LANGUAGE TANSLATOR

Python Project - EISYSTEMS

TO
MR. MAYUR DEV SEWAK &
MS. MALLIKA SRIVASTAVA

BY
SAI NISHITHA DANGETI

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO.
1	ABSTRACT	2
2	PROJECT SUMMARY	3
3	OBJECTIVES OF PROJECT	4
4	INTRODUCTION	4-5
5	MOTIVATION	5-6
6	PLAN OF RESEARCH	7
7	METHODLOGY	7-9
8	DETAILS OF PROCESS	10-12
9	SYSTEM REQUIREMENTS	13
10	DATA FLOW DIAGRAMS	14
11	INPUT/OUTPUT SCREENSHOTS	15-16
12	SOURCE CODE	17-18
13	REFERENCES	19

ABSTRACT OF PROJECT:

Translation is the communication of meaning from one language (the source) to another language (the target). Translation refers to written information, whereas interpretation refers to spoken information.

The purpose of translation is to convey the original tone and intent of a message, taking into account cultural and regional differences between source and target languages.

Translation has been used by humans for centuries, beginning after the appearance of written literature. Modern-day translators use sophisticated tools and technologies to accomplish their work, and rely heavily on software applications to simplify and streamline their tasks.

Organizations around the world, encompassing a multitude of sectors, missions, and mandates, rely on translation for content as diverse as product labels, technical documentation, user reviews, promotional materials, annual reports, and much, much more.

Text translation from one language to another is increasingly becoming common for various websites as they cater to an international audience. The python package which helps us do this is called translate.

PROJECT SUMMARY:

All people on earth are divided by their cultures, and language is a particular feature that makes us so different from each other. Speaking the same language, we can easily share our ideas and emotions. We use language when we talk, and when we write. Various languages represent different cultures, and ways of thinking. We can say that, many years ago, people faced much less problems with communication, since they didn't need to talk to somebody out of their community. Developing new communication devices, we make people from all over the world closer to each other. Now we have to talk to people from other countries much more often than even twenty years ago. Thus, problems caused by different languages arise more often, so we have to adapt, and learn new languages, to better understand each other. These problems are caused not only by different vocabularies, but also by different grammatical structures, and spellings. Translation is a process that helps people overcome such problems. When we translate a text, we transfer the meaning of a certain phrase, or word. This meaning is transferred from a source language to a target language. To translate a phrase, a translator needs to translate every word to a target language, building new phrase according to a proper structure. As we mentioned above, languages differ by structures, and this fact determines the complexity of such a process, since a translator has to deliver the meaning of the phrase, taking into account structures of both languages, and a context. Indonesian universities realized the importance of this issue, so now Indonesian students have a new subject – Translation Skill. Now University of Jakarta is focused on Indonesian – English translations. Googletrans is a free and unlimited python library that implemented Google Translate API. This uses the Google Translate Ajax API to make calls to such methods as detect and translate. The translation is done with the Translator's translate() method. If we do not specify the source and the destination languages, googletrans tries to detect the language and translates it into English. We translate a Russian text. We get a translated object.

OBJECTIVES OF PROJECT

Language Translator using Google Translate APIs in Python – Instantly Translate texts, words, phrases from one language to another.

Translation enables communication between people from different regions. It provides meaningful communication from one language to another language.

About Language Translator

A language translator or text translator is a tool to translate text, words, phrases from one language to any other language. It is like a dictionary where we can translate the text.

The objective of this project is to translate text from one language to any other language in real-time with a button click. This project will be built using the Tkinter, googletrans libraries.

In this project, the user enters text in any language and get it translated in any other language by selecting the output language.

INTRODUCTION:

API stands for Application Programming Interface. It acts as an intermediate between two applications or software. In simple terms, API acts as a messenger that takes your request to destinations and then brings back its response for you. Google API is developed by Google to allow communications with their servers and use their API keys to develop projects.

In this tutorial, we are going to use Google API to build a language Translator which can translate one language to another language. On the internet, we can see lots of projects on Speech Recognitions, Speech to text, text to speech, etc. but here in this project we are going to build something more advance than that.

Let's assume a scenario, we are traveling in Spain and we don't know how to speak Spanish or we are in any other country and we don't know their native language, then we can use this tool to overcome the problem. We can translate between all those languages which are present in google translator.

Python provides various options for developing graphical user interfaces (GUIs). Most important are listed below.

- Tkinter − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.
- wxPython − This is an open-source Python interface for wxWindows http://wxpython.org.
- JPython − JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine http://www.jython.org.

There are many other interfaces available, which you can find them on the net.

Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

MOTIVATION:

Translation Studies is a field of study that deals with the theory, description, and application of translation. Because it examines translation both as an interlingual transfer, and as an intercultural communication, Translation Studies can also be described as an inter-discipline which touches on other diverse fields of knowledge, including comparative literature, cultural studies, gender studies, computer science, history, linguistics, philosophy, rhetoric, and semiotics.

The skills of translation are becoming ever more important and desirable. Today's multicultural and multilingual society demands effective, efficient, and empathetic communication between languages and cultures. That's important for various reasons, which we'll now explore.

Not Everybody Speaks English

English is the most prominent language in the world. As a result, one might question the importance of translation, and ask, *why doesn't everybody just speak English*?

The reality, however, is that not everybody *can* speak English, fewer still are able to speak it well enough to communicate effectively, and perhaps even more importantly: language is much more than the communication of words. It is also an expression of culture, society, and belief. Promoting a universal language, therefore, would likely lead to a loss of the culture and heritage communicated through native languages.

It Enables A Global Economy

As communication and travel advance, geography is becoming less and less of a barrier to doing business. Companies benefit from working overseas. They can take advantage of the

lower cost of products and services in some countries, the professional and industrial expertise of others, and additional markets to trade in.

When they trade in countries with a different native language, they need high-quality translation to communicate effectively. When there's a demand for translation there are opportunities for translators. When there's a demand for translators, there's a demand for Translation Studies. They need to learn the skills to practice at a high level, and perhaps even contribute to advancing the field even further.

Looking ahead, whilst English is the world's most prominent language at the moment, it may not always be. When a market emerges and grows rapidly, like the Chinese market has in recent years, the demand for translation to and from its native language is also likely to increase.

The Spread of Information and Ideas

Translation is necessary for the spread of information, knowledge, and ideas. It is absolutely necessary for effective and empathetic communication between different cultures. Translation, therefore, is critical for social harmony and peace.

Translation is also the only medium through which people come to know different works that expand their knowledge. For example:

- Arabic translators were able to keep the ideas of ancient Greek philosophers alive throughout the Middle Ages
- The Bible has been translated into at least 531 languages
- Translation is helping sports teams and organisations overcome language barriers and transcend international boundaries
- TED Talks run open translation projects that allow people around the world to understand their talks, offering non-English speakers to learn from some of the best educators in the world.

The Role of Translation Studies

Effective, efficient, and empathetic translation requires highly skilled practitioners. Courses in Translation Studies are a great way for linguists, language graduates, and translators to develop a deep understanding of the academic field, and the skills to practice as a translation professional.

Translation enables effective communication between people around the world. It is a courier for the transmission of knowledge, a protector of cultural heritage, and essential to the development of a global economy. Highly skilled translators are key. Translation Studies helps practitioners develop those skills.

PLAN OF RESEARCH:

Unless you have been hiding under a rock, you have probably used Google Translate on many occasions in your life. Whenever you try to translate a word or a sentence from a certain language to another, it is the Google Translate API which brings you the desired results in the background. Though you can translate anything by simply going to the Google Translate web page, you can also integrate Google Translate API into your web applications or desktop programs. The best thing about the API is that it is extremely easy to set up and use.

You can actually do a lot of things with the help of the Google Translate API ranging from detecting languages to simple text translation, setting source and destination languages, and translating entire lists of text phrases. In this article, you will see how to work with the Google Translate API in the Python programming language.

METHODOLOGY:

Google trans package contains many functionalities, but here we are discussing about googletrans translate functionality.

Basic Use: First, we need to create Translator class object.

```
# Language Translator
from googletrans import Translator  # Import Translator module from googletrans
package

translator = Translator() # Create object of Translator.
```

Now call the translate API to translate the source string, if we do not define source language, Google translate auto detects the source language and translates string into English by default.

```
translated = translator.translate('안녕하세요')

# Source language auto detect by google trans
# By default destination language is English
```

Return value of translate API is a Translated class object, which has the following member variables.

- src – source language (default: auto)
- dest – destination language (default: en)
- origin – original text
- text – translated text
- pronunciation – pronunciation

To access the translated string and source language, access member variables of translated object.

```
print(" Source Language:" + translated.src)
# Output: Source Language: ko

print(" Translated string:" + translated.text)
# Output: Translated string: Good evening
```

We can also provide the source and destination language in translate API.

```
translated = translator.translate('안녕하세요', src='ko') # Pass only source
language
translated = translator.translate('안녕하세요', dest='en') # Pass only destination
language
translated = translator.translate('안녕하세요', src='ko', dest='en') # Pass both
source and destination
```

We can also get the pronunciation of the source string.

```
translated = translator.translate('안녕하세요', src='ko', dest='ja')

print(" Source Language:" + translated.src)
# Output: Source Language: ko

print(" Translated string:" + translated.text)
# Output: Translated string: こんにちは

print(" Pronunciation:", translated.pronunciation)
# Output: Pronunciation: Kon'nichiwa
```

Bulk request: We can pass list of source string to translate in destination language.

```
translatedList = translator.translate(['Hello Friends','Welcome on Codeproject',
'Have a good day'], dest='ja')

for translated in translatedList:
          print(translated.origin, '->', translated.text)

# Output:
# Hello Friends -> 皆さん、こんにちは
# Welcome on Codeproject -> Codeprojectへようこそ
# Have a good day -> 良い一日を過ごしてください
```

Detect Language: Google trans also provides detect API, as its name implies, identifies the language also provide the confidence.

```
detected = translator.detect(' 皆さん、こんにちは')
```

Return value of detect API is a Detected class object, which has the following member variables.

- lang – detected language
- confidence – the confidence of detection result (0.00 to 1.00)

To access the detected language and the confidence, perform:

```
print('Detected Language:', detected.lang, ' with confidence: ',
detected.confidence)

# Detected Language: ja with confidence:  1
```

DETAILS OF PROCESS:

- Create a display window
- Create input and output text widget
- Define Combobox to select a language
- Define function
- Create a translate button

1. Import Modules

```
from tkinter import *
from tkinter import ttk
from googletrans import Translator, LANGUAGES
```

We import ttk modules from tkinter library and Translator, LANGUAGES modules from googletrans library.

2. Create a display window

```
root = Tk()
root.geometry('1080x400')
root.resizable(0,0)
root.config(bg = 'ghost white')
```

We use tkinter library to create a window where we'll enter the text which we want to convert into voice.

- Tk() initialized tkinter which means window created
- geometry() set the width and height of the window
- resizable(0,0) set the fixed size of the window
- bg = '' use to set the background color

```
root.title("Language Translator")
Label(root, text = "LANGUAGE TRANSLATOR", font = "arial 20 bold", bg='white smoke').pack()
Label(root,text ="Project Gurukul", font = 'arial 15 bold', bg ='white smoke' , width = '20').pack(side = 'bottom')
```

- title() used to set the title of the window
- Label() widget use to display one or more than one line of text that users aren't able to modify.

- root is the name which we refer to our window
- text which we display on the label
- font in which the text is written
- pack organized widget in block

3. Create an Input-output text widget

```
Label(root,text ="Enter Text", font = 'arial 13 bold', bg ='white smoke').place(x=200,y=60)

Input_text = Text(root,font = 'arial 10', height = 11, wrap = WORD, padx=5, pady=5, width = 60)

Input_text.place(x=30,y = 100)

Label(root,text ="Output", font = 'arial 13 bold', bg ='white smoke').place(x=780,y=60)

Output_text = Text(root,font = 'arial 10', height = 11, wrap = WORD, padx=5, pady= 5, width =60)

Output_text.place(x = 600 , y = 100)
```

The above code creates two text widgets one for entering text and the other for displaying translated text.

- Text() widget is used for multiple text lines.
- wrap = WORD will stop the line after the last word that will fit.
- padx puts an extra bit of space to the left and right of the widget.
- pady adds an extra bit of space to the top and bottom.

4. Define Combobox to select the language

```
language = list(LANGUAGES.values())

src_lang = ttk.Combobox(root, values= language, width =22)

src_lang.place(x=20,y=60)

src_lang.set('choose input language')

dest_lang = ttk.Combobox(root, values= language, width =22)

dest_lang.place(x=890,y=60)

dest_lang.set('choose output language')
```

From the above code, users can pick a seperate language for both input data and to translate their data.

- language gets all the values from the ‘LANGUAGES’ dictionary in the form of a list.

- ttk.Combobox() widget is a class of ttk modules. It is a drop-down list, which can hold multi-value and show one item at a time. Combobox is useful to select one option from many option.

5. Define Function

```
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)
```

The Translate function will translate the message and give the output.

- src gets the language selected as input text language
- dest gets the language select to translate
- text gets the input text entered by the user.”1.0” means that the input should be read from zero characters to line one
- The END part means to read the text until the end is reached
- translator = Translator() used to create a Translator class object
- Output_text.delete(1.0, END) delete all the text from line one to end
- Output_text.insert (END, translated.text) will insert the translated text in Output_text

6. Create a translate button

```
trans_btn = Button(root, text = 'Translate',font = 'arial 12 bold',pady = 5,command = Translate , bg = 'royal blue1', activebackground = 'sky blue')

trans_btn.place(x = 490, y= 180 )

root.mainloop()
```

When we click on the Translate button it will call the translate function

Button() widget used to display button on our window

- command is called when we click the button
- activebackground sets the background color to use when the button is active

root.mainloop() is a method that executes when we want to run our program.

SYSTEM REQUIREMENT:

- Operating system.
- Minimum CPU or processor speed.
- Minimum GPU or video memory.
- Minimum system memory (RAM)
- Minimum free storage space.

To implement this project, we will use the basic concepts of Python, Tkinter, and googletrans libraries.

Tkinter is a standard GUI Python library. ttk module gives access to the Tk themed widget set.

googletrans is a module to translate text. We import the Translator from googletrans, which is used to do translations. We also import LANGUAGES from googletrans which lists all supported languages in a Python dictionary.

To install the library, use pip install to the command prompt:

```
pip install tkinter

pip install googletrans
```

DATA FLOW DIAGRAMS:

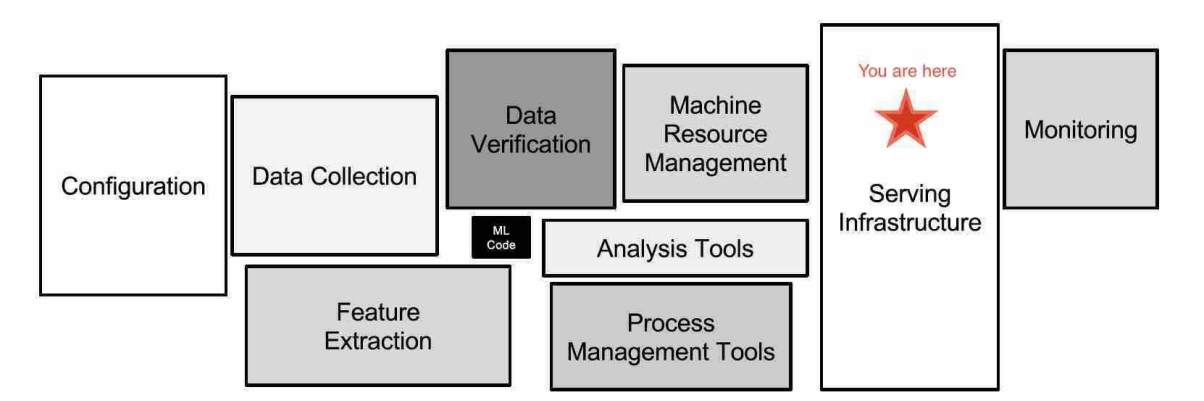


Figure 1- Google translate API in python

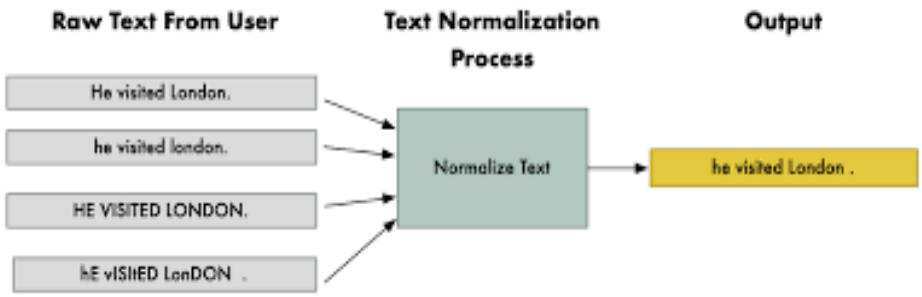


Figure 2 – Functionality of google trans API

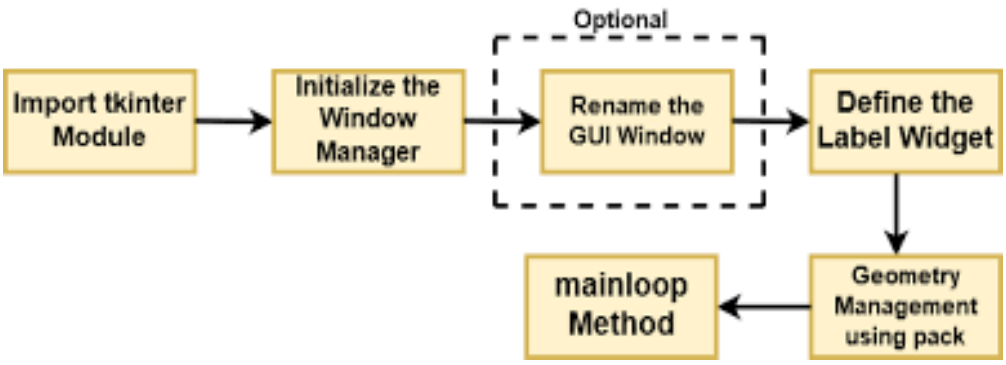
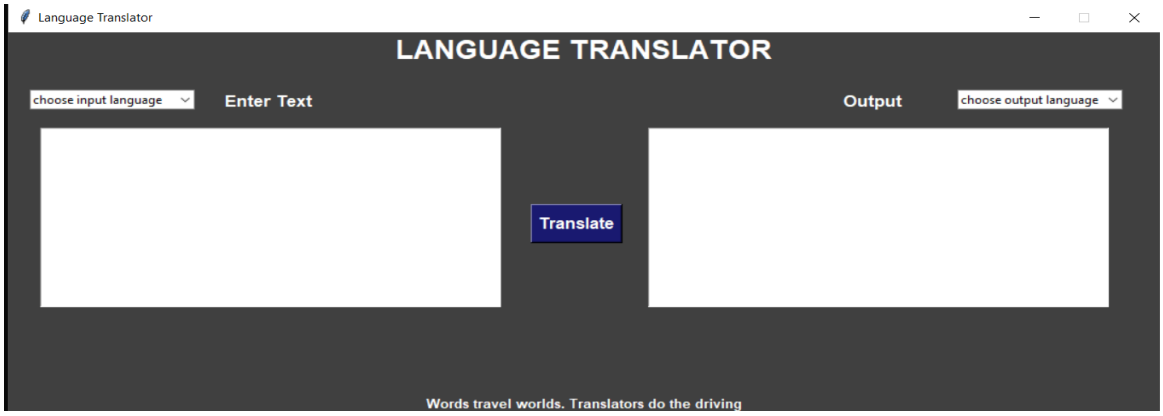


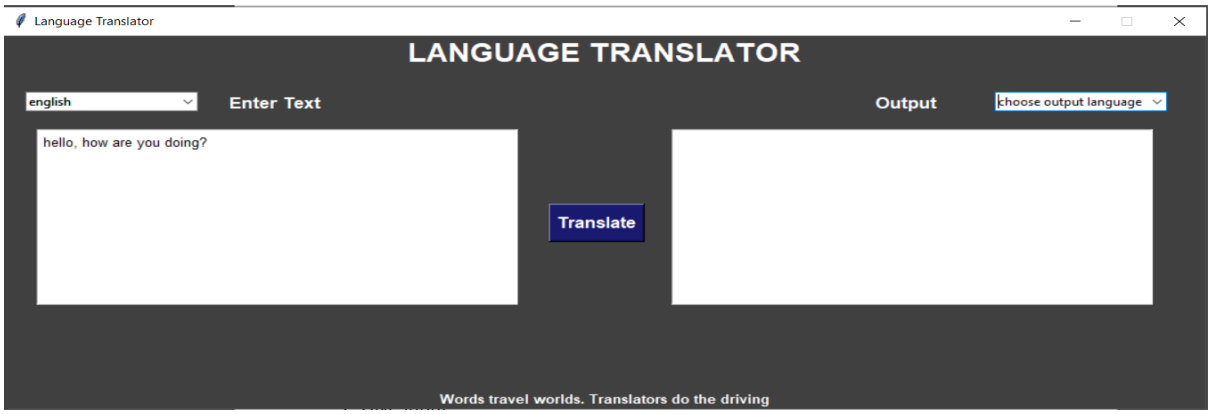
Figure 3- Functionality of tkinter

INPUT / OUTPUT SCREENSHOTS:

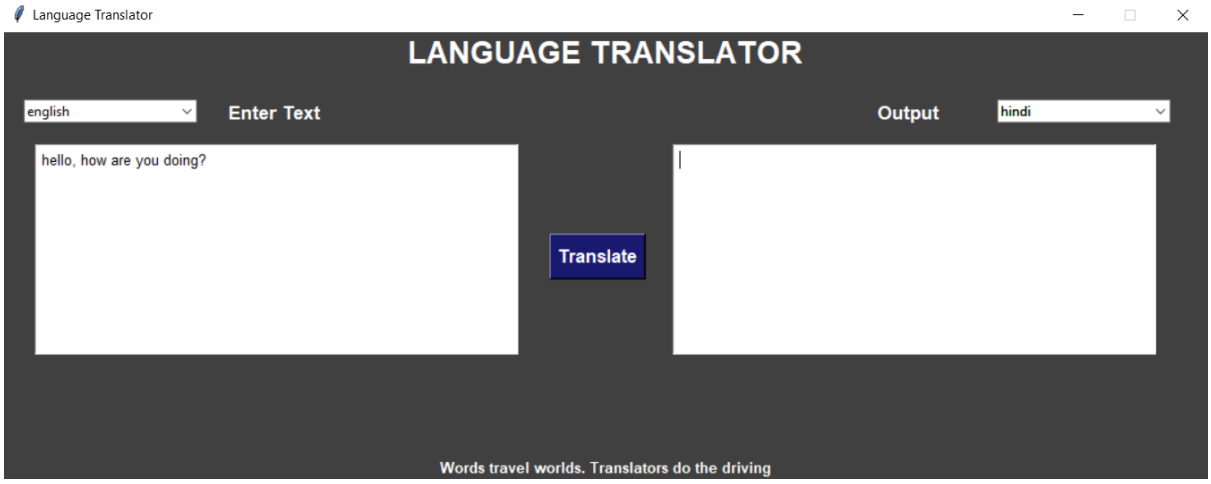
1. opening window:



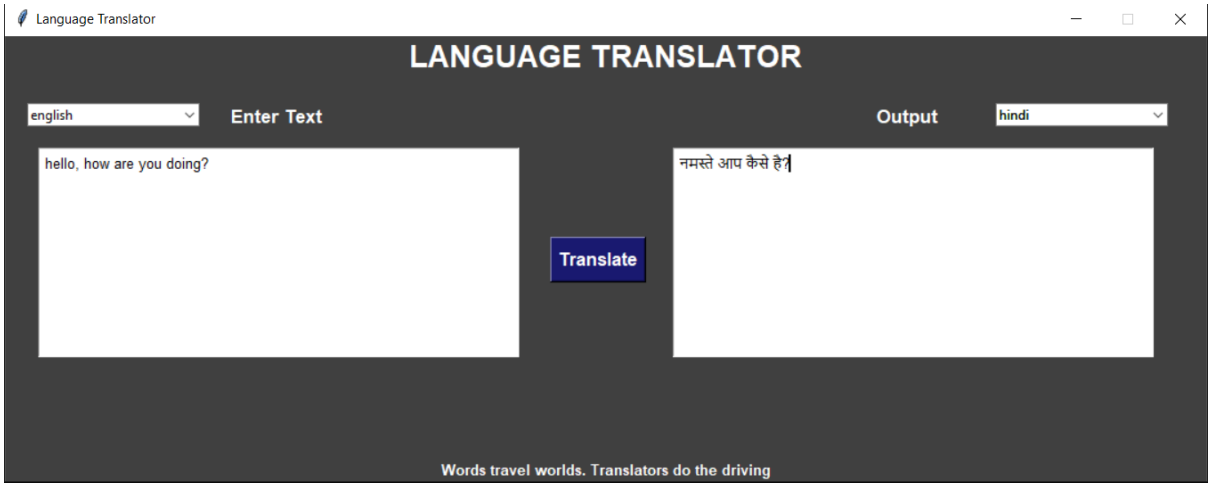
2. Give input:



3. choose output language:



4. click on translate button:



SOURCE CODE:

```
from tkinter import *

from tkinter import ttk

from googletrans import Translator , LANGUAGES

root = Tk()

root.geometry('1080x400')

root.resizable(0,0)

root.title("Language Translator")

root.config(bg = 'gray25')

#heading

Label(root, text = "LANGUAGE TRANSLATOR", font = "arial 20 bold",foreground="white", bg='gray25').pack()

Label(root,text="Words travel worlds. Translators do the driving", font = 'arial 10 bold',foreground="white", bg ='gray25', width = '45').pack(side = 'bottom')

#INPUT AND OUTPUT TEXT WIDGET

Label(root,text ="Enter Text", font = 'arial 13 bold',foreground="white", bg ='gray25').place(x=200,y=60)

Input_text = Text(root,font = 'arial 10', height = 11, wrap = WORD, padx=5, pady=5, width = 60)

Input_text.place(x=30,y = 100)

Label(root,text ="Output", font = 'arial 13 bold',foreground="white", bg ='gray25').place(x=780,y=60)

Output_text = Text(root,font = 'arial 10', height = 11, wrap = WORD, padx=5, pady= 5, width =60)

Output_text.place(x = 600 , y = 100)

###################

language = list(LANGUAGES.values())

src_lang = ttk.Combobox(root, values= language, width =22)

src_lang.place(x=20,y=60)

src_lang.set('choose input language')

dest_lang = ttk.Combobox(root, values= language, width =22)
```

```
dest_lang.place(x=890,y=60)

dest_lang.set('choose output language')

########################################  Define function #######

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)

##########  Translate Button ########

trans_btn = Button(root, text = 'Translate',font = 'arial 12 bold',pady = 5,command = Translate , bg = 'midnight blue', foreground="white",activebackground = 'cornflower blue')

trans_btn.place(x = 490, y = 180)

root.mainloop()
```

REFERENCES:

https://www.geeksforgeeks.org/language-translator-using-google-api-in-python/

https://hub.packtpub.com/create-an-rnn-based-python-machine-translation-system-tutorial/

https://readthedocs.org/projects/translate-python/downloads/pdf/latest/

https://pypi.org/project/googletrans/

https://www.tutorialspoint.com/python_text_processing/python_text_translation.htm

https://docs.transifex.com/tracking/reports