language-detection

April 7, 2025

[5]:

```python
# Install required libraries (check first to avoid redundant installations)
import subprocess
import sys
import os

def install(package):
    subprocess.check_call([sys.executable, "-m", "pip", "install", package])

try:
    import langdetect
except ImportError:
    install("langdetect")

try:
    import googletrans
except ImportError:
    install("googletrans==4.0.0-rc1")

try:
    import nltk
except ImportError:
    install("nltk")

# Import necessary libraries
from langdetect import detect
from googletrans import Translator
import nltk
from nltk import pos_tag, word_tokenize

# Suppress NLTK download messages by redirecting output to os.devnull
def silent_download(package_name):
    with open(os.devnull, 'w') as f:
        nltk.download(package_name, quiet=True)

# Download NLTK data files silently (run once)
silent_download('punkt')
silent_download('averaged_perceptron_tagger')
```

```
# Function for language detection and translation
def translate_and_pos(input_text):
    # Detect language of the input text
    detected_lang = detect(input_text)
    print(f"Detected Language: {detected_lang}")

    # Initialize Google Translator
    translator = Translator()

    # Define supported languages (added 8 languages)
    supported_languages = {
        "Hindi": ('hi', 'Hindi'),
        "Marathi": ('mr', 'Marathi'),
        "Urdu": ('ur', 'Urdu'),
        "Tamil": ('ta', 'Tamil'),
        "Bengali": ('bn', 'Bengali'),
        "Gujarati": ('gu', 'Gujarati'),
        "Kannada": ('kn', 'Kannada'),
        "Telugu": ('te', 'Telugu')
    }

    # Translate text to supported languages
    translations = {}
    for lang_code, lang_name in supported_languages.values():
        try:
            translations[lang_name] = translator.translate(input_text,␣
 ↪src='en', dest=lang_code).text
        except Exception as e:
            print(f"Error translating to {lang_name}: {e}")

    # Display translations and perform POS tagging
    for lang, translated_text in translations.items():
        print(f"{lang} Translation: {translated_text}")

        # Perform POS tagging on the translated text
        tokens = word_tokenize(translated_text)
        pos_tags = pos_tag(tokens)
        print(f"POS Tags for {lang}: {pos_tags}\n")

# Input text from user
input_text = input("Enter text in English: ")
translate_and_pos(input_text)
```

```
Enter text in English: rose color is red
Detected Language: en
Hindi Translation:
```

```
POS Tags for Hindi: [('  ', 'JJ'), (' ', 'NNP'), (' ', 'NNP'), ('  ',
'NNP'), (' ', 'NN')]

Marathi Translation:
POS Tags for Marathi: [('   ', 'JJ'), (' ', 'NNP'), ('  ', 'NNP'),
(' ', 'NN')]

Urdu Translation:
POS Tags for Urdu: [('  ', 'JJ'), (' ', 'NNP'), ('  ', 'NNP'), ('  ',
'NNP'), (' ', 'NN')]

Tamil Translation:
POS Tags for Tamil: [('  ', 'JJ'), ('  ', 'NNP'), ('   ', 'NN')]

Bengali Translation:
POS Tags for Bengali: [('   ', 'JJ'), (' ', 'NNP'), ('  ', 'NN')]

Gujarati Translation:
POS Tags for Gujarati: [('   ', 'JJ'), (' ', 'NNP'), ('  ', 'NNP'),
(' ', 'NN')]

Kannada Translation:
POS Tags for Kannada: [('  ', 'JJ'), ('  ', 'NNP'), ('  ', 'NN')]

Telugu Translation:
POS Tags for Telugu: [('  ', 'JJ'), ('  ', 'NNP'), ('   ', 'NN')]
```

[ ]: