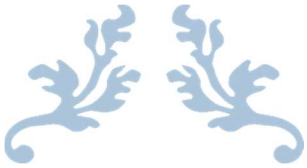


LANGUAGE TRANSLATOR

By using PYTHON

STUDENT: NAVEED HAYDER

SUBMITTED TO: SIR ABDULLAH ARIF
(BANO QABIL 2.0)

Overview

Our task is to make a Language translator which would help us to translate a word, sentence or even a paragraph to another language. We will try to include as many languages as possible. We will be using Tkinter module to shape our GUI for the project and googletrans library to present us with a number of languages that are part of it. Also, we will use the Text Bob library for processing textual data.

Benefits

A language translator in Python can indeed be a valuable problem-solving tool for users, depending on their needs and circumstances. Here are some scenarios where a language translator can address user challenges:

Cross-Language Communication:

A language translator enables users who speak different languages to communicate more effectively. This can be particularly useful in international business, travel, or collaborative projects involving participants from diverse linguistic backgrounds.

Language Learning:

Language learners can use a translator to check their understanding of words, phrases, or sentences in another language. It can aid in comprehension and serve as a valuable tool for language acquisition.

Access to Information:

Users can leverage a language translator to access information, articles, or resources that are available in languages they may not be proficient in. This can broaden their knowledge base and facilitate learning from diverse sources.

Cultural Understanding:

Translating text can contribute to a better understanding of different cultures. Users can explore literature, news, or content in other languages to gain insights into cultural nuances and perspectives.

Travel Assistance:

Travelers can use language translators to overcome language barriers when navigating in foreign countries. It helps with tasks such as reading signs, communicating with locals, or understanding menus.

Global Collaboration:

In collaborative projects involving team members from different parts of the world, a language translator can facilitate effective communication, making it easier for team members to collaborate and contribute.

Accessibility:

Language translators can enhance accessibility for users who may have difficulty understanding content in a particular language. It can contribute to a more inclusive online environment.

Personal Communication:

Individuals with friends or family members who speak different languages can use language translators to facilitate personal communication, especially in written form through emails or messages.

Explanation of the code logic and design

Design:

Marian MT Model is a family of neural machine translation models trained on the OPUS dataset, a large-scale multilingual corpus. These models are designed for translating text between multiple languages, and they achieve state-of-the-art performance in many translation tasks.

Step 1: Install Hugging Face Transformers Library

In a new code cell, run the following command to install the Hugging Face Transformers library:

```
!pip install transformers
```

Step 3: Install Sentence Piece Library

In a new code cell, run the following command to install the SentencePiece library, which is required for the MarianTokenizer:

```
!pip install sentencepiece
```

Step 4: Import Required Libraries

Import the necessary libraries for the project:

```
from transformers import MarianMTModel, MarianTokenizer
```

Step 5: Define Translation Function

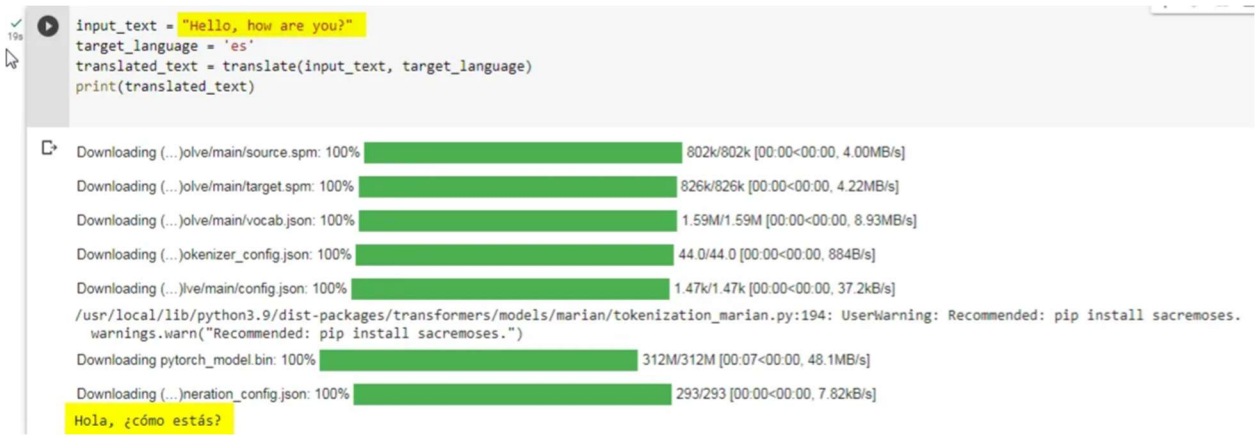
Create a function to translate text from English to a target language using a pre-trained model:

```
def translate(text, target_language):
    model_name = f'Helsinki-NLP/opus-mt-en-{target_language}'
    tokenizer = MarianTokenizer.from_pretrained(model_name)
    model = MarianMTModel.from_pretrained(model_name)

    inputs = tokenizer.encode(text, return_tensors="pt")
    outputs = model.generate(inputs, num_beams=4, max_length=50,
early_stopping=True)
    translated_text = tokenizer.decode(outputs[0], skip_special_tokens=True)

    return translated_text
```

Step 6: Translate Text

Use the translation function to translate text from English to the desired target language. Replace `'es'` with the language code of your choice (e.g., `'fr'` for French, `'ru'` for Russian, or `'zh'` for Mandarin):

```
input_text = "Hello, how are you?"
target_language = 'es'
translated_text = translate(input_text, target_language)
print(translated_text)
```



Language Codes

Here is a table of some popular languages supported by MarianMTModel along with their language codes:

Language	Language Code
Arabic	ar
Chinese	zh
Dutch	nl
English	en
Finnish	fi
French	fr
German	de
Hindi	hi
Italian	it
Japanese	ja
Korean	ko
Polish	pl
Portuguese	pt
Russian	ru
Spanish	es
Swedish	sv
Turkish	tr

Conclusion

Congratulations! You have successfully created a translation system using a pre-trained MarianMTModel from the Hugging Face Transformers library in Google Colab. You can now translate text from English to various languages by simply changing the target language code in the `translate` function.

My GitHub Repository

English to various languages by simply changing the target language code in the `translate` function.