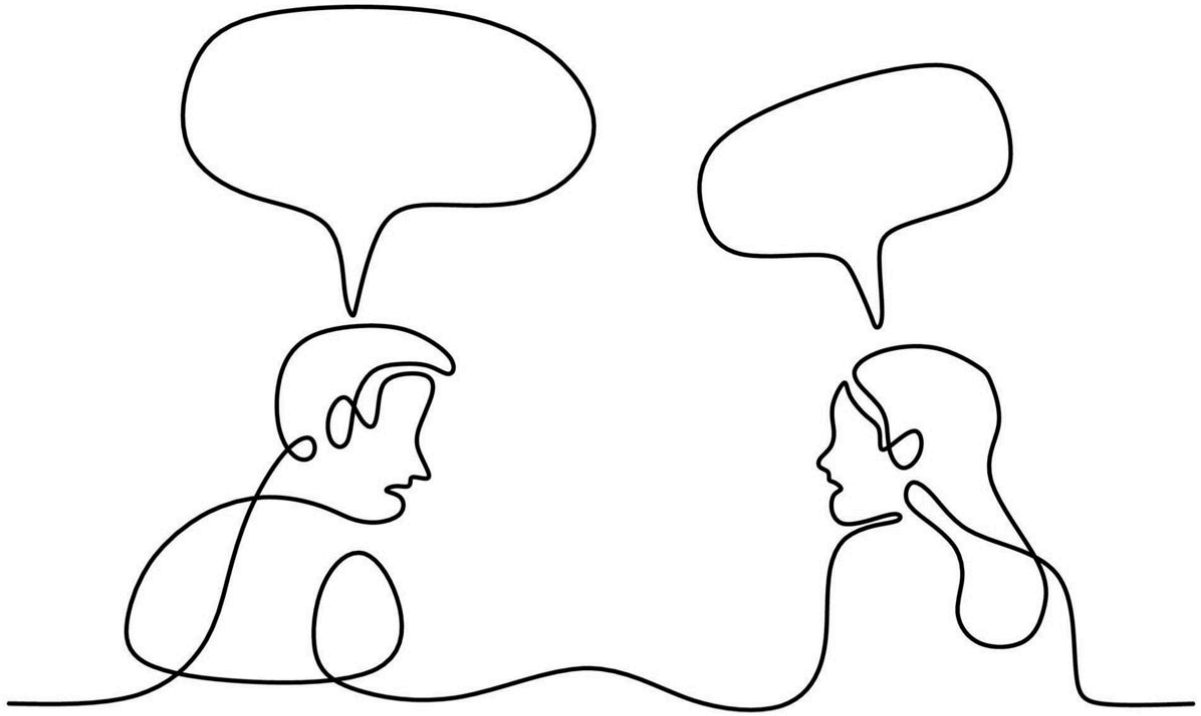


AI Based Project



Name - Sai Purvi S

Date - 21st October, 2025

Project Name - AI Language Translator

Tools Used - Python (PyTorch & Hugging Face and its pre trained model)

Problem Statement –

The goal of this project is to build an English-to-Other-Languages Translator using Hugging Face's Transformers and PyTorch. The purpose of this project is primarily to learn and understand how to leverage pre-trained NLP models for translation tasks and gain practical experience with transformer-based models.

Tools Used –

1. Transformers (by Hugging Face):

Purpose: This library provides access to pre-trained NLP models for tasks like translation, text generation, and summarization. We used it to load the MarianMT model, which helps us translate English to other languages like French, German, Hindi, etc.

2. torch (PyTorch):

Purpose: PyTorch is a popular deep learning framework used to perform computations for models like MarianMT. It allows the model to understand text, process it, and generate translations. PyTorch is the engine that runs the underlying neural networks for language translation.

3. sentencepiece:

Purpose: This is a tokenizer used to break down text into smaller units (tokens). It is commonly used with transformer models, like MarianMT, to process text before it's fed into the model. Although it wasn't strictly necessary for our use case, it helps with more efficient tokenization and performance.

Implementation –

This project builds an English-to-Other-Languages Translator using the Hugging Face Transformers library and PyTorch.

1. Library Setup:

The required libraries were installed using pip: transformers (for pre-trained models), torch (for model computations), and sentencepiece (for tokenization).

2. Loading the Model:

The MarianMT model from Hugging Face was used for translation (e.g., English to French, German, Italian, Dutch, Hindi)

3. Translation Function:

The translation function converts English text into tokens, feeds them into the model, and generates the translation, which is then decoded into readable text.

4. User Interaction:

The user selects a target language and inputs English text. The program then translates the text and displays the result.

This is done in a loop, allowing continuous translations until the user types quit.

5. Model Caching:

To optimize performance, once a model is loaded for a language, it is cached for reuse, avoiding repeated downloads and speeding up subsequent translations.

Input Code –

```
LangTranslate.py 1 X
LangTranslate.py > ...
1  from transformers import MarianMTModel, MarianTokenizer
2
3  # Dictionary of target languages and their Hugging Face models
4  models = {
5      "French": "Helsinki-NLP/opus-mt-en-fr",
6      "German": "Helsinki-NLP/opus-mt-en-de",
7      "Italian": "Helsinki-NLP/opus-mt-en-it",
8      "Dutch": "Helsinki-NLP/opus-mt-en-nl",
9      "Hindi": "Helsinki-NLP/opus-mt-en-hi"
10 }
11
12 # Cache loaded models to avoid re-downloading
13 loaded_models = {}
14
15 def translate(text, target_lang):
16     if target_lang not in loaded_models:
17         model_name = models[target_lang]
18         tokenizer = MarianTokenizer.from_pretrained(model_name)
19         model = MarianMTModel.from_pretrained(model_name)
20         loaded_models[target_lang] = (tokenizer, model)
21     else:
22         tokenizer, model = loaded_models[target_lang]
23
24     tokens = tokenizer(text, return_tensors="pt", padding=True)
25     translated = model.generate(**tokens)
26     return tokenizer.decode(translated[0], skip_special_tokens=True)
27
28 # Main loop
29 print("=== Multi-Language Translator (English → Other Languages) ===")
30 print("Type 'quit' at any time to exit.\n")
31
```

Activate Windows

```
LangTranslate.py 1 X
LangTranslate.py > translate
32 while True:
33     print("Available target languages:")
34     for i, lang in enumerate(models.keys(), start=1):
35         print(f"{i}. {lang}")
36
37     lang_choice = input("Choose the target language (name or number): ")
38
39     if lang_choice.lower() == "quit":
40         print("Exiting translator. Goodbye!")
41         break
42
43     # Map number to language if user entered a number
44     if lang_choice.isdigit():
45         idx = int(lang_choice) - 1
46         if idx < 0 or idx >= len(models):
47             print("Invalid choice. Try again.\n")
48             continue
49         target_lang = list(models.keys())[idx]
50     else:
51         target_lang = lang_choice.title()
52         if target_lang not in models:
53             print("Invalid choice. Try again.\n")
54             continue
55
56     text = input(f"Enter text in English to translate to {target_lang}: ")
57     if text.lower() == "quit":
58         print("Exiting translator. Goodbye!")
59         break
60
61     translation = translate(text, target_lang)
62     print(f"Translated text ({target_lang}): {translation}")
63     print("-" * 50)
```

Activate Windows

Go to Settings to activate Windows.

Output –

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\Language Translator> python LangTranslate.py
=== Multi-Language Translator (English → Other Languages) ===
Type 'quit' at any time to exit.

Available target languages:
1. French
2. German
3. Italian
4. Dutch
5. Hindi
Choose the target language (name or number): 1
Enter text in English to translate to French: Open your eyes and see the world around you
Translated text (French): Ouvre les yeux et vois le monde qui t'entoure
-----
Available target languages:
1. French
2. German
3. Italian
4. Dutch
5. Hindi
Choose the target language (name or number): 2
Enter text in English to translate to German: Im glad the code works proper
Translated text (German): Ich bin froh, dass der Code richtig funktioniert
-----
Available target languages:
1. French
2. German
3. Italian
4. Dutch
5. Hindi
Choose the target language (name or number): 3
Enter text in English to translate to Italian: Work hard and live happily
Translated text (Italian): Lavora sodo e vivi felicemente
-----
Available target languages:
1. French
2. German
3. Italian
4. Dutch
5. Hindi
Choose the target language (name or number): 4
Enter text in English to translate to Dutch: Hard work will always brings peace
Translated text (Dutch): Hard werken zal altijd vrede brengen.
-----
Choose the target language (name or number): 5
Enter text in English to translate to Hindi: Vrindavan is the best place. its heaven
Translated text (Hindi): वैष्णव सबसे अच्छा ग्रह है.
-----
Available target languages:
1. French
2. German
3. Italian
4. Dutch
5. Hindi
Choose the target language (name or number): quit
Exiting translator. Goodbye!
PS C:\Users\Lenovo\Desktop\Language Translator> |
```

Conclusion –

This project helped me understand how Python libraries like PyTorch and Hugging Face Transformers can be used to build real-world applications such as a language translator. By implementing this, I learned how pre-trained models work, how text is tokenized and translated, and how to interact with machine learning models using simple Python code. Overall, the project improved my knowledge of AI-based translation, library integration, and practical programming skills in Python.