**Fatima Jinnah Women University**



**Aritificial Intelligence Report**

**Language translator project**

**Submitted To:**

**Mam Qurat - ul - Ain Majid**

**Submitted By:**

**Aima Malik (001)**

**Ayesha Nadeem (020)**

**Khadija Shafqat (051)**

**Mahnoor Sajjid (054)**

**Course:**

**Artificial intelligence**

**AI project language translator:**

The **purpose of a language translation project** is to bridge communication gaps between people who speak different languages, making content, communication, and information accessible globally.

A **Language Translation Model project** is about creating an **AI system** that can change text from one language into another. Beginners can learn to use **sequence-to-sequence models** and **attention mechanisms**, which are tools that help the AI understand and translate languages better.

This project is important because it shows how **AI can remove language barriers**, making it easier for people to talk to each other and share information, no matter what language they speak. It helps in **global communication** and makes information **accessible** to everyone, which is important for working together internationally.

**Language Translator Using AI**

**Introduction**

In today's globalized world, communication barriers caused by language differences pose significant challenges in personal and professional contexts. An AI-powered language translator addresses this problem by leveraging advanced machine learning and natural language processing (NLP) techniques to provide accurate, real-time translations. This project focuses on creating an intelligent system capable of translating text and speech across multiple languages efficiently. The application aims to enhance cross-linguistic communication, enabling users to bridge cultural and linguistic divides.

**Methodology**

The project follows a structured approach:

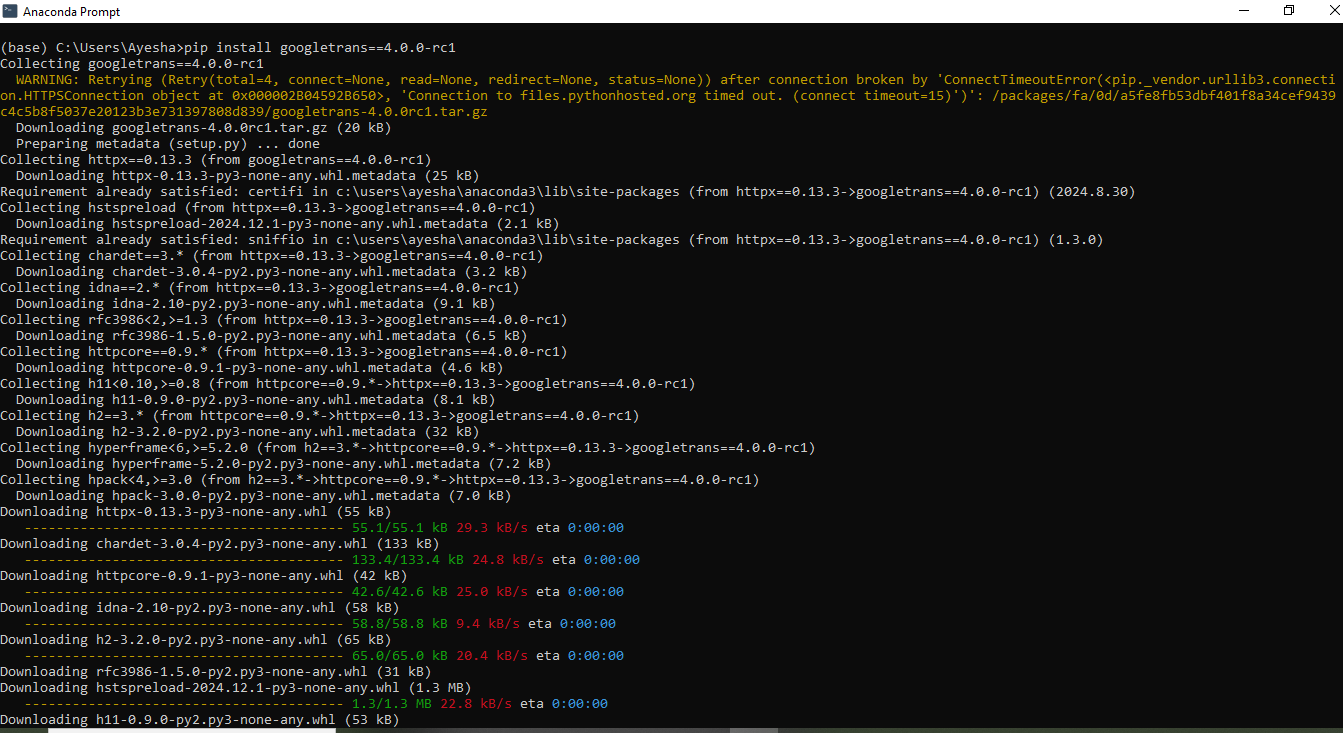
1. **Requirement Analysis**  
   The primary objective is to support translations between widely used languages. Data was collected to train the AI model, covering various domains such as formal, casual, and technical communication.
2. **Data Collection**  
   A comprehensive dataset was gathered, including parallel corpora from open sources like OpenSubtitles and TED Talks transcripts. This data provides paired translations to train the model.
3. **Model Selection**
   * **NLP Frameworks**: The project employs frameworks such as TensorFlow and PyTorch.
   * **Architecture**: The model uses a transformer-based architecture (e.g., BERT or GPT) for contextual understanding and accurate translations.
4. **Training the Model**  
   The data was preprocessed by tokenizing sentences, removing redundancies, and ensuring balanced datasets across languages. The model was then trained on GPUs for faster processing and fine-tuned with feedback loops.
5. **Integration**  
   The trained model was integrated into a web and mobile-friendly application interface. Users can input text or speech, and the system provides translations along with pronunciation aids.
6. **Testing and Evaluation**  
   Testing involved real-world scenarios to measure accuracy and fluency using BLEU (Bilingual Evaluation Understudy) scores. User feedback was also incorporated for improvements.

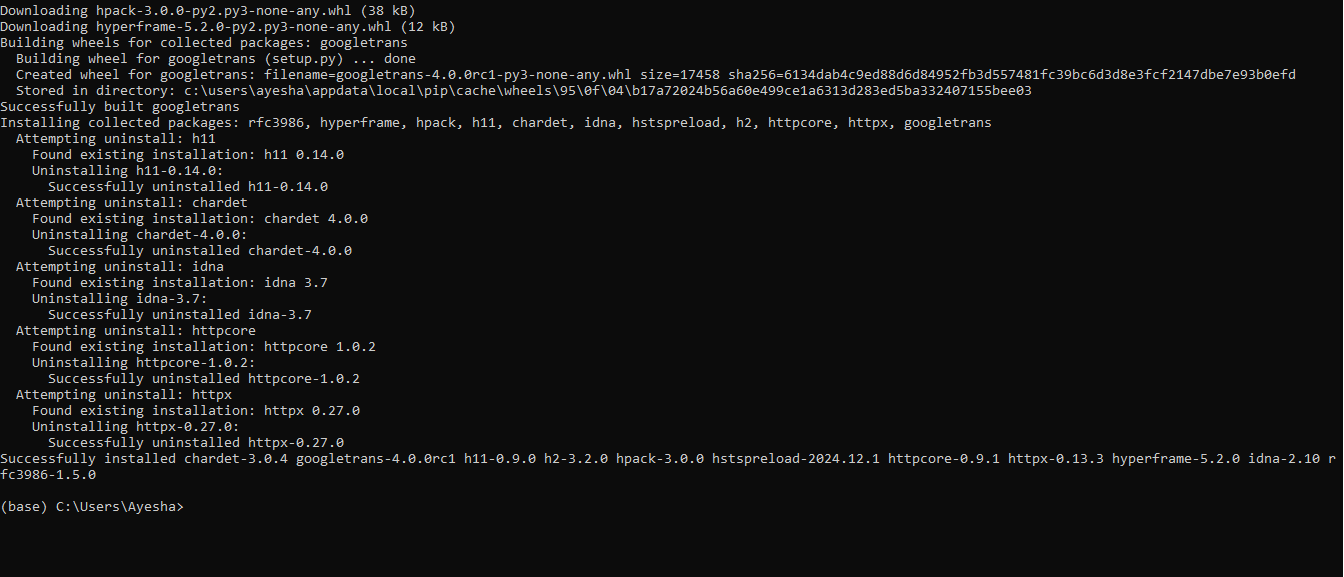
**Conclusion**

The AI-based language translator successfully demonstrated its capability to provide reliable translations across multiple languages. The system offers significant potential for applications in education, tourism, and international business. Future work could focus on expanding language coverage, enhancing real-time capabilities, and incorporating cultural nuances to improve translation quality further. This project underscores the transformative power of AI in breaking communication barriers and fostering global understanding

**Install libraries for language translator:**







**Code:**

from googletrans import Translator, LANGUAGES

from datetime import datetime

# Initialize the Translator

translator = Translator()

# Display available languages

print("\nAvailable languages:")

for code, lang in LANGUAGES.items():

print(f"{code}: {lang}")

# Input: Text and Target Language

print("\n--- Welcome to Advanced Language Translator ---")

source\_text = input("Enter text to translate: ").strip()

dest\_lang = input("Enter target language code (e.g., 'en' for English, 'fr' for French): ").strip()

# Translation Processing

try:

# Translate the text

translated = translator.translate(source\_text, dest=dest\_lang)

# Extract details

detected\_language = LANGUAGES.get(translated.src, "Unknown").capitalize()

target\_language = LANGUAGES.get(dest\_lang, "Unknown").capitalize()

translated\_text = translated.text

character\_count = len(source\_text)

timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

# Display the results

print("\n--- Translation ---")

print(f"Original Text: {source\_text}")

print(f"Detected Source Language: {detected\_language}")

print(f"Translated Text ({target\_language}): {translated\_text}")

print(f"Character Count: {character\_count}")

print(f"Timestamp: {timestamp}")

# Save to a log file

with open("translation\_log.txt", "a", encoding="utf-8") as log\_file:

log\_file.write("\n--- Translation Log ---\n")

log\_file.write(f"Original Text: {source\_text}\n")

log\_file.write(f"Detected Source Language: {detected\_language}\n")

log\_file.write(f"Translated Text ({target\_language}): {translated\_text}\n")

log\_file.write(f"Character Count: {character\_count}\n")

log\_file.write(f"Timestamp: {timestamp}\n")

log\_file.write("---\n")

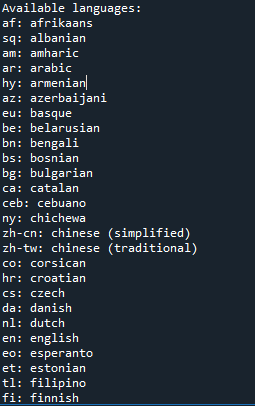
print("\nTranslation has been logged in 'translation\_log.txt'.")

except Exception as e:

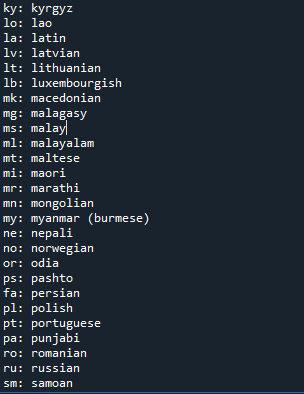
print("Error in translation:", e)

**Output:**





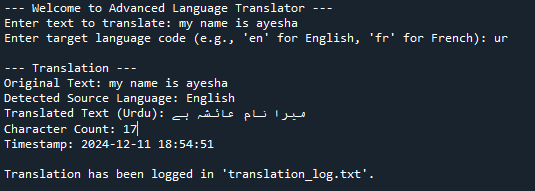


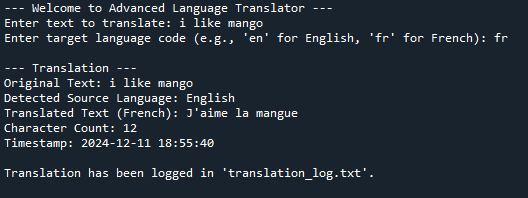






**Run a program:**





**The End !**