



AMITY UNIVERSITY

-----Maharashtra-----

Project Report on

Kratzz: Language and Sentiment Detecting Chatbot
using Natural Language Processing

Submitted to

Amity Institute of Information Technology

In partial fulfillment of the requirements for the award of the degree of

Bachelor of Computer Applications

Submitted To

Dr. Preeti Gupta

Submitted By

Ms. Krati Rastogi

Enrolment No: A71004818006

Batch: 2018-2021

DECLARATION BY STUDENT

I, **Krati Rastogi**, student of BCA hereby declare that the Project titled “**Kratzz: Language and Sentiment Detecting Chatbot using Natural Language Processing**” which was submitted by me to **Dr. Preeti Gupta**, Associate professor, Amity Institute of Information Technology, Amity University Mumbai, Maharashtra, in partial fulfillment of requirement for the award of the degree of **Bachelors of Computer Applications**, has not previously formed the basis for the award of any degree, diploma or other similar title or recognition.

The Author attests that permission has been obtained for the use of any copyrighted material appearing in the Dissertation / Project report other than brief excerpts requiring only proper acknowledgement in scholarly writing and all such use is acknowledged.



Place: - Mumbai

Ms. Krati Rastogi

A71004818006

GUIDE CERTIFICATE

I hereby certify that the Seminar Report by **Ms. Krati Rastogi**, student of **Bachelors of Computer Applications, Semester-6, Enrolment Number A71004818006** with title **“Kratzz: Language and Sentiment Detecting Chatbot using Natural Language Processing”** which is submitted to **Amity Institute of Information Technology, Amity University Maharashtra**, Mumbai in partial fulfillment of requirement for the award of the degree of **Bachelors of Computer Applications** is an original contribution with existing knowledge and faithful record of work carried out by him/her under my guidance and supervision and 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: - Mumbai

Dr. Preeti Gupta

Assistant Professor

ABSTRACT

A chatbot is a software application used to conduct an online conversation. It is a computer program that anybody can talk to in a normal language. With the technology change, developers are building Artificial Intelligence based chatbots which use reinforcement learning concept, i.e the machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback. Although chatbots can perform many tasks, the primary function they have to play is to understand the utterances of humans and to respond appropriately. This project merges three concepts- application of chatbot, language detecting and translating function and classifying the sentiment of the sentence. Combining these three concepts results in “Kratzz: Language and Sentiment Detecting Chatbot”. With the help of various libraries used for Natural Language Processing, the accuracy of translation and detection is much higher when a user feeds a complete and accurate sentence. The chatbot uses the same technology as Google translator to translate the sentences efficiently. It takes inputs in two different forms that are via text and speech. The report also discusses the potential improvements that can be done in the future. The chatbot can take other mediums like an audio file to take the input which will be converted into a text file and get translated using Google API.

ACKNOWLEDGEMENT

It is a high privilege for me to express my deep sense of gratitude to those entire faculty Members who helped me in the completion of the project, especially my internal guide Dr. Preeti Gupta who was always there at hour of need.

My special thanks to all other faculty members, Batch mate & seniors of Amity Institute of Information Technology, Amity University, Maharashtra for helping me in the completion of project work and its report submission.

I would also like to give a special thanks to Dr. Manoj Devare, Head of Institution, and Ms. Kirti Kakde for giving us a proper platform and guidance to work on this project.

Ms. Krati Rastogi

A71004818006

TABLE OF CONTENTS

S.No	Contents	Page No
1.	DECLARATION BY STUDENT	ii
2.	GUIDANCE CERTIFICATE	iii
3.	ABSTRACT	iv
4.	ACKNOWLEDGEMENT	v
5.	LIST OF FIGURES	vi
Chapter-1	Introduction	9
	1.1 Preliminary	12
	1.2 Motivation	12
	1.3 Project Definition	12
	1.4 Objective	13
Chapter-2	Feasibility Study	14
Chapter-3	Proposed System	15
	3.1 Technologies Used	15
Chapter-4	System Analysis	18
Chapter-5	Architecture	20
Chapter-6	Implementation	21
	6.1 Detecting the language	21
	6.2 Translating the language	23
	6.3 Recognizing speech and converting into text	24

	6.4 Analyze the sentiment of a text	25
	6.5 Developing user interface using Qt designer	26
Chapter-7	Testing	28
Chapter-8	Future Scope & Conclusion	33
	References	34

LIST OF FIGURES

Figure No	Caption of the Figure	Page No.
Figure 1	Flowchart of Kratzz Chatbot	19
Figure 2	Basic architecture of Kratzz Chatbot	20
Figure 3	Implementation of detecting language	22
Figure 4	Implementation of translating text function	23
Figure 5	Implementing Speech-to-text function	24
Figure 6	Implementing Sentiment Analysis	25
Figure 7	Graphical User Interface of QT Designer	26
Figure 8	Early Draft of UI	27
Figure 9	Basic UI Design	27
Figure 10	Displays message when the input is incorrect	28
Figure 11	Displays the incorrect message	29
Figure 12	Displays wrong output due to short sentence	29
Figure 13	Displays try again when the language is fed incorrectly	30
Figure 14	The user select hindi as a preferred language	30
Figure 15	If the user selects 'N' for the sentiment analysis	31
Figure 16	If the user selects speech as an input medium	31
Figure 17	Throws error	32

Chapter 1- Introduction

Generally, chatbots are based on Artificial Intelligence. Now what is Artificial Intelligence and how does it work in the chatbot?

What is Artificial Intelligence?

The term artificial intelligence (AI) refers to the ability of a computer or machine to have the ability to act as a human mind that learns from the given knowledge and various experiments and experiences. The experience can include recognizing objects, emotions, making decisions based on the previous experience, and solving issues. By integrating all these capabilities, AI can perform functions a human might perform. For instance, AI can predict the next word a user might use next in a sentence. Other examples could be recommending movies and songs on platforms like Netflix and Spotify. The model trains itself by consuming the data used by the user. For example, if a user has watched movies that come under the action and comedy genre then the AI will recommend similar kinds of movies^[1].

Artificial Intelligence falls under two broad categories:

1. **Narrow AI:** it is generally referred to as Weak AI as it operates within a limited context. It primarily focuses on performing a single task with accuracy. The machines may seem intelligent but they are getting operated under limitations and constraints. There are few examples of Narrow AI are given below:
 - a. Google search
 - b. Image recognition software
 - c. Siri, Alexa, and other personal assistants
 - d. Self-driving cars
 - e. IBM's Watson

-
2. **Artificial General Intelligence (AGI):** it is generally referred to as a Strong AI. It is a machine with general intelligence and it is more like a human mind as it can apply intelligence to solve any problem^[2].

To build a chatbot, Artificial Intelligence brings Machine Learning and Natural Language Processing into play. Chatbots are used almost everywhere, be it in a banking website, e-commerce store, or any other website, they provide real-time customer service assistance.

What a Chatbot is?

A chatbot is a computer program that is designed to simulate conversations with users via text or speech. It is more like the virtual assistant and they are programmed in such a way that they communicate with each other like humans. The program can be executed using a chat interface or by voice call. There are several ways in which a chatbot can help.

- It can increase operational efficiency
- It can automate customer request
- It can handle basic jobs and instructions
- It has a multi-language support
- It can improve the response rate and the customer engagement

How does a chatbot work?

As discussed above, chatbots are computer programs designed to converse with the user to resolve the query. It has an application layer, a database, and APIs. To simplify the working of a chatbot, it can be said that it works on a pattern matching technique to classify the input and produce a suitable response for it. Chatbots are of different types, depending on the requirements. Mainly there are three types of chatbots, they are as follows:

-
1. **Rule-based chatbots:** a basic chatbot with which a user interacts by using a predefined option. To get the results, the user has to click on certain options, these kinds of chatbots collect the user's request, analyse it and display the result. They replace the frequently asked questions when it comes to complex queries.
 2. **Independent chatbots:** these chatbots analyse what the user wants and respond appropriately. They use customizable keywords and machine learning to determine how to respond to user's queries efficiently and effectively.
 3. **NLP chatbots:** they are the most advanced chatbots till now as they are a combination of rule-based and independent chatbots. They use natural language processing to understand the context and intent in user's requests and thus act accordingly. They can also handle multiple requests.

What is Natural Language Processing?

Natural language processing or NLP for short is defined as the automatic manipulation of natural languages like speech or text. Using NLP, a computer program can communicate with other users. In addition to it, NLP is a branch of artificial intelligence that helps computers in understanding, interpreting, and manipulating human language. It makes it possible for computers to read text, recognize speech, interpret it and measure sentiments of the text.

Basic NLP uses tokenization, parsing, lemmatization, part-of-speech tagging, language detection, transformers for identification and generating accurate results^[3].

In this project, the concepts of Natural Language Processing are used to make an interactive tool. The chatbot takes two types of inputs, that is text and speech, and translates them according to the user's requirement. It also detects whether the sentence is positive or negative. It translates the given sentence into four languages and displays the current language used in the sentence.

1.1 Preliminary

The chatbot can translate the user's text and speech into four different languages which are Hindi, English, German and French. For example: in the beginning, the chatbot will introduce itself as Kratzz and will ask the user to select the type of input. If the user selects 'Text' as input, he will feed the machine a proper sentence which he wishes to translate. Initially, the chatbot will display the language of the sentence then Kratzz will ask the user their preference and will display four language options. Once the user has selected the language, Kratzz will translate the sentence and display it on the UI. It further asks the user whether they want to implement sentiment analysis on the sentence which will classify the sentence into positive or negative.

1.2 Motivation

The areas like Machine Learning and Natural Language Processing are the booming technology that created the foundation and motivated to work on this project. Along with this, the quest to explore new foreign languages sparked the interest to develop this project. The purpose behind working on the chatbot is to freely use this tool to communicate with others. Since the number of Indian graduates wanting to study abroad is increasing, they may find it helpful in their day-to-day lives.

1.3 Problem Definition

This project is developed to resolve the issue of detecting the language and translating it into the language which a user understands. There are times when a person is not able to understand the language used by the other person. In such cases, they can simply use this tool to detect and translate. Along with these features, the chatbot can also analyse the sentiments of the sentence and classify whether it is a positive or negative sentence.

1.4 Objective

The objective of this project is to resolve the communication issue faced by the users by detecting, translating, and implementing sentiment analysis on the text or speech fed by the user. The chatbot 'Kratzz' was made step by step to ensure that it gives accurate results with efficiency.

1. Worked on the functions for detecting, translating, converting speech to text, and implementing sentiment analysis
2. Integration of all the functions to execute the program as per the user demands
3. Development of user interface
4. Integration of the main code with the user interface

Chapter 2- Feasibility Study

This project is about developing a chatbot that has various functions. As the chatbot is based on the concepts of Natural Language Processing, it detects and translates the given sentence. Along with it, the chatbot leverages the use of sentiment analysis to predict the view of the sentence. On the user interface, the chatbot displays the output of the function that has been asked by the user.

This project can have few potential solutions as follows:

- Creating a mobile application of a chatbot like Siri or Bixby so that it becomes easy for a user to access it.
- Developing a web application and executing the chatbot on a local machine.

The possible solutions of this project are evaluated and compared by the following criteria:

- Availability of the resources
- Concept clarity
- Deadline of the project
- Ease of Implementation

After evaluating the possible solutions, the most feasible solution for this project is to build a web application so that a user can interact with the chatbot with a user-friendly interface. While considering the current status of knowledge about the concept, this solution is identified to be the perfect fit.

The main target of this chatbot is to analyse the sentiment of the text, convert it from one language into another and detect the language used. At present, developing a web application is a practical approach to implement the functions.

Chapter 3- Proposed System

- The chatbot is developed for a web application.
- It is user-friendly and easy to access.
- It can work on any local machine by installing a few dependencies in the virtual environment and pulling the code from the repository.
- The chatbot takes the user's input and proceeds accordingly.
- It asks the user about the medium of input whether it is text or speech.
- The chatbot recognizes the speech and converts it into text which then further detects and translates the given input.
- The project focuses on three main functions which are: detecting, translating, and analysing sentiments
- The chatbot analyses whether the sentence is portraying positive or negative emotions.
- The model uses dependencies like:
 - Langdetect: detects the language
 - Google_trans_new: translates the sentence
 - Transformers: analysing sentiments
 - Speech Recognition: to recognize the speech

3.1 Technologies Used

- **Python** is the programming language on which this entire project was coded on. I chose python since it is cross-platform, easy to code, and has many open-source libraries. We can write more complex functions thanks to shorter and simpler syntax, which also allowed me to concentrate on quickly prototyping the idea, rather than focusing on the code itself. Due to the abundance of libraries in python, it saved me countless hours of

coding basic functionality on my own. Some of the libraries that I have used are described below.

- **Pycharm** is one of the best, cross-platform, full-featured, and versatile IDEs for python development. It saves a lot of time with routine tasks. One can access it from the command line. It can connect to a database and create a virtual environment. It provides code analysis, a graphical debugger, an integrated unit tester, integration with a version controlling system^[4]. There are various features of Pycharm listed below:
 - Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
 - Project and code navigation: specialized project views, file structure views, quick jumping between files, classes, methods, and usages
 - Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down, and others
 - Support for web frameworks: Django, web2py, and Flask
 - Integrated Python debugger
 - Integrated unit testing, with line-by-line code coverage
 - Google App Engine Python development
 - Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce, and CVS with change lists and merge
 - Support for scientific tools like matplotlib, numpy and scipy
- **Pyqt5** is the latest version of a GUI widgets toolkit. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI libraries. PyQt5 is a blend of the Python programming language and the Qt library.
- **Googletrans** is a free and unlimited python library that implements Google Translate API. This uses the google translate Ajax API to make calls to such methods as detect and translate. It is compatible with Python 3.6 +. It has updated its translation service with a ticketing mechanism to prevent a lot of crawler programs. Some basic features of googletrans are:
 - Fast and reliable as it uses the same servers that translate.google.com uses

-
- Auto language detection
 - Bulk translations
 - Customizable service URL
 - HTTP/2 Support
 - **Transformers** have been evolved with neural architecture search to perform sequence-to-sequence tasks. The library downloads pre-trained models for Natural Language Understanding (NLU) tasks, such as analyzing the sentiment of a text, and Natural Language Generation (NLG), such as completing a prompt with new text or translating in another language. The easiest way to use a pre-trained model on a given task is to use a `pipeline()`. When the command runs for the first time, a pre-trained model and its tokenizer are downloaded and cached^[5].
 - **Langdetect** supports 55 languages. The algorithm is non-deterministic, which means that if you try to run it on a text which is either too short or too ambiguous, you might get different results every time you run it. This library is a direct port of Google's language-detection library from Java to Python. All the classes and methods are unchanged.
 - **Git** and **Github** were used for source control.

Chapter 4- System Analysis

The entire system is based on natural language processing concepts. This project has a user interface that is developed using PyQt5, though this client interacts with the chatbot.

This NLP-based chatbot is used only for structured data which means that it is highly scripted, it simplifies the programming but it reduces the possible questions a user can ask. Natural Language Processing acts as a fundamental pillar for the recognition of language. It allows technology to recognize human natural text and speech-based commands.

Since this system uses structured data, the chatbot works on keywords that they fetch from the predefined libraries. The quality of the communication depends on how well the libraries are constructed, and the software running the chatbot. NLP makes the chatbot understand input messages and generate appropriate responses. It converts the user's text or speech data into structured data, which is then processed to fetch a suitable answer. After following the process by using built-in functions like detect and translate which uses google translator technology to translate the text. The chatbot responds with the information that is required. The chatbot also determines the emotional tone behind a series of words used to gain an understanding of the attitude, opinions, and emotions expressed.

Figure 1 shows the flowchart of the working of the chatbot. It starts with a small introduction and asks the user whether they want to give text or speech input. Then the bot asks the user to give the sentence which they want to translate and displays the list of languages in which the bot can translate. After translating the sentence, the bot asks the user if they want to run a sentiment analysis on the sentence. If the user selects yes then it will show the result otherwise it will ask the user whether they want to translate further.

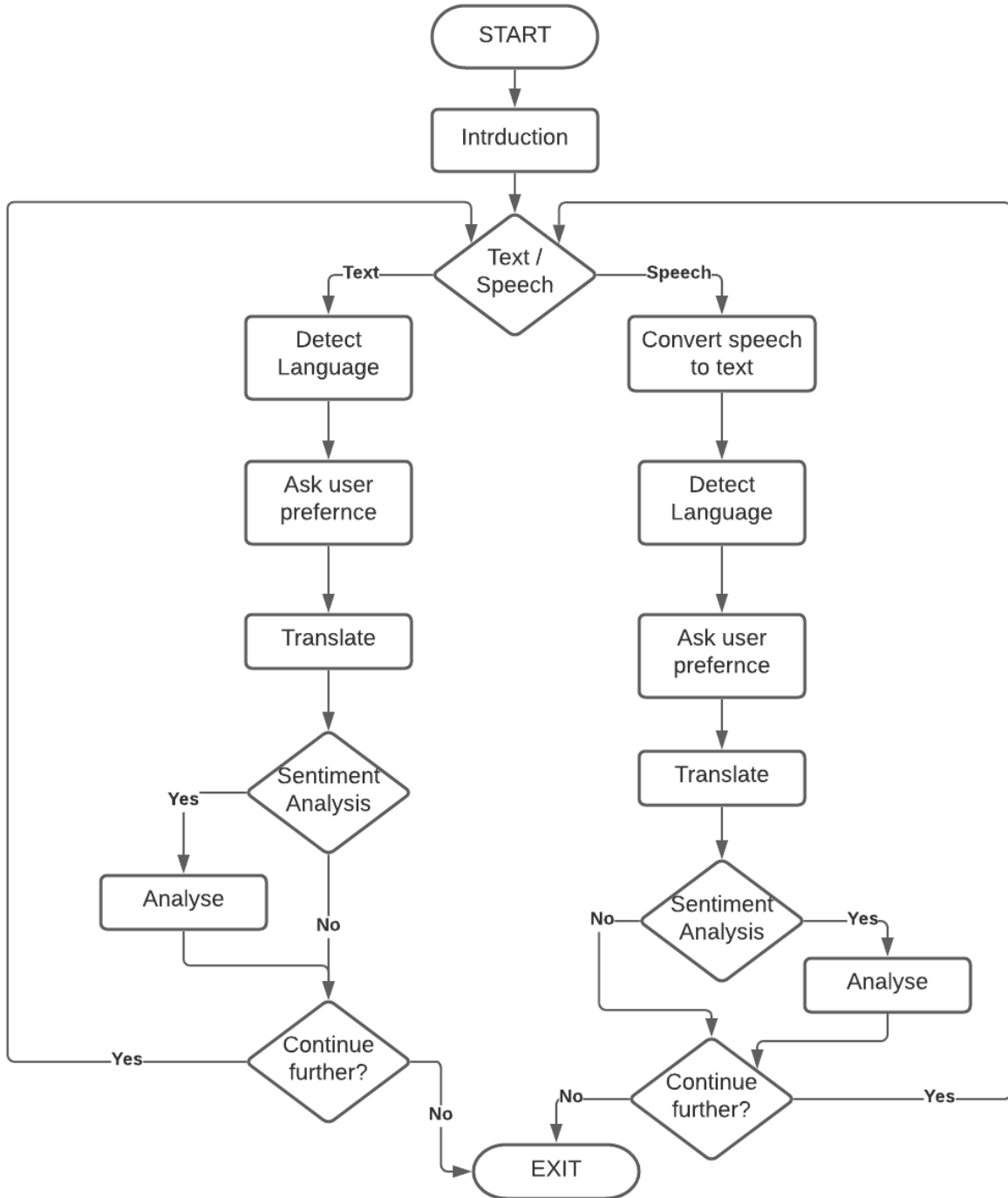


Figure 1: Flowchart of Kratzz chatbot

Chapter 5- Architecture

This chatbot uses Retrieval-based architecture which works on the principle of graphs or directed flows. The responses are based on existing information. These chatbots use techniques like keyword matching, machine learning, and deep learning. The chatbots provide the best possible response from a database of predefined responses ^[6].

Several algorithms and APIs are readily available for developers to build chatbots on this architectural model. This makes the chatbot consider the best and accurate response.

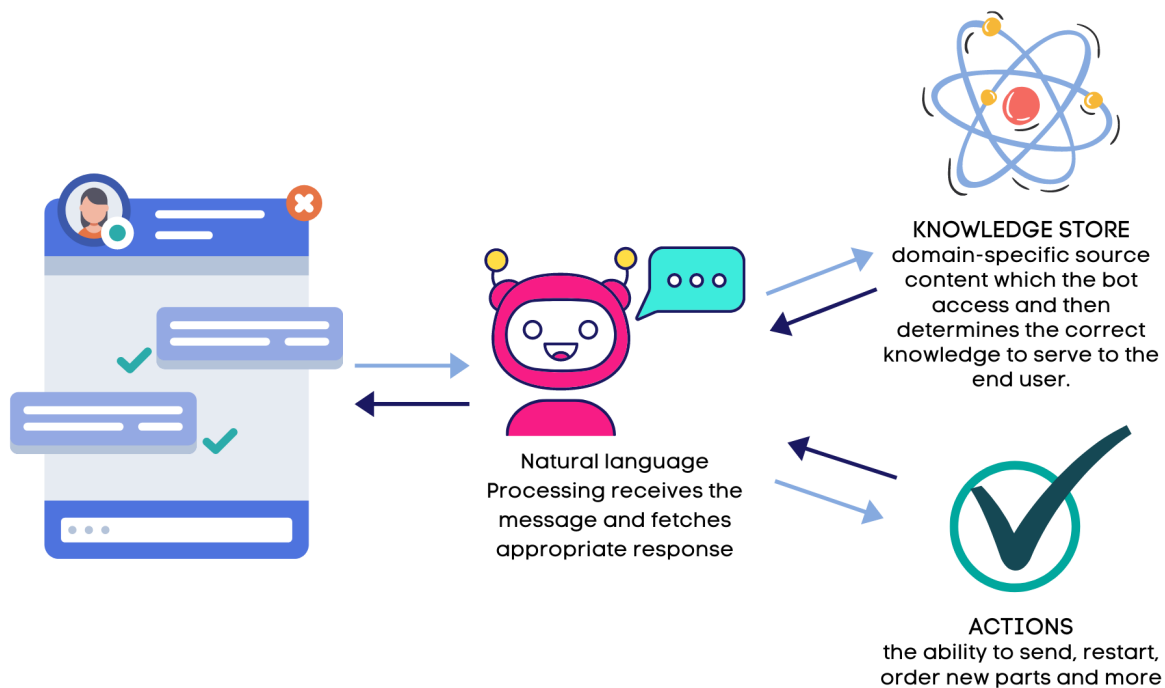


Figure 2: Basic Architecture of Kratzz chatbot ^[7]

Chapter 6- Implementation

The working form of the project is developed during the implementation phase. It involves the building of the targeted project result. Figure 1 shown above explains the flow of the chatbot. It requires defining various functions to get the job done. It has functions like converting speech to text which gets triggered when the user chooses to give a speech input. It uses the microphone as a medium to detect the voice of the user and later convert it into text. Then the other built-in function is there to detect the language of the sentence provided by the user. There is another function that translates the text into the desired language, it uses Google translator API to translate. It also uses a concept of deep learning where it analyses the sentiments of the text using transformers.

6.1 Detecting the Language

To implement the features in a chatbot, every function was tested separately. This function is to detect the language and to do that, install the language detecting dependency by using '*pip install langdetect*'. For testing the library, *from langdetect import detect*. To get the accurate result, one must feed a proper statement to detect the language^[8]. There can be few statements and vocabulary which is common and gets used in various languages, so sometimes the model does not give the desired output. To overcome this, there is another library which tells the percentage of the language detected. In order to use the function, *from langdetect import detect_langs* For instance, in figure 3, code on line number 4, it first imports the library and assigns a string value to a variable. By using *detect_langs*, it tells the possibility of languages. In the given output, the sentence '*hey, how are you?*' detects two possible languages that is english and somali (used in afro-asia). Similarly, it is done for '*die stuhl ist klein*' where the model detects, deutsch (used in Germany) and afrikaans (used in Indo-European).

```
In [1]: pip install langdetect

Requirement already satisfied: langdetect in c:\users\krati\anaconda3\lib\site-packages
Requirement already satisfied: six in c:\users\krati\anaconda3\lib\site-packages
Note: you may need to restart the kernel to use updated packages.

In [1]: from langdetect import detect

s = "ich bin krati."
print(detect(s))

sv

In [2]: g = "Das ist eine Tasse Wasser"
print(detect(g))

de

In [4]: #detect language will find out the probabilities of possible languages
#short text will affect the accuracy

from langdetect import detect_langs
g = "Hey, How are you?"
print(detect_langs(g))
print(detect(g))

[en:0.8571394906676708, so:0.1428578463818774]
en

In [5]: d = "Die stuhl ist klein"
print(detect_langs(d))

[de:0.857141134263975, af:0.14285886573563805]
```

Figure 3: Implementation of detecting language

6.2 Translating the Language

To implement the translating feature, the model uses Google Translator API to produce the accurate rate. For translating the language, install the dependency using *'pip install google_trans_new'*. Then import google_translator to use the library to translate the given sentence.

```
pip install google_trans_new
```

```
Requirement already satisfied: google_trans_new in c:\users\krati\anaconda3\lib\site-pa  
Note: you may need to restart the kernel to use updated packages.
```

```
from google_trans_new import google_translator  
  
t = google_translator()  
s = "die stuhl ist klein aber teuer"  
trans_text = t.translate(s)  
#by default it will translate the sentence into english until and unless specified  
print(trans_text)
```

```
The chair is small but expensive
```

Figure 4: Implementation of translating text function

6.3 Recognizing Speech and Converting into Text

To recognize the speech, '*pip install speechrecognition*' and then import `speech_recognition`. For using the microphone as the source, '*pip install pyaudio*'. The given code will run for 6 seconds and then it will convert the audio into text. The function will display the converted text and the language used. If the user fails to speak then the program will throw Unknown value error and if the function fails to detect the speech, it will throw Request Error^[9]

```
In [1]: pip install speechrecognition

Requirement already satisfied: speechrecognition in c:\users\
Note: you may need to restart the kernel to use updated packa

WARNING: You are using pip version 21.0.1; however, version 2
You should consider upgrading via the 'C:\Users\Krti\anacond

In [2]: import speech_recognition as sr

In [3]: pip install pyaudio

Requirement already satisfied: pyaudio in c:\users\krati\anac
Note: you may need to restart the kernel to use updated packa

WARNING: You are using pip version 21.0.1; however, version 2
You should consider upgrading via the 'C:\Users\Krti\anacond

In [4]: r1 = sr.Recognizer()

with sr.Microphone() as source:
    print('speak now')
    audio = r1.listen(source)
    r1.pause_threshold = 1
    r1.adjust_for_ambient_noise(source)
    audio = r1.listen(source, phrase_time_limit=6)
    try:
        get = r1.recognize_google(audio)
        print(get)
        print(detected(get))
    except sr.UnknownValueError:
        print('error')
    except sr.RequestError as e:
        print('failed'.format(e))
```

Figure 5: Implementing Speech-to-Text function

6.4 Analyse the Sentiment of a Text

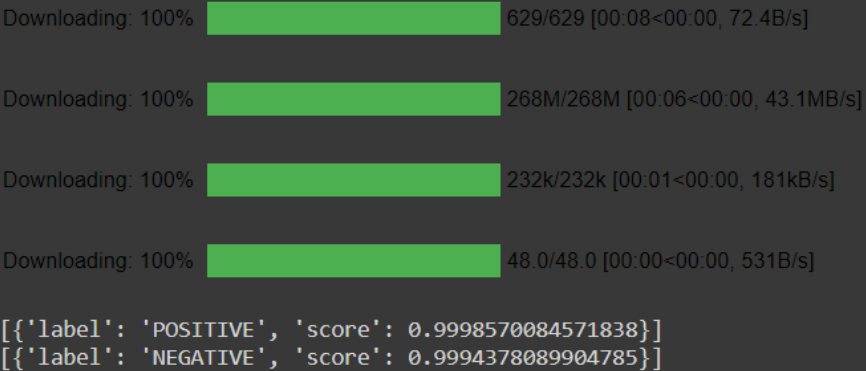
To implement the sentiment analysis on a text, '*pip install transformers*' and from transformers import pipeline. Then specify sentiment-analysis in a pipeline. After executing the program, it will display whether the sentence is positive or negative with the score.

```
[ ] from transformers import pipeline

text1 = "it is a pleasant day, I am going for a walk"
text2 = "I have a terrible headache"

my_model = pipeline('sentiment-analysis')

#pass the starting sequence as input to generate text
print(my_model(text1))
print(my_model(text2))
```



```
Downloading: 100% ██████████ 629/629 [00:08<00:00, 72.4B/s]
Downloading: 100% ██████████ 268M/268M [00:06<00:00, 43.1MB/s]
Downloading: 100% ██████████ 232k/232k [00:01<00:00, 181kB/s]
Downloading: 100% ██████████ 48.0/48.0 [00:00<00:00, 531B/s]

[{'label': 'POSITIVE', 'score': 0.9998570084571838}]
[{'label': 'NEGATIVE', 'score': 0.9994378089904785}]
```

Figure 6: Implementing Sentiment Analysis

When each implementation is implemented successfully, the integration process starts where all the functions are integrated to work according to the sequence.

6.5 Developing user interface using Qt Designer

To develop a GUI for Kratzz, Qt Designer was used. In Qt designer, one can directly design the user interface through the designer or the other option is to hand code the GUI in plain python code. Qt is a tool to create a user interface by dragging and dropping the widgets on an empty form. The objects can be arranged into the GUI using different layouts.

It is a platform that is programming language independent and it creates a .ui file. The content of these files can be changed using python code by importing uic from pyqt5.

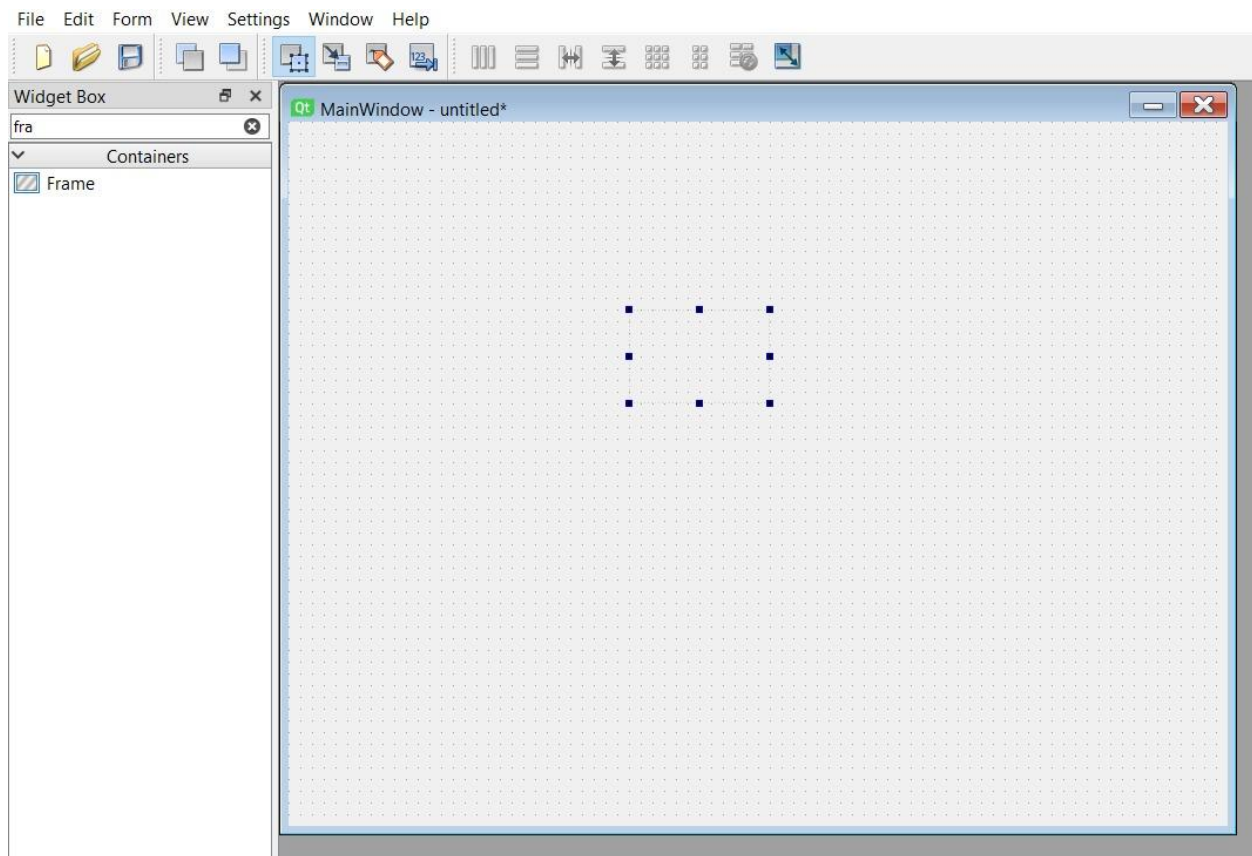


Figure 7: Graphical User Interface of Qt Designer

Frames are the basic foundation of designing a UI through Qt. Through extensive manipulation of any number of frames, one can get really well designed UI's

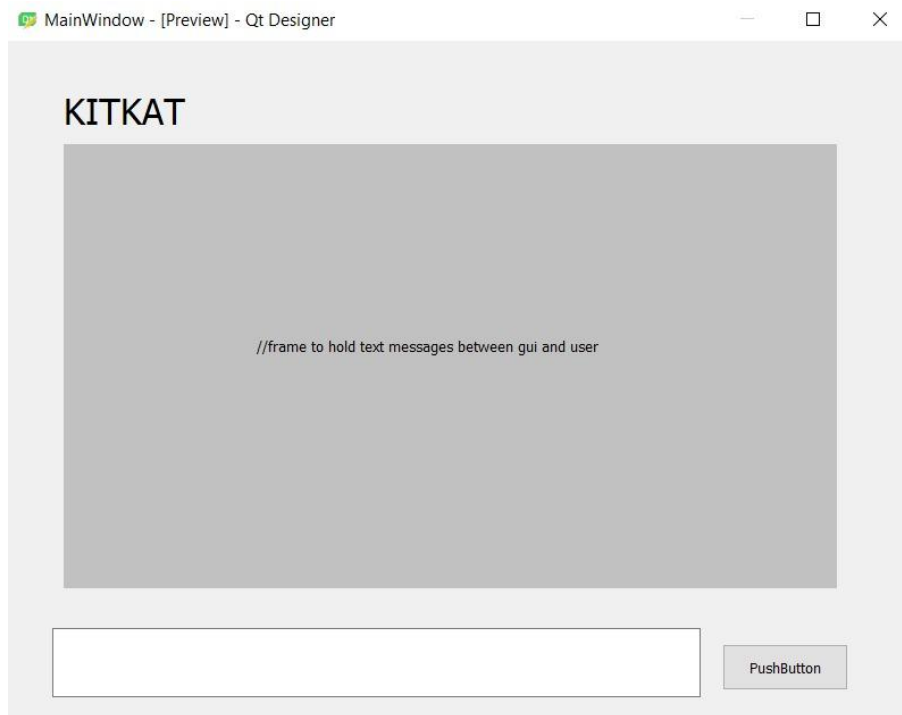


Figure 8: Early draft of UI

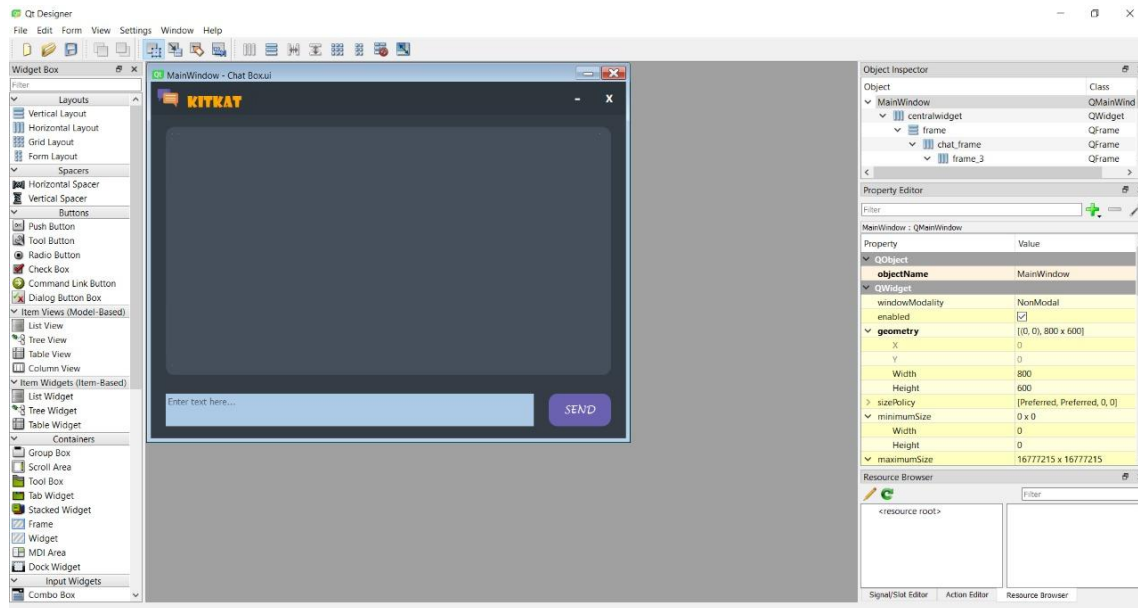


Figure 9: Basic UI design

Chapter 7- Testing

This phase investigates and examines the progress of a given project to provide information about the actual level and quality of the project. It evaluates and tests the declared requirements, features, and expectations regarding the project before its delivery to ensure the project matches the initial requirements stated.

In this project, the chatbot was manually tested and was given the wrong inputs to check the result. As in the code, I have already declared that whenever the wrong input is given, it will show a message saying “I couldn’t understand...”

The code ensures that the textbox is not case sensitive as it converts each input into a lower case.

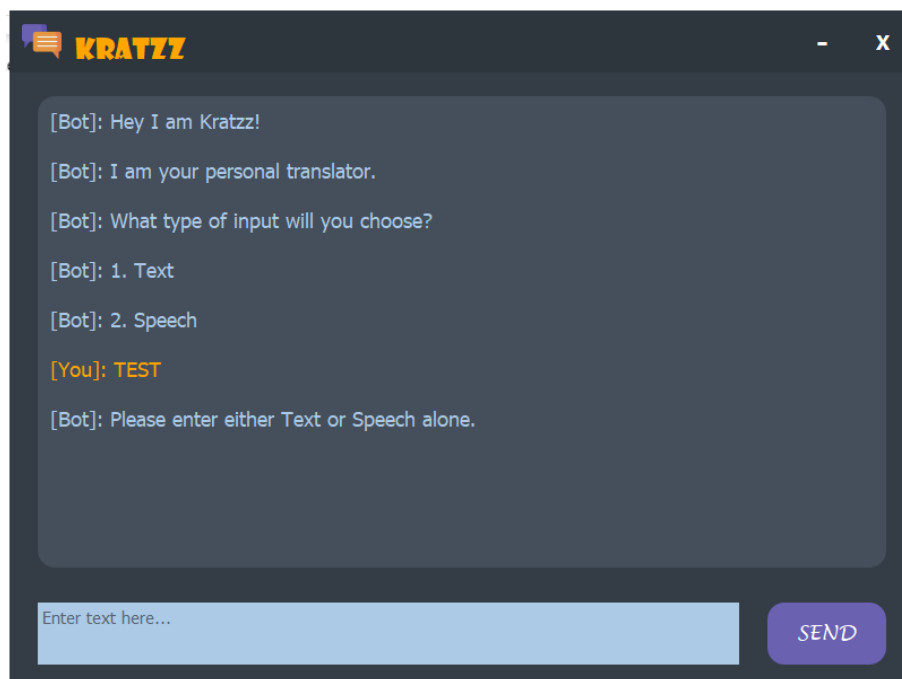


Figure 10: Displays message when the input is incorrect

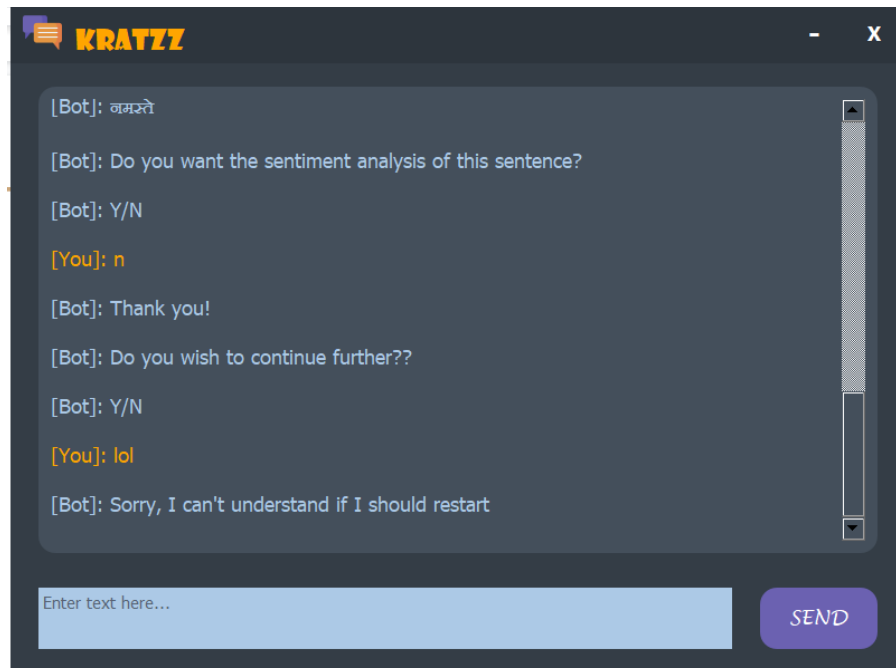


Figure 11: Displays the incorrect message



Figure 12: Displays wrong output due to short sentence



Figure 13: Displays try again when the language is fed incorrectly



Figure 14: The user select hindi as a preferred language

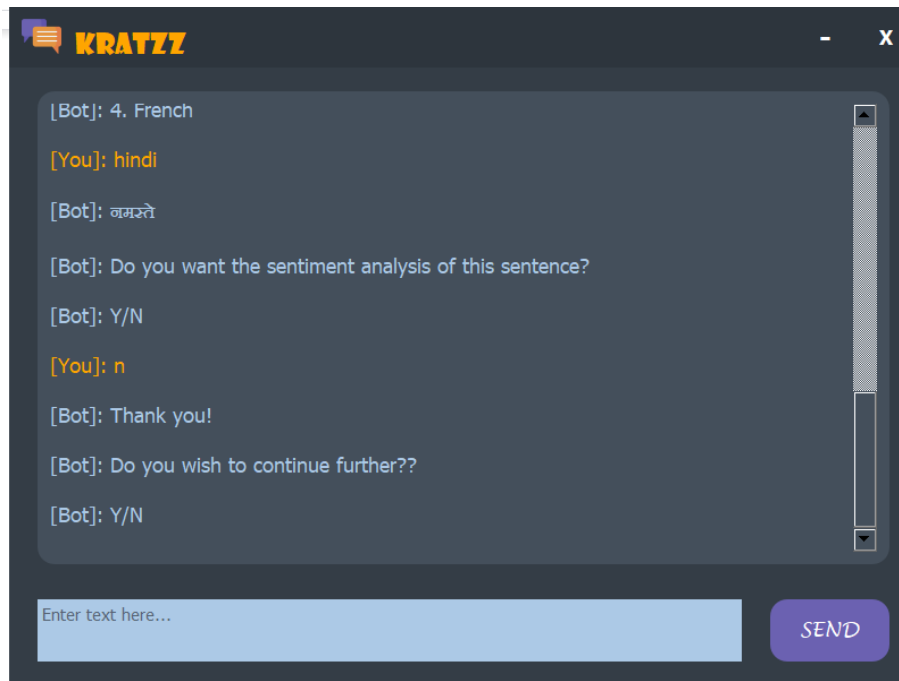


Figure 15: If the user selects ‘N’ for the sentiment analysis

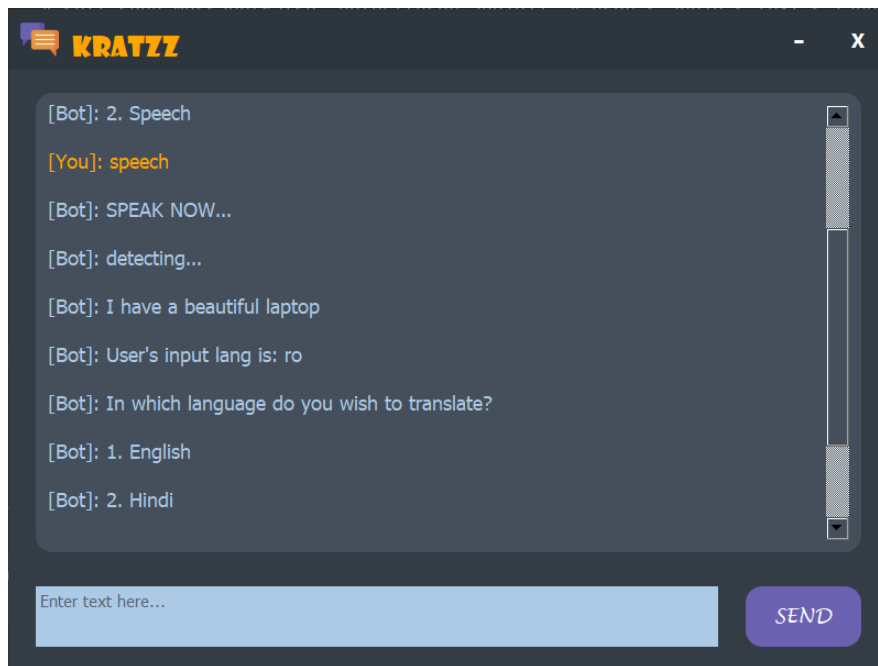


Figure 16: If the user selects speech as an input medium

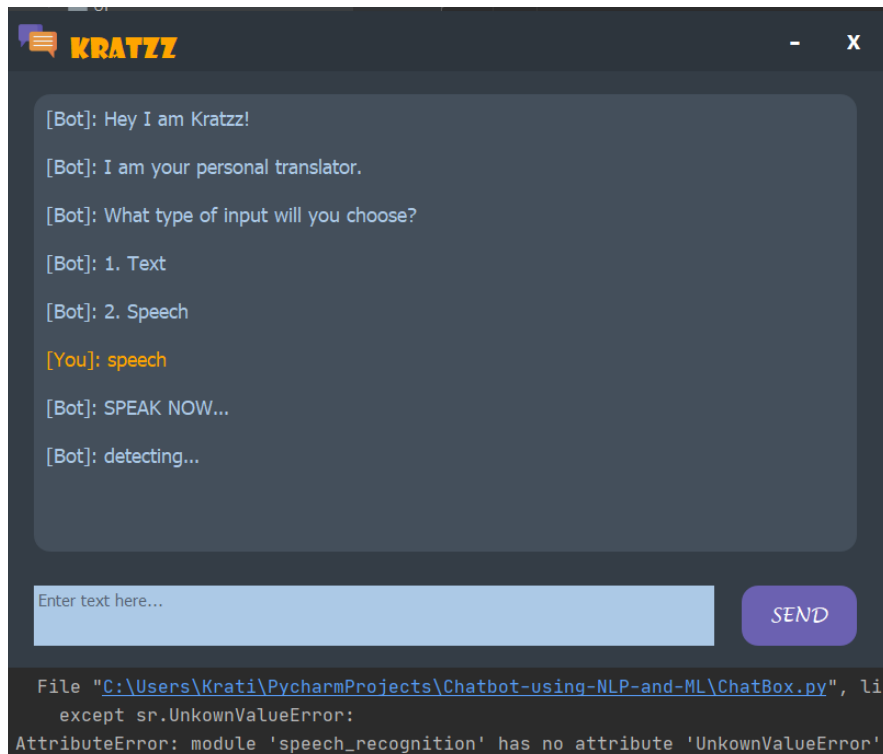


Figure 17: Throws error

Chapter 8- Future Scope & Conclusion

To conclude the whole project, the user interface was successfully designed and was made user-friendly with a minimalistic approach. The functionality of a chatbot is working properly with accuracy. It is detecting and translating the sentences with accuracy of 95%. The sentiment analysis function gives accurate results. It is easy for the user to understand how the chatbot proceeds as it asks for input from the user after every step.

The scope of Kratzz can be enhanced by adding the following functionalities:

- Normal conversation with the chatbot by implementing machine learning
- More accurate result when detecting a single word
- Recognizing speech commands
- Having an option to upload audio files and then translating it
- Giving voice outputs
- Converting a web application to an android application

References

1. By: IBM Cloud Education. (n.d.). What is Artificial Intelligence (AI)? IBM. <https://www.ibm.com/in-en/cloud/learn/what-is-artificial-intelligence>.
2. What is Artificial Intelligence? How Does AI Work?: Built In. What is Artificial Intelligence? How Does AI Work? | Built In. (n.d.). <https://builtin.com/artificial-intelligence>.
3. What is Natural Language Processing? SAS. (n.d.). https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html.
4. Real Python. "PyCharm for Productive Python Development (Guide)." *Real Python*. Real Python, 23 Jan. 2021. Web. 04 May 2021.
5. Cheng, Raymond. "Sentiment Analysis with Pretrained Transformers Using Pytorch." *Medium*. Towards Data Science, 20 Jan. 2021. Web. 04 May 2021.
6. Fainchtein, Leah. "Generative vs Retrieval Based Chatbots: A Quick Guide." *Medium*, CloudBoost, 2 Dec. 2020,
7. "How To Build A Chatbot: End-to-End Guide." *Intellectsoft Blog*. 16 July 2020. Web. 04 May 2021.
8. David, Davis. "How to Detect and Translate Languages for NLP Project." *Medium*, Towards Data Science, 9 Jan. 2021, towardsdatascience.com/how-to-detect-and-translate-languages-for-nlp-project-dfd52af0c3b5.
9. "Using the Speech-to-Text API with Python | Google Codelabs." *Google*, Google, codelabs.developers.google.com/codelabs/cloud-speech-text-python3#6.