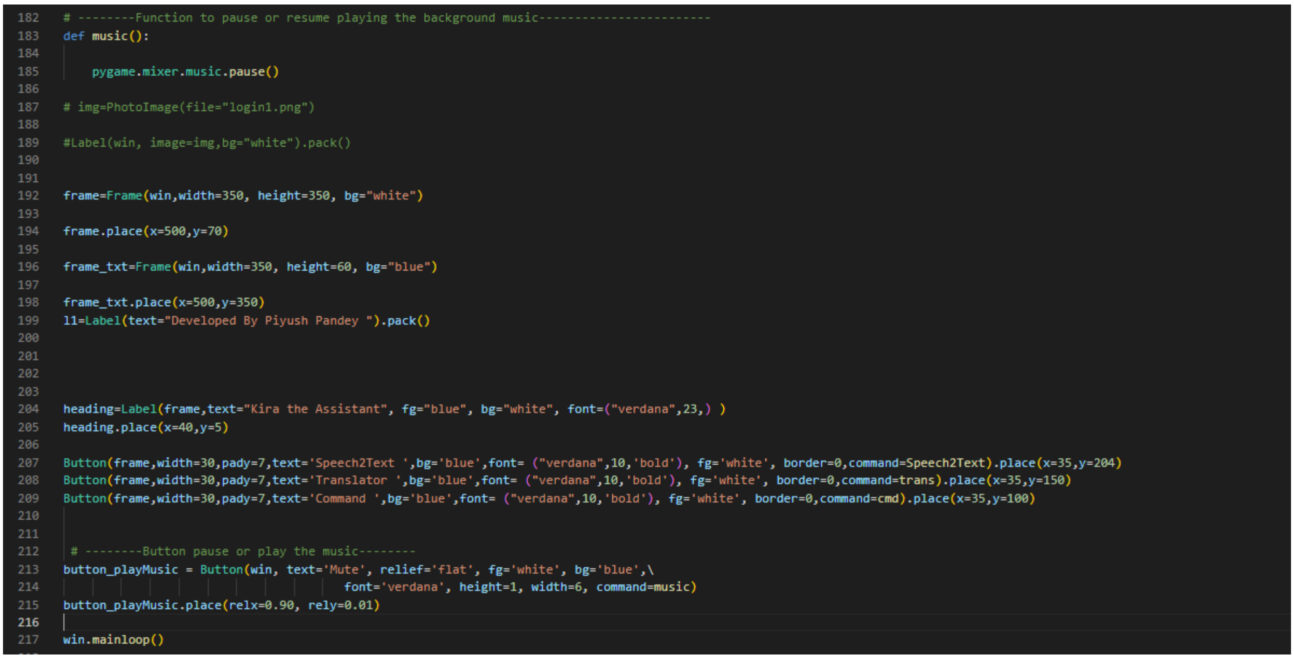
```python
#Command line function ------------------------------------------------------------------------------------
def cmd ():


    class App:
        def __init__(self, master):
            self.label = Label(master, text="Welcome to Kira! Commands", font=("verdana",12,'bold'))
            self.label.pack()
            self.label = Label(master, text="Press the button and start speaking!", font=("verdana",12,'bold'))
            self.label.pack()

            self.button = Button(master, width=20, height=2 ,text="Start", activebackground="red", background="green", command=self.start_recording)
            self.button.pack()
            self.text = Text(master)
            self.text.pack()

            self.recognizer = sr.Recognizer()

        def start_recording(self):
            with sr.Microphone() as source:
                self.recognizer.adjust_for_ambient_noise(source)
                audio = self.recognizer.listen(source)
                try:
                    text = self.recognizer.recognize_google(audio)
                    self.text.insert(END, '\n'+text+"\n")
                    if text== "youtube.com":
                        webbrowser.open_new("https://"+text)
                    elif text== "google.com":
                        webbrowser.open_new("https://"+text)
                    elif text== "wikipedia.com":
                        webbrowser.open_new("https://"+text)
                    else:

                        webbrowser.open_new("https://www.google.com/search?q="+text)

                except sr.UnknownValueError:
                    self.text.insert(END, "\n Sorry, I couldn't understand what you said.")
                except sr.RequestError as e:
                    self.text.insert(END, "\n Sorry, an error occurred while processing your request: {}".format(e))

    root1 = Tk()
    app = App(root1)
    root1.title("command")
    root1.mainloop()
```



```python
#Pygame music library------------------------------------------------------------------------------

pygame.mixer.init()
pygame.mixer.music.load("2.mp3")
pygame.mixer.music.play(-1)
```

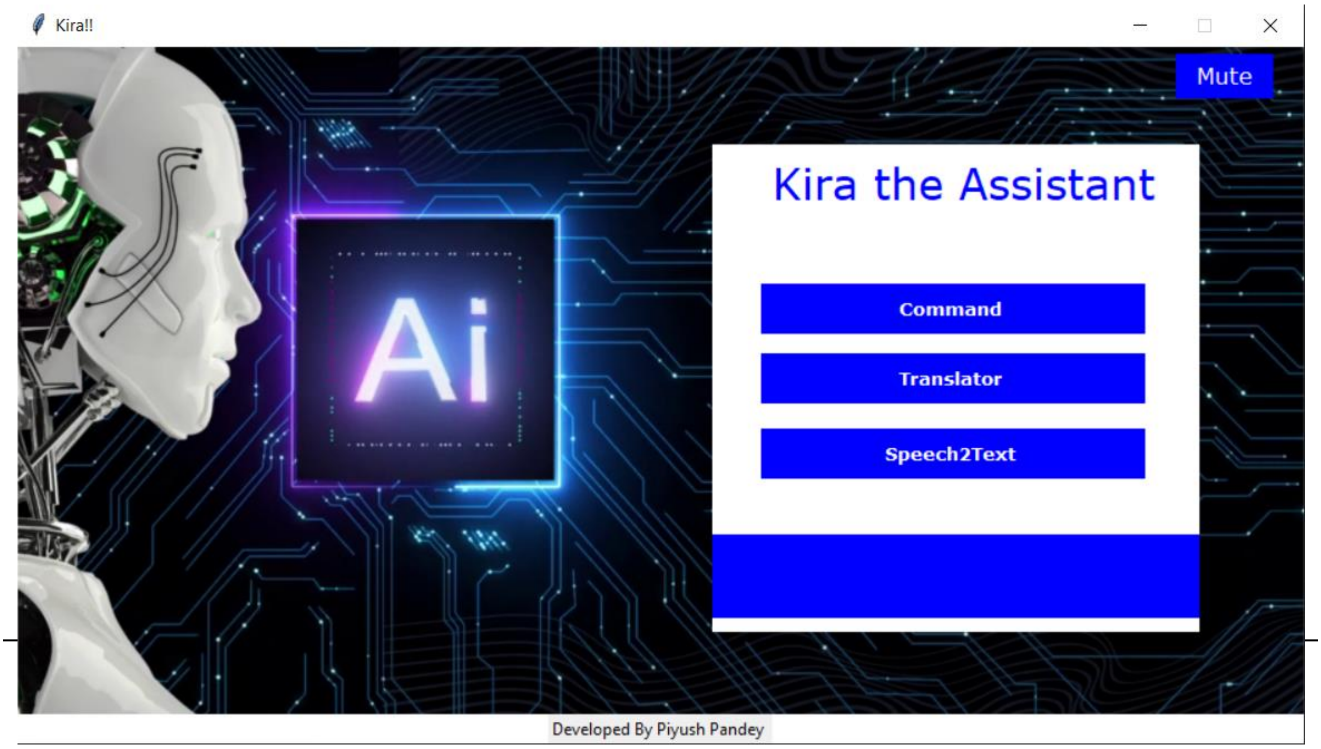


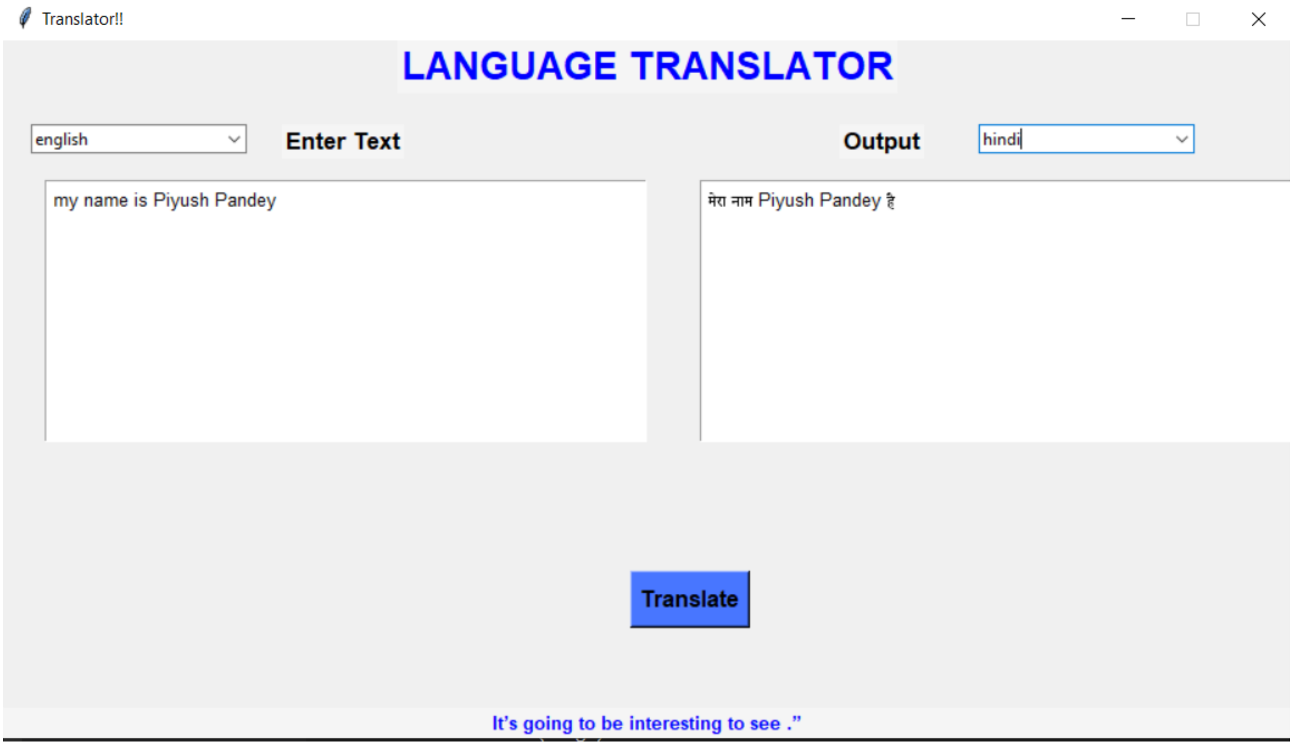
```python
#------------------main function------------------------------------------------------------------------------------------

win=Tk()

def on_closing():
    if messagebox.askokcancel("Quit", "Do you want to quit?"):
        win.destroy()
        pygame.mixer.music.stop()


win.title("Kira!!")
win.geometry("925x500")
win.configure(bg="white")
win.resizable(FALSE,FALSE)
myFont = font.Font(family='Courier',)
win.protocol("WM_DELETE_WINDOW", on_closing)
videoplayer = TkinterVideo(win, scaled=False)
videoplayer.load(r"3.mp4")
videoplayer.pack(expand=True, fill="both")

videoplayer.play()



```



```python
# --------Function to pause or resume playing the background music-----------------------
def music():

    pygame.mixer.music.pause()

# img=PhotoImage(file="login1.png")

#Label(win, image=img,bg="white").pack()


frame=Frame(win,width=350, height=350, bg="white")

frame.place(x=500,y=70)

frame_txt=Frame(win,width=350, height=60, bg="blue")

frame_txt.place(x=500,y=350)
l1=Label(text="Developed By Piyush Pandey ").pack()




heading=Label(frame,text="Kira the Assistant", fg="blue", bg="white", font=("verdana",23,) )
heading.place(x=40,y=5)

Button(frame,width=30,pady=7,text='Speech2Text ',bg='blue',font= ("verdana",10,'bold'), fg='white', border=0,command=Speech2Text).place(x=35,y=204)
Button(frame,width=30,pady=7,text='Translator ',bg='blue',font= ("verdana",10,'bold'), fg='white', border=0,command=trans).place(x=35,y=150)
Button(frame,width=30,pady=7,text='Command ',bg='blue',font= ("verdana",10,'bold'), fg='white', border=0,command=cmd).place(x=35,y=100)


 # --------Button pause or play the music--------
button_playMusic = Button(win, text='Mute', relief='flat', fg='white', bg='blue',\
                                        font='verdana', height=1, width=6, command=music)
button_playMusic.place(relx=0.90, rely=0.01)

win.mainloop()
```

3.1 OUTPUT:





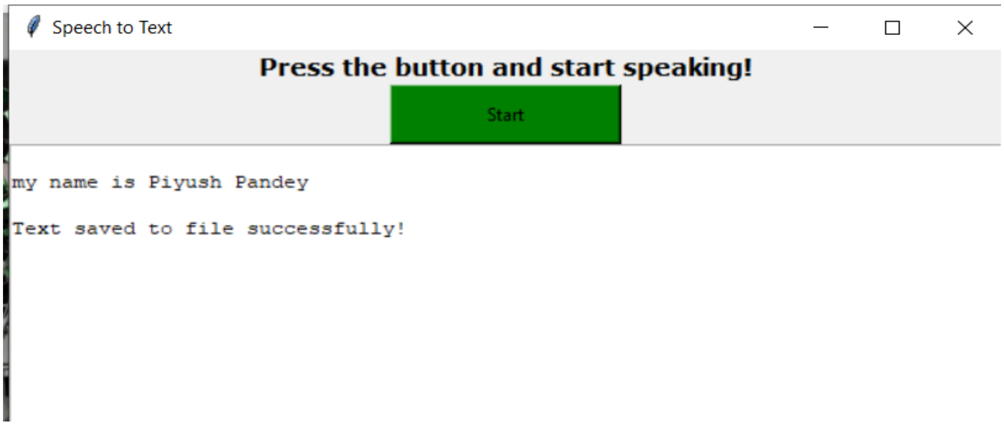
english

my name is Piyush Pandey

hindi

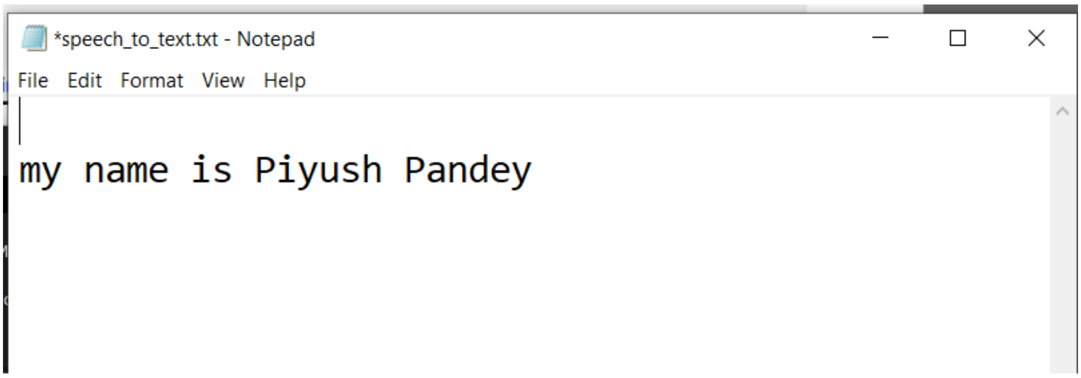
मेरा नाम Piyush Pandey है

It’s going to be interesting to see .”

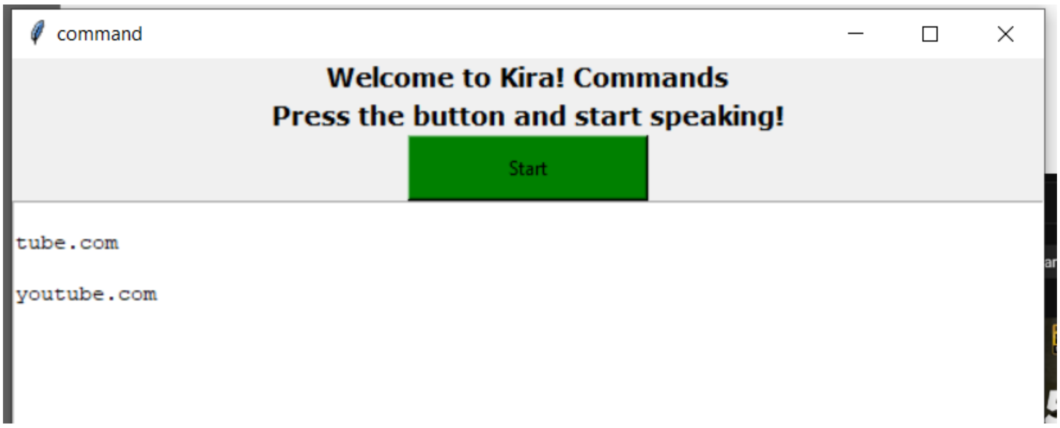


Press the button and start speaking!

```
my name is Piyush Pandey

Text saved to file successfully!
```

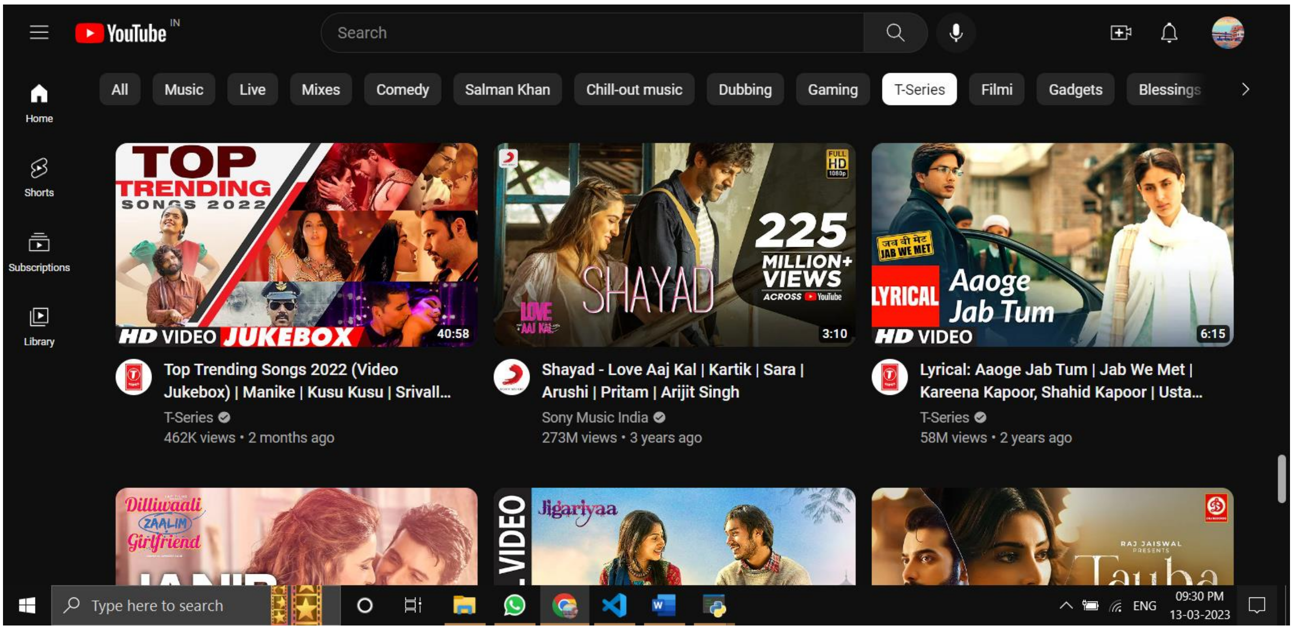


```
my name is Piyush Pandey
```



Welcome to Kira! Commands
Press the button and start speaking!

```
tube.com

youtube.com
```



4. Contribution towards society:

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on enabling machines to understand, interpret and manipulate human language. NLP has numerous applications that can contribute to society in various ways. Here are some examples:

Enhancing accessibility: NLP can be used to develop assistive technologies such as text-to-speech and speech-to-text software, which can make it easier for people with disabilities to interact with technology and access information.

Improving healthcare: NLP can help medical professionals analyze and understand medical records, reports and other healthcare data, leading to better diagnoses, treatments and patient outcomes.

Streamlining customer service: NLP-powered chatbots can assist customers in answering their queries in a more efficient and timely manner, thereby reducing wait times and improving customer satisfaction.

Enhancing education: NLP can help personalize learning experiences for students, assist teachers in grading assignments and providing feedback, and develop more interactive and engaging educational materials.

Enhancing security: NLP can be used in cybersecurity to detect and prevent fraudulent activities and identity theft.

Improving language translation: NLP can help break down language barriers by developing better translation software, which can make it easier for people to communicate and share ideas across languages and cultures.

Overall, NLP has the potential to improve many aspects of society by enabling more efficient and effective communication, decision-making, and problem-solving

It’s unsurprising that artificial intelligence grew rapidly post-1900, but what *is* surprising, is how many people thought about AI hundreds of years before there was even a word to describe what they were thinking about.

4.1 Challenges:

Learning Natural Language Processing (NLP) can be challenging, especially for those who are new to the field of artificial intelligence and machine learning. Here are some common challenges that one might face when learning NLP:

Understanding the concepts: NLP involves a variety of complex concepts such as text pre-processing, feature engineering, and neural network architectures. Understanding these concepts can be difficult, especially for those who are not familiar with programming or data analysis.

Accessing high-quality datasets: Developing machine learning models for NLP requires high-quality datasets, which can be difficult to find or create. Additionally, some datasets may be biased or incomplete, which can negatively impact the accuracy of the models.

Managing large datasets: NLP datasets can be very large, making it difficult to manage and process the data efficiently.

Choosing the right algorithms: There are many algorithms available for NLP, and choosing the right one for a particular task can be challenging. Additionally, some algorithms may require a deep understanding of mathematics or statistics to implement and optimize.

Keeping up with new research: NLP is a rapidly evolving field, with new research and techniques emerging regularly. Keeping up with the latest developments can be challenging but is essential for developing state-of-the-art models.

Ethical considerations: As with any technology, NLP can be used for harmful purposes, such as spreading disinformation or invading people's privacy. Therefore, it's important to be aware of the ethical implications of NLP and use it responsibly.

In summary, learning NLP can be challenging, but with persistence and dedication, one can overcome these challenges and develop the skills necessary to work with natural language data and develop innovative solutions that contribute to society.

5. TOOLS AND TECHNIQUE USED:

Python is a popular programming language for developing language translators due to its ease of use, versatility, and large ecosystem of libraries and tools. Here are some ways Python can be used in developing language translators:

Pre-processing: Python can be used for pre-processing text data, such as tokenization, stemming, and lemmatization. These techniques help convert raw text into a format that is easier for algorithms to process.

Machine learning: Python has several powerful machine learning libraries, such as Scikit-learn and TensorFlow, which can be used to train and deploy machine learning models for language translation tasks.

Neural networks: Python has several deep learning libraries, such as PyTorch and Keras, which can be used to build and train neural networks for language translation tasks. These networks can perform well for complex translation tasks.

API integration: Python can be used to integrate with existing translation APIs, such as Google Translate or Microsoft Translator, to develop custom translation applications that meet specific business needs.

Natural Language Toolkit (NLTK): NLTK is a powerful library for NLP tasks in Python that provides a wide range of tools for language processing, including stemming, lemmatization, part-of-speech tagging, and more. It can be used to build custom language translation applications.

Overall, Python is a versatile and powerful language that can be used to build a variety of language translation applications, from simple rule-based systems to complex neural network models.

Tkinter is a standard Python library for creating graphical user interfaces (GUIs). It is a built-in module that allows developers to create applications with a graphical interface for their users. Tkinter is a cross-platform GUI toolkit that works on Windows, macOS, and Linux operating systems.

Tkinter is based on the Tcl/Tk GUI toolkit and provides a set of tools for creating GUI applications, such as buttons, labels, text boxes, and more. It also provides a canvas widget that can be used to create complex graphics and animations.

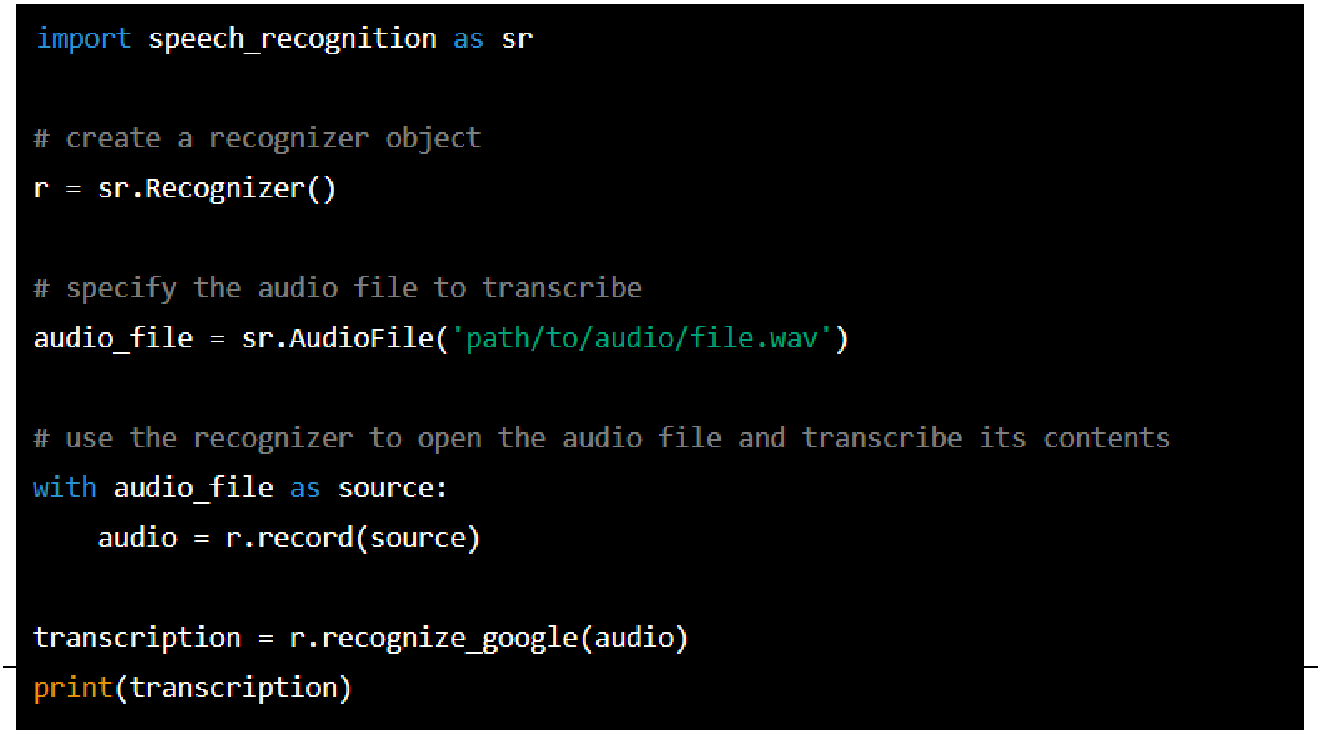
The core of Tkinter is the Tk library, which is a toolkit for creating graphical user interfaces. Tkinter provides Python bindings for the Tk library, allowing developers to create GUI applications with Python code. Tkinter also provides a set of event-driven functions that allow developers to respond to user input and update the GUI accordingly.

Googletrans library is a Python package that provides a simple interface for using the Google Translate API. It allows users to translate text from one language to another using Google's machine translation service.

To use googletrans, you will need to install it using pip:

Copy code
pip install googletrans

In Python, you can use the SpeechRecognition library to perform speech recognition. This library supports several popular speech recognition APIs, including Google Cloud Speech API, IBM Speech to Text API, and Sphinx.



```python
import speech_recognition as sr

# create a recognizer object
r = sr.Recognizer()

# specify the audio file to transcribe
audio_file = sr.AudioFile('path/to/audio/file.wav')

# use the recognizer to open the audio file and transcribe its contents
with audio_file as source:
    audio = r.record(source)

transcription = r.recognize_google(audio)
print(transcription)
```

6.Future Aspects :

The future of assistants and Natural Language Processing (NLP) is very promising. As technology continues to advance, assistants and NLP will become even more sophisticated and intuitive, making them more useful and user-friendly.

One major trend in the future of assistants and NLP is the integration of voice recognition technology. With the rise of smart speakers and voice-activated devices, voice recognition has become an increasingly important aspect of NLP. As a result, assistants will become more conversational and interactive, allowing users to have more natural and seamless interactions.

Another trend in the future of assistants and NLP is the increased use of machine learning and artificial intelligence (AI) to improve their performance. Machine learning algorithms can be used to analyze large amounts of data and improve the accuracy of assistants and NLP systems. This will lead to more personalized and customized experiences for users.

Additionally, there will be a growing emphasis on privacy and security in the future of assistants and NLP. As these technologies become more ubiquitous, there will be a greater need to protect user data and ensure that it is not misused or compromised.

Overall, the future of assistants and NLP looks very promising, with continued advancements in technology leading to more sophisticated and intuitive systems that can provide even greater value to users.

7.Conclusion :

Artificial intelligence advancements are occurring at an unprecedented rate. That being said, we can expect that the trends from the past decade will continue swinging upward in the coming year. A few things to keep our eyes on in 2019 include. The ability of computers (AI) is growing at a faster pace as compared to the past decade. All things these also in mind; these are no second thought about the rise & evolvement it has provided towards the technological advancement.

8. Glossary :

1. A.I.	-	Artificial Intelligence
2. N.L.P.	-	Natural Language Processing
3. R.P.A.	-	Robotics Process Automation
4. M.L.	-	Machine Learning
5. IoT	-	Internet of Things