

NAME : JAISURYAH K P
REGNO: 727822TUIT063
DEPT : BTECH INFORMATION TECHNOLOGY
SECTION : IT - A
COURSE : ARTIFICIAL INTELLIGENCE
COURSE CODE : 22IT401

LANGUAGE TRANSLATOR USING NLP MODEL

Abstract:

This report presents a Python script for translating text from English to Hindi utilizing the MarianMTModel from the Transformers library. The script employs a pre-trained model specifically designed for English-Hindi translation. It provides an overview of the implementation, including necessary packages, code explanation, and usage instructions. The process involves tokenizing the input English text using the MarianTokenizer, followed by generating the translated Hindi text using the MarianMTModel. The model utilizes a transformer architecture trained on large bilingual corpora to produce accurate translations. Various parameters such as maximum length of output text and number of beams are adjusted to optimize translation quality and efficiency. The developed translation function allows users to input English text for translation, providing a seamless experience for converting English content into Hindi. The translated text is then presented to the user, enabling cross-lingual communication and facilitating accessibility to Hindi-speaking audiences.

Introduction:

Machine translation plays a pivotal role in breaking down language barriers by automatically converting text from one language to another. This report focuses on the task of translating English text to Hindi using a state-of-the-art pre-trained model provided by the Transformers library. Machine translation, the automated process of converting text from one language to another, has become increasingly essential in our interconnected global society. With the advent of advanced Natural Language

Processing (NLP) techniques and powerful deep learning models, machine translation systems have made significant strides in accurately and fluently translating text across multiple languages. In this study, we focus on the task of translating English text to Hindi, a widely spoken language with millions of speakers worldwide.

The ability to translate English text to Hindi opens up avenues for cross-cultural communication, content localization, and accessibility to Hindi-speaking audiences. It facilitates the dissemination of information, fosters intercultural understanding, and promotes inclusivity in various domains such as education, business, and technology. With the proliferation of digital content on the internet, the demand for high-quality translation services has surged, driving the need for robust and efficient machine translation models.

In this context, the Helsinki-NLP/opus-mt-en-hi model emerges as a valuable resource for English-to-Hindi translation tasks.

Top language models

1. BERT
2. Claude
3. Cohere
4. Ernie
5. Falcon 40B
6. Galactica
7. GPT-3
8. GPT-3.5
9. GPT-4
10. Lamda
11. Llama
12. Orca
13. Palm
14. Phi-1
15. StableLM
16. Vicuna 33B

Required Packages:

The successful execution of the provided Python script necessitates the installation of the following packages:

- torch: PyTorch, a powerful machine learning framework that underpins many natural language processing (NLP) applications.
- transformers: The Hugging Face Transformers library, an essential toolkit for accessing pre-trained models and utilities for various NLP tasks, including translation.

Ensure that these packages are installed within your Python environment using the following commands:

pip install torch transformers

Implementation Overview:

The Python script follows a systematic approach to translate English text to Hindi using the MarianMTModel. Here's a breakdown of the implementation:

- Model and Tokenizer Loading: The script loads a pre-trained MarianMTModel and its corresponding tokenizer specifically designed for English-Hindi translation.
- Translation Function: It defines a translation function responsible for tokenizing the input English text, translating it to Hindi, and decoding the translated output into readable text.
- User Input: The script prompts the user to input the English text they wish to translate to Hindi.
- Translation: It translates the input English text to Hindi using the defined translation function.
- Output Display: Finally, the original English text and its corresponding Hindi translation are printed for user reference.

the task is to translate English text to Hindi using the Helsinki-NLP/opus-mt-en-hi model. Let's break down the code and explain the packages and language models used:

Packages Used:

- transformers: This is a Python library by Hugging Face that provides pre-trained models for Natural Language Processing (NLP) tasks. It includes a wide range of models for tasks such as text classification, translation, summarization, and more.
- MarianTokenizer: This class is used for tokenizing text input for the MarianMTModel. It converts text into numerical tokens that can be processed by the model.
- MarianMTModel: This class is a part of the MarianMT (Marian Machine Translation) architecture, which is designed for machine translation tasks. It takes tokenized input and produces translated text as output.

Language Model Used:

- Helsinki-NLP/opus-mt-en-hi: This is a model for English-to-Hindi machine translation. It is trained and fine-tuned on large bilingual corpora to perform translation tasks specifically from English to Hindi. The model is part of the Open-Source Project for a Universal Sentence Encoder (OPUS) provided by the Helsinki NLP group.

Function for Translation (translate):

- This function takes three parameters: the input text to be translated, the pre-trained model for translation, and the tokenizer for processing the input text.
- It first tokenizes the input text using the provided tokenizer.
- Then, it generates the translated text using the pre-trained model. The generate method of the model is used for text generation, where various parameters such as maximum length of the output text, number of beams, and early stopping conditions are specified.
- Finally, it decodes the generated tokens into human-readable text using the tokenizer and returns the translated text.

User Input and Output:

- The code prompts the user to input the English text they want to translate into Hindi.
- After translating the text, it prints both the original English text and the translated Hindi text.

this code snippet demonstrates how to perform machine translation using a pre-trained MarianMT model for translating English text to Hindi, utilizing the Hugging Face transformers library.

CODING:

```
import torch
from transformers import MarianMTModel, MarianTokenizer

# Load pre-trained model and tokenizer
model_name = 'Helsinki-NLP/opus-mt-en-hi'
tokenizer = MarianTokenizer.from_pretrained(model_name)
model = MarianMTModel.from_pretrained(model_name)

# Function for translation
def translate(text, model, tokenizer):
    inputs = tokenizer(text, return_tensors="pt", padding=True, truncation=True,
max_length=512)
    outputs = model.generate(**inputs, max_length=128, num_beams=4,
early_stopping=True)
    translated_text = tokenizer.decode(outputs[0], skip_special_tokens=True)
    return translated_text

# Take user input for English text
english_text = input("Enter the English text you want to translate to Hindi: ")

# Translate to Hindi
hindi_translation = translate(english_text, model, tokenizer)

# Print translated text
print("English:", english_text)
```

```
print("Hindi:", hindi_translation)
```

OUTPUT:

Enter the English text you want to translate to Hindi: my name is jaisurya

English: my name is jaisurya

Hindi: मेरा नाम तसोरिया है

Recommendation:

- **Model Selection:** Experiment with different pre-trained models and hyperparameters to optimize translation quality and performance based on specific use cases and requirements.
- **Error Handling:** Implement robust error handling mechanisms to gracefully handle unexpected input or errors during the translation process.
- **Usage Compliance:** Ensure compliance with usage policies and terms of service when utilizing pre-trained models, especially in production environments, to avoid potential legal or ethical issues.

Conclusion:

The provided Python script offers a straightforward yet effective solution for translating English text to Hindi using a pre-trained MarianMTModel. By leveraging pre-trained models, developers can achieve accurate and efficient language translation with minimal effort, thereby facilitating cross-lingual communication and understanding.