"""

Offline Multilingual Translator using NLLB Distilled Model

Author: [Your Name]

Description: This script provides a Tkinter-based GUI for translating text between multiple languages using a local NLLB distilled model.

"""

import tkinter as tk

from tkinter import ttk, messagebox

from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

# === Load model and tokenizer from local path ===

MODEL\_PATH = "./nllb\_distilled\_model" # Ensure this path contains the model files

tokenizer = AutoTokenizer.from\_pretrained(MODEL\_PATH)

model = AutoModelForSeq2SeqLM.from\_pretrained(MODEL\_PATH)

# === Language code mapping for NLLB-200 ===

LANGUAGES = {

"English": "eng\_Latn",

"French": "fra\_Latn",

"Hindi": "hin\_Deva",

"Spanish": "spa\_Latn",

"Bengali": "ben\_Beng",

"Tamil": "tam\_Taml",

"Telugu": "tel\_Telu",

"Arabic": "arb\_Arab",

"Russian": "rus\_Cyrl",

"Chinese": "zho\_Hans",

"Japanese": "jpn\_Jpan"

}

# === Translation logic ===

def translate(text, src\_lang\_code, tgt\_lang\_code):

try:

tokenizer.src\_lang = src\_lang\_code

encoded = tokenizer(text, return\_tensors="pt")

forced\_bos\_token\_id = tokenizer.convert\_tokens\_to\_ids(tgt\_lang\_code)

output = model.generate(\*\*encoded, forced\_bos\_token\_id=forced\_bos\_token\_id)

translated = tokenizer.batch\_decode(output, skip\_special\_tokens=True)[0]

return translated

except Exception as e:

return f"❌ Error: {str(e)}"

# === GUI setup ===

def create\_gui():

root = tk.Tk()

root.title("🌐 Offline Translator - NLLB 200")

root.geometry("800x500")

root.configure(bg="#E3F2FD")

root.resizable(False, False)

# === Custom Style ===

style = ttk.Style()

style.theme\_use("clam")

style.configure("TLabel", background="#E3F2FD", font=("Segoe UI", 11))

style.configure("TButton", font=("Segoe UI", 11, "bold"), foreground="white", background="#1565C0")

style.map("TButton", background=[("active", "#1976D2")])

style.configure("TCombobox", font=("Segoe UI", 11))

# === Header ===

header = tk.Label(root, text="🌍 Multilingual Translator", font=("Segoe UI", 18, "bold"), bg="#1976D2", fg="white")

header.pack(fill="x", pady=(0, 10))

# === Input Text Box ===

tk.Label(root, text="Enter Text:", font=("Segoe UI", 12, "bold")).pack(anchor="w", padx=15, pady=(5, 0))

input\_text = tk.Text(root, height=5, font=("Segoe UI", 12), wrap="word", bg="#FFFFFF", relief="groove", bd=2)

input\_text.pack(fill="x", padx=15, pady=(0, 10))

# === Language Selection Dropdowns ===

frame = tk.Frame(root, bg="#E3F2FD")

frame.pack(pady=10)

src\_var = tk.StringVar(value="French")

tgt\_var = tk.StringVar(value="English")

ttk.Label(frame, text="Translate From:").grid(row=0, column=0, padx=10)

src\_menu = ttk.Combobox(frame, textvariable=src\_var, values=list(LANGUAGES.keys()), state="readonly", width=20)

src\_menu.grid(row=0, column=1, padx=5)

ttk.Label(frame, text="To:").grid(row=0, column=2, padx=10)

tgt\_menu = ttk.Combobox(frame, textvariable=tgt\_var, values=list(LANGUAGES.keys()), state="readonly", width=20)

tgt\_menu.grid(row=0, column=3, padx=5)

# === Translate Button and Output Display ===

def on\_translate():

text = input\_text.get("1.0", tk.END).strip()

if not text:

messagebox.showwarning("Input Required", "Please enter some text to translate.")

return

src\_code = LANGUAGES[src\_var.get()]

tgt\_code = LANGUAGES[tgt\_var.get()]

result = translate(text, src\_code, tgt\_code)

output\_text.delete("1.0", tk.END)

output\_text.insert(tk.END, result)

translate\_btn = ttk.Button(root, text="🔁 Translate", command=on\_translate)

translate\_btn.pack(pady=10)

tk.Label(root, text="Translation:", font=("Segoe UI", 12, "bold")).pack(anchor="w", padx=15, pady=(5, 0))

output\_text = tk.Text(root, height=5, font=("Segoe UI", 12), wrap="word", bg="#F1F8E9", relief="groove", bd=2)

output\_text.pack(fill="x", padx=15, pady=(0, 10))

root.mainloop()

# === Run GUI ===

create\_gui()