

# Language Translator

## Introduction

In this project we are building a neural machine translator using English as the source language and target language was Hindi . We will be using Long Short Term Memory(LSTM) units in keras. You will learn how to:

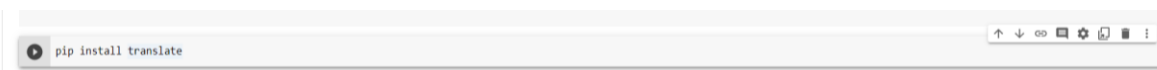
## Project File Structure:

Steps to develop Python Language Translator with GUI:

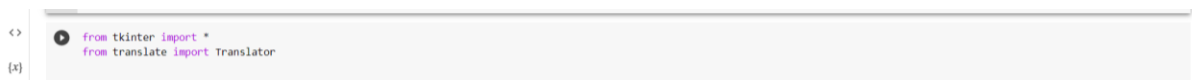
1. Installing translate
2. Importing translate and tkinter
3. Initializing window
4. Creating tuple for choosing languages
5. Creating a function for translating the text
6. Choice for input language and the language in which the text is to be translated
7. Input and Output text

## 1. Installing translate:

Installation of this package is required before starting the project. Translation of major languages is provided by this package. To install this package run the following command on the command prompt or terminal window.



## 2. Importing translate and tkinter:



## Code Explanation:

- **From tkinter import \*:** The easiest and most effective way to develop GUI applications is using tkinter. \* means importing everything from tkinter.
- **From translate import Translator:** This package helps to translate major languages.
- 

## 3. Initialising Window

```
• Screen = tk();  
Screen.title("Language Translator with GUI ")  
  
InputLanguageChoice = StringVar()  
TranslateLanguageChoice = StringVar()
```

## Code Explanation:

InputLanguageChoice is a variable that stores the language of the text that is being translated.

TranslateLanguageChoice is a variable that stores the language in which the text is to be translated.

- **Tk():** Root window is created with the help of Tk() class.
- **title():** It is used to display the title on the top of the root window.

## 4. Creating tuple for choosing language:

```
• LanguageChoices = ('Hindi','English','French','German','Spanish')  
InputLanguageChoice.set('English')  
TranslateLanguageChoice.set('Hindi')
```

## Code Explanation:

LanguageChoices is a tuple that stores five languages from which input language and translated language can be chosen.

- **set():** It helps to set the value of the variable. For instance: value of InputLanguageChoice is set as English initially which can be changed afterwards and selected from the tuple.

## 5. Creating a function for translating the text:

```
[ ] def Translate():
    translator = Translator(from_lang= InputLanguageChoice.get(),to_lang=TranslateLanguageChoice.get())
    Translation = translator.translate(TextVar.get())
    OutputVar.set(Translation)
```

### **Code Explanation:**

This function is created to translate the text. OutputVar is a variable that stores the translated text. TextVar is a variable that contains the text that is to be translated.

- **Translator():** It helps to translate the text.
- **from\_lang :** It is the language of the text that is being translated.
- **to\_lang:** It is the language of the text in which the text is to be translated.
- **get():** Value of the item is returned with the help of this method.

## 6.Choice for input language and the language in which the text is to be translated:

```
<>
(x) #choice for input language
    InputLanguageChoiceMenu = OptionMenu(Screen,InputLanguageChoice,"Languagechoices")
    Label(Screen,text="Choose a Language").grid(row=0,column=1)
    InputLanguageChoiceMenu.grid(row=1,column=1)

#choice in which the language is to be translated
    NewLanguageChoiceMenu = OptionMenu(Screen,TranslateLanguageChoice,"Languagechoices")
    Label(Screen,text="Translated Language").grid(row=0,column=2)
    NewLanguageChoiceMenu.grid(row=1,column=2)
```

InputLanguageChoiceMenu provides a choice of input languages. NewLanguageChoiceMenu provides a choice of languages in which translation of text is possible.

- **OptionMenu():** It provides the options that are available to the user.
- **Label():** This widget helps to implement display boxes where the text can be placed.
- **grid():** Widgets are arranged on the screen with the help of grid.

## 7.Input and Output text :

```
[ ] Label(Screen,text="Enter Text").grid(row=2,column =0)
    TextVar = StringVar()
    TextBox = Entry(Screen,textvariable=TextVar).grid(row=2,column = 1)

    Label(Screen,text="Output Text").grid(row=2,column =2)
    OutputVar = StringVar()
    TextBox = Entry(Screen,textvariable=OutputVar).grid(row=2,column = 3)

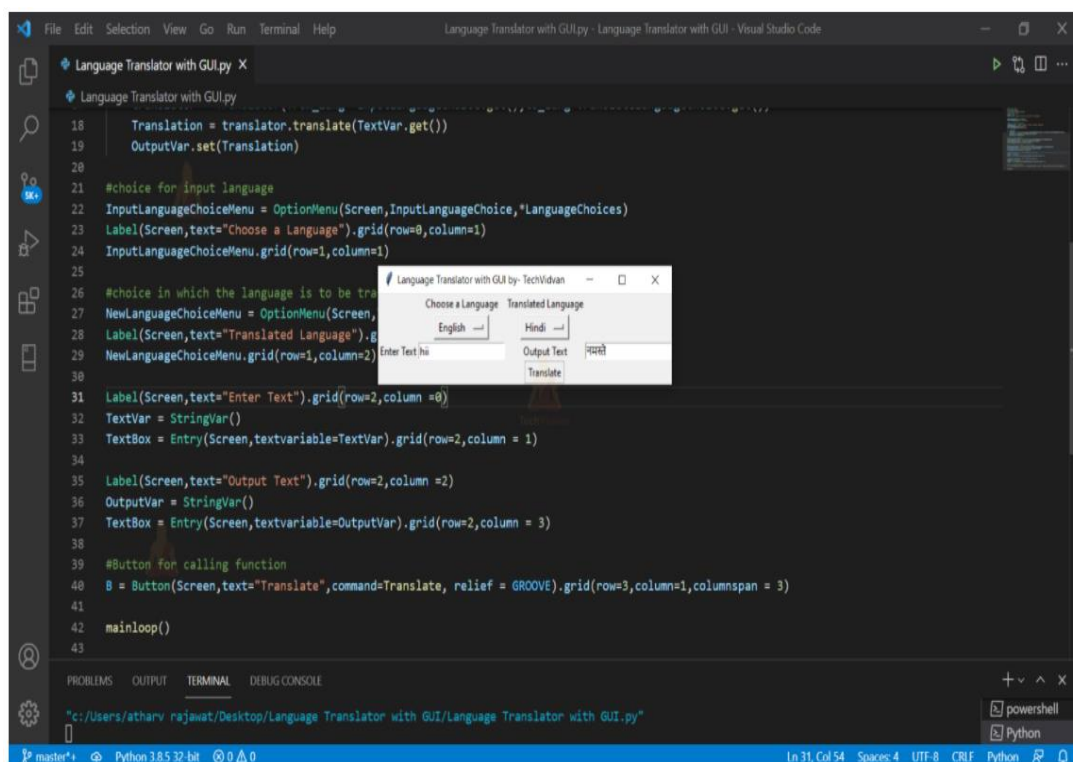
#Button for calling function
    B = Button(Screen,text="Translate",command=Translate, relief = GROOVE).grid(row=3,column=1,columnspan = 3)

    mainloop()
```

### Code Explanation:

- **Entry():** This widget helps to enter or display a single line text on the screen.
- **Button():** This widget creates a button.
- **relief:** It helps to provide 3-D effects around the outside of the widget.
- **mainloop():** It helps to run the tkinter event loop.

### FINAL OUTPUT:



### Summary:

I have successfully developed python language translator project with translate and tkinter. This is an interesting python project for beginners, based on requirements you can add more functionalities.