Language Translator Project

INTRODUCTION

INTRODUCTION:

The Language Translation Application is a user-friendly tool developed using Python, designed to perform real-time translation of text between multiple languages. This application leverages Natural Language Processing (NLP) concepts through the googletrans library, which provides a simple interface to Google's powerful translation service.

Built with the Tkinter library for the graphical user interface, the application enables users to enter text in one language and translate it into another using easy dropdown selections. The main focus is to facilitate seamless communication across language barriers, catering to users such as travelers, language learners, and professionals working in multilingual environments.

By integrating an intuitive GUI with multilingual support, the project provides a hands-on demonstration of NLP capabilities in everyday applications, without requiring any prior technical knowledge from the user.

MODULES

MODULES:

1. Language Selection:

This module allows users to choose the source and target languages for translation. Implemented using ttk.Combobox widgets, users can select from a comprehensive list of supported languages provided by the googletrans library. Internally, these languages are mapped to language codes that the API recognizes. The selection system ensures flexibility and adaptability for diverse translation needs.

Language Translator Project

2. Text Input and Output:

This module provides two separate text boxes within the GUI. The first box accepts the user's input text in the source language. After translation, the translated text is displayed in the second box. Both areas support multi-line input, ensuring convenience for translating longer passages or paragraphs.

3. Translation Engine:

This module is the core of the application. It utilizes the Translator class from the googletrans library to send the input text to the Google Translate API. Upon receiving the response, the translated text is shown in the output area. Exception handling is incorporated to manage any errors like invalid language codes or network issues gracefully.

4. Graphical User Interface (GUI):

Developed using Python's Tkinter library, the GUI connects all modules together in an intuitive layout. Key components include labels, text boxes, dropdowns for language selection, and buttons for translating and clearing the text. The GUI is designed for simplicity and clarity, ensuring that even first-time users can navigate it easily.

CODE

```
from tkinter import Tk, Label, Text, Button, ttk, StringVar
from googletrans import Translator, LANGUAGES

# Create the Translator object
translator = Translator()

# Function to perform translation
def translate_text():
    try:
        input_text = text_input.get("1.0", "end-1c")
        source_lang = src_lang.get()
        target_lang = tgt_lang.get()
        translated = translator.translate(input_text, src=source_lang, dest=target_lang)
        text_output.delete("1.0", "end")
```

```
        text_output.insert("1.0", translated.text)
    except Exception as e:
        text_output.delete("1.0", "end")
        text_output.insert("1.0", f"Error: {e}")

# Function to clear text fields
def clear_text():
    text_input.delete("1.0", "end")
    text_output.delete("1.0", "end")

# Initialize the main window
root = Tk()
root.title("Google Translator App")
root.geometry("600x400")
root.resizable(False, False)

Label(root, text="Source Language:", font=("Arial", 12)).place(x=30, y=20)
src_lang = StringVar()
src_dropdown = ttk.Combobox(root, textvariable=src_lang, values=list(LANGUAGES.values()),
state="readonly", width=30)
src_dropdown.place(x=150, y=20)
src_dropdown.set("english")

Label(root, text="Target Language:", font=("Arial", 12)).place(x=30, y=60)
tgt_lang = StringVar()
tgt_dropdown = ttk.Combobox(root, textvariable=tgt_lang, values=list(LANGUAGES.values()),
state="readonly", width=30)
tgt_dropdown.place(x=150, y=60)
tgt_dropdown.set("hindi")

Label(root, text="Text to Translate:", font=("Arial", 12)).place(x=30, y=100)
text_input = Text(root, height=6, width=70, wrap="word", font=("Arial", 10))
text_input.place(x=30, y=130)

Label(root, text="Translated Text:", font=("Arial", 12)).place(x=30, y=250)
text_output = Text(root, height=6, width=70, wrap="word", font=("Arial", 10), bg="#f0f0f0")
text_output.place(x=30, y=280)

translate_button = Button(root, text="Translate", font=("Arial", 12), bg="blue", fg="white",
command=translate_text)
translate_button.place(x=470, y=20)

clear_button = Button(root, text="Clear", font=("Arial", 12), bg="red", fg="white",
command=clear_text)
clear_button.place(x=470, y=60)

root.mainloop()
```