LANGUAGE TRANSLATION WITH

SEQUENCE MODELS

by

Chitransha Bhatt 202410116100054
Devansh Batta 202410116100058
Anurag Kushwaha 202410116100039
Bhawna Rajput 202410116100049

Session: 2024-2025 (II Semester)

Under the supervision of

Mr. Apoorv Jain

KIET Group of Institutions, Delhi-NCR, Ghaziabad

DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS, DELHI-NCR,
GHAZIABAD-201206
(MAY-2025)

CERTIFICATE

Certified that **Chitransha Bhatt 202410116100054, Devansh Batta 202410116100058, Anurag Singh Kushwaha 202410116100039, Bhavna Rajput 202410116100049** has/ have carried out the project work having Language Translation with Sequence Models (**AI, AI101B**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Ms. Apoorv Jain
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Akash Rajak
Dean
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Mr. Apoorv Jain** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Akash Rajak, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Chitransha Bhatt

Devansh Batta

Anurag Singh Kushwaha

Bhavna Rajput

TABLE OF CONTENTS

Certificate ii

Abstract iii

Acknowledgements iv

Table of Contents v

1 Introduction 1-16

1.1 Overview 2

1.1.1 ……. 4

1.1.1.1 ……. 4

1.1.1.2 **……..** 7

1.1.2 …….. 10 1.1.2.1 …… 10

1.2 …… 12

1.2.1 …… 13

1.2.2 …….. 13

1.3 ……. 14

1.4 …… 15

1.5 …….. 16

2 Feasibility Study/Literature Review 17-36

2.1 ……. 18

2.2 …… 18 2.3 …… 35 2.4 …… 35

2.5 …… 36

3 Project / Research Objective

4 Hardware and Software Requirements

5 Project Flow/ Research Methodology

6 Project / Research Outcome

References/ Bibliography

1.INTRODUCTION

Language Translation with Sequence Models

Language translation is one of the most impactful applications of Artificial Intelligence and Natural Language Processing (NLP). It enables machines to automatically convert text or speech from one language to another—bridging communication gaps across cultures, countries, and communities.

Traditional rule-based translation systems relied heavily on linguistic rules and dictionaries, but they often struggled with context, grammar, and idioms. Modern AI-based approaches, especially **sequence models**, have transformed the field with far greater fluency and accuracy.

What are Sequence Models?

Sequence models are deep learning architectures designed to handle **sequential data**—like sentences, where word order matters. In language translation, they learn to map an input sequence in one language (e.g., English) to an output sequence in another language (e.g., French).

These models understand not just word meanings, but also **context, grammar, and sentence structure**, enabling more natural and accurate translations.

Why is this classification Important?

1. **Bridges Communication Gaps**
   - Allows people from different linguistic backgrounds to **communicate effectively**.
   - Essential for **international collaboration**, education, tourism, and diplomacy.
2. **Boosts Machine Understanding**
   - Sequence models like LSTMs and Transformers help machines understand **not just words, but the full meaning and structure of sentences**.
   - This makes translations more **accurate, context-aware, and natural-sounding**.
3. **Enables Real-Time Applications**
   - Supports real-time translation in tools like:
     - **Google Translate**
     - **Multilingual chatbots**
     - **Voice assistants (like Alexa or Siri)**

4. **Improves Access to Information**
    - Breaks down language barriers on the internet.
    - Let's users read **books, research papers, and news articles** in different languages.
5. **Supports Global Business**
    - Helps businesses offer **multilingual support**, localize content, and expand into international markets.
    - Enhances **customer experience** in e-commerce, tech support, and marketing.
6. **Advances AI Research**
    - Sequence models used in translation (like Transformers) have become **foundational in AI**, leading to breakthroughs in other tasks like:
        - Text summarization
        - Question answering
        - Sentiment analysis

Approaches

Over the years, different approaches have evolved to improve the accuracy and fluency of machine translation. Here are the major ones:

1. Rule-Based Machine Translation (RBMT)

(Traditional approach – not based on learning)

- Uses manually crafted grammar rules and dictionaries.
- Translates based on predefined linguistic patterns.
- Limitations: Poor at handling idioms, slang, or complex sentence structures.

2. Statistical Machine Translation (SMT)

(Data-driven, but not deep learning)

- Learns translation probabilities from bilingual corpora.
- Uses models like phrase-based or word-based translation.
- Eg: IBM Models, Google Translate (early version).
- Limitations: Lacks deep contextual understanding.

3. Neural Machine Translation (NMT)

(Modern, deep learning-based)
This is where sequence models come in.

a) Seq2Seq (Sequence-to-Sequence) Models

- Based on Encoder-Decoder architecture.
- Encoder processes the input sentence → produces a context vector.
- Decoder generates the translated output from this vector.
- Commonly used with LSTM or GRU units.

b) Attention Mechanism

- Improves upon vanilla Seq2Seq by allowing the decoder to "attend" to specific parts of the input sentence at each step.
- Solves problems related to long sentences and memory limitations.
- Makes translations more context-aware.

c) Transformer Models (SOTA)

- Introduced in the paper "Attention is All You Need" (2017).
- Fully based on self-attention, no RNNs involved.
- Allows parallel processing → much faster and more accurate.
- Powers popular models like:
    - BERT (Bidirectional Encoder Representations from Transformers)
    - GPT (Generative Pre-trained Transformer)
    - MarianMT (used in many real-world translation apps)

Applications:

1. Real-Time Translation Services

- Google Translate, Microsoft Translator, DeepL use advanced sequence models for fast and accurate translations.
- Enables travelers, students, and professionals to communicate across language barriers.

2. Chatbots and Virtual Assistants

- Multilingual AI assistants like Alexa, Siri, and Google Assistant use language translation to respond to users in different languages.
- Customer support bots can handle queries in multiple languages without needing a human translator.

3. Education and E-Learning

- Platforms like Duolingo or Coursera offer content and instructions in multiple languages.
- Helps students learn new languages using translated learning materials.

4. Mobile Applications

- Language learning apps, translator apps, and travel assistants on smartphones utilize on-device NMT (Neural Machine Translation) models.
- Many now support offline translations using lightweight transformer models.

5. Content Localization and Global Media

- Translation tools help localize movies, games, websites, books, and social media content.
- Subtitle generation, voice-over synchronization, and script translation are all powered by sequence models.

6. Business and E-commerce

- Global brands like Amazon, eBay, and Alibaba use translation to:
- Translate product descriptions.
- Enable multi-language customer interactions.
- Manage international communication.

7. Healthcare & Government Services

- Used in hospitals and public service sectors to translate documents, prescriptions, and instructions for non-native speakers.
- Helps in disaster response and legal translation where accuracy is crucial.

8. Research and Data Access

- Enables researchers to read papers and journals published in other languages.
- Translates academic articles, government reports, and historical documents.

9. Travel and Tourism

- Assists travelers with menu translation, signage reading, and real-time voice interpretation.
- Apps like Google Lens combine camera and NMT to translate text in real-world scenes.

2. METHODOLOGY

The methodology outlines the step-by-step process used to perform language translation using deep learning-based sequence models. Here's how it typically works: **1. Problem Definition**

Step 1: Text Preprocessing

- Input Sentence Tokenization: The input sentence is broken down into smaller units (tokens), such as words or subwords.
- Lowercasing & Cleaning: Optional normalization like removing punctuation, converting to lowercase, etc.
- Language Detection (Optional): Libraries like langdetect or fastText can be used to identify the source language.

Step 2: Encoder-Decoder Architecture (Seq2Seq)

Encoder

- Converts the input sentence (in source language) into a fixed-length vector.
- Uses RNNs, LSTMs, or GRUs to understand context and sentence structure.

Decoder

- Takes the context vector from the encoder and generates a sentence in the target language.
- Produces the output word by word.

Step 3: Attention Mechanism (Enhanced Seq2Seq)

- Instead of relying on a single context vector, attention allows the decoder to **focus on specific parts of the input sequence** during translation.
- Significantly improves results, especially for long or complex sentences.

Step 4: Transformer Model (Modern Approach)

- Replaces RNNs/LSTMs with **self-attention and feed-forward layers**.

- Allows **parallel processing** of the entire input sequence → faster and more scalable.
- Architecture consists of:
    - **Positional Encoding**
    - **Multi-head Attention**
    - **Layer Normalization & Feedforward Layers**

Step 5: Training the Model (If Not Pretrained)

- Collect a **parallel corpus** (sentence pairs in source and target languages).
- Train the model using **cross-entropy loss** to minimize the difference between predicted and actual translations.
- **Evaluation metrics** like BLEU score are used to measure performance.

Step 6: Inference / Translation Output

- During inference, given a new sentence:
    - The model encodes it.
    - Translates it using **greedy decoding** or **beam search** to generate the most likely output sentence.

3.Performance Criteria

To assess how well a language translation model performs, several qualitative and quantitative metrics are used. These criteria help evaluate accuracy, fluency, speed, and generalizability.

1. BLEU Score (Bilingual Evaluation Understudy)

- Most commonly used metric in machine translation.
- Compares machine-translated output with one or more reference human translations.
- Calculates n-gram overlaps (1-gram, 2-gram, etc.) between predicted and actual output.
- Score ranges from 0 to 1 (or 0 to 100 when scaled) → higher is better.

2. Translation Quality (Human Evaluation)

- Human experts rate the translation on:
    - Adequacy (Does it convey the original meaning?)
    - Fluency (Is the grammar natural in the target language?)

- Ratings typically on a scale of 1–5 or "Good / Acceptable / Bad".

3. TER (Translation Edit Rate)

- Measures the number of edits needed to change the model's output into the reference translation.
- Includes insertions, deletions, substitutions, and shifts.
- Lower TER = better translation.

4. Accuracy & Precision (for classification)

- In classification-based approaches (e.g., choosing the best translation from a set), standard metrics like accuracy, precision, recall, and F1-score may be used.

5. Inference Speed

- Time taken to translate a sentence or document.
- Important for real-time applications like voice translators or chatbots.

6. Model Size & Efficiency

- Especially critical when deploying on mobile or low-resource devices.
- Smaller models like DistilBERT or TinyML versions of NMT models trade off some accuracy for speed and size.

7. Language Coverage

- Ability to translate between many language pairs.
- Important for models used in global applications.

4. CODE

```
import numpy as np

from flask import Flask, render_template_string, request, jsonify

from transformers import MarianMTModel, MarianTokenizer

from langdetect import detect

import os
```

```
app = Flask(__name__)



# Preload models and tokenizers at startup

model_mapping = {

    'English': "Helsinki-NLP/opus-mt-mul-en",

    'Spanish': "Helsinki-NLP/opus-mt-en-es",

    'French': "Helsinki-NLP/opus-mt-en-fr",

    'German': "Helsinki-NLP/opus-mt-en-de",

    'Chinese': "Helsinki-NLP/opus-mt-en-zh",

    'Hindi': "Helsinki-NLP/opus-mt-en-hi"

}



models = {lang: MarianMTModel.from_pretrained(model_name) for lang, model_name in
model_mapping.items()}

tokenizers = {lang: MarianTokenizer.from_pretrained(model_name) for lang, model_name in
model_mapping.items()}



# Placeholder language list for UI
```

```
languages = list(model_mapping.keys())



def translate_text(input_text, target_language):
    """Translate the input text to the target language."""
    try:
        # Step 1: Detect the source language
        source_language = detect(input_text)



        # Step 2: If the source language is not English, translate to English first
        if source_language != "en":
            intermediate_model = models.get("English")  # Model for translating to English
            intermediate_tokenizer = tokenizers.get("English")
            if not intermediate_model or not intermediate_tokenizer:
                return "Translation model for intermediate step not available."
            inputs = intermediate_tokenizer.encode(input_text, return_tensors="pt", truncation=True)
            outputs = intermediate_model.generate(inputs, max_length=40, num_beams=4,
early_stopping=True)
            input_text = intermediate_tokenizer.decode(outputs[0], skip_special_tokens=True)
```

```
        # Step 3: Translate from English to the target language
        model = models.get(target_language)
        tokenizer = tokenizers.get(target_language)
        if not model or not tokenizer:
            return "Translation model not available for the selected language."
        inputs = tokenizer.encode(input_text, return_tensors="pt", truncation=True)
        outputs = model.generate(inputs, max_length=40, num_beams=4, early_stopping=True)
        return tokenizer.decode(outputs[0], skip_special_tokens=True)

    except Exception as e:
        return f"Error during translation: {str(e)}"

def translate_text(input_text, target_language):
    """Translate the input text to the target language."""
    try:
        model = models.get(target_language)
        tokenizer = tokenizers.get(target_language)
        if not model or not tokenizer:
```

```
            return "Translation model not available for the selected language."
        inputs = tokenizer.encode(input_text, return_tensors="pt", truncation=True)
        outputs = model.generate(inputs, max_length=40, num_beams=4, early_stopping=True)
        return tokenizer.decode(outputs[0], skip_special_tokens=True)
    except Exception as e:
        return f"Error during translation: {str(e)}"

# HTML template with Bootstrap for UI
html_template = """
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Language Translator</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" />
    <style>
        body {
```

```
    background: linear-gradient(135deg, #667eea, #764ba2);
    color: white;
    min-height: 100vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    padding: 2rem;
}
.translator-container {
    background: rgba(0,0,0,0.6);
    padding: 2rem;
    border-radius: 15px;
    max-width: 600px;
    width: 100%;
    box-shadow: 0 8px 32px 0 rgba(31, 38, 135, 0.37);
}
textarea {
```

```
    resize: none;
}
.result-box {
    background: #222;
    padding: 1rem;
    border-radius: 10px;
    margin-top: 1rem;
    min-height: 100px;
    white-space: pre-wrap;
    font-size: 1.2rem;
}
h1 {
    margin-bottom: 1rem;
    font-weight: 700;
    text-align: center;
}
.spinner-border {
    display: none;
```

```
        }
    </style>
</head>
<body>
    <div class="translator-container">
        <h1>Language Translator</h1>
        <form id="translateForm">
            <div class="mb-3">
                <label for="inputText" class="form-label">Enter text to translate:</label>
                <textarea class="form-control" id="inputText" rows="4" required></textarea>
            </div>
            <div class="mb-3">
                <label for="languageSelect" class="form-label">Select target language:</label>
                <select class="form-select" id="languageSelect" required>
                    <option value="" disabled selected>Select language</option>
                    {% for lang in languages %}
                    <option value="{{ lang }}">{{ lang }}</option>
                    {% endfor %}
```

```
            </select>
        </div>
        <button type="submit" class="btn btn-primary w-100">Translate</button>
    </form>
    <div id="sourceLanguage" class="mt-3"></div>
    <div id="loadingSpinner" class="spinner-border text-light mt-3" role="status">
        <span class="visually-hidden">Loading...</span>
    </div>
    <div id="result" class="result-box mt-3"></div>
</div>

<script>
    const form = document.getElementById('translateForm');
    const resultBox = document.getElementById('result');
    const sourceLanguageBox = document.getElementById('sourceLanguage');
    const loadingSpinner = document.getElementById('loadingSpinner');

    form.addEventListener('submit', async (e) => {
```

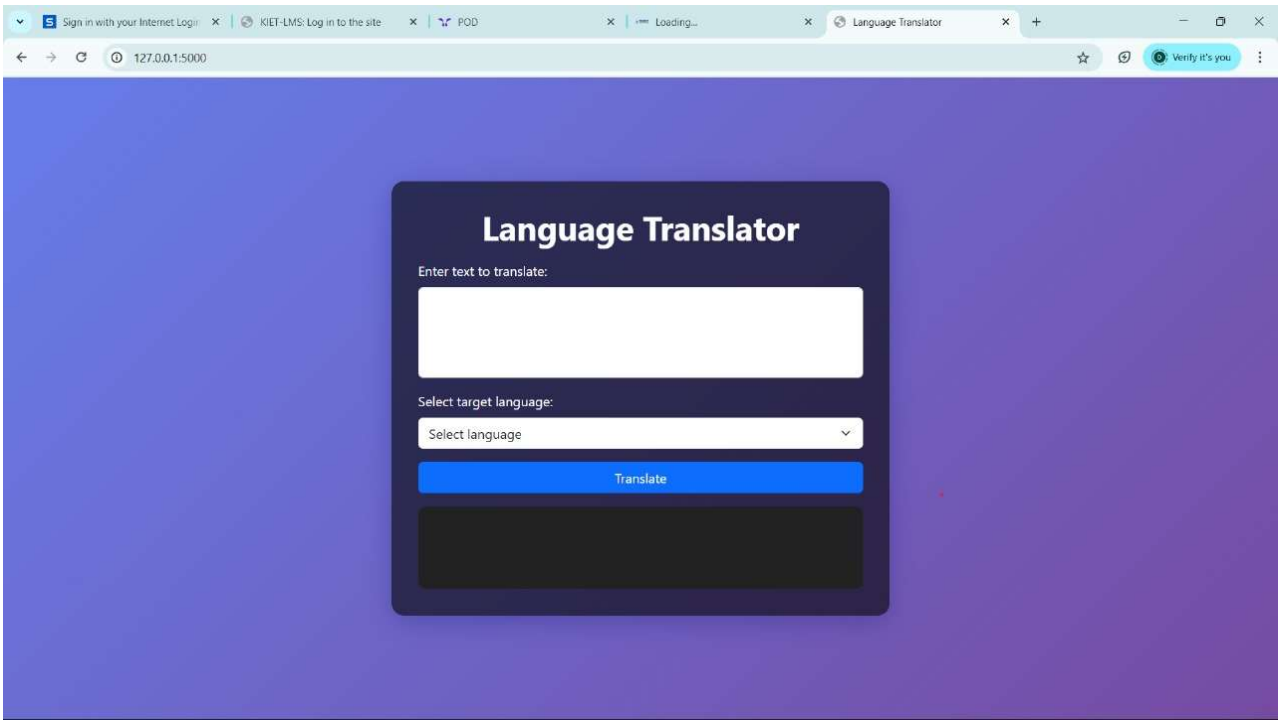
```
e.preventDefault();

const inputText = document.getElementById('inputText').value.trim();

const targetLanguage = document.getElementById('languageSelect').value;



if (!inputText || !targetLanguage) {

    resultBox.textContent = 'Please enter text and select a language.';

    return;

}



resultBox.textContent = '';

sourceLanguageBox.textContent = '';

loadingSpinner.style.display = 'block';



try {

    const response = await fetch('/translate', {

        method: 'POST',

        headers: { 'Content-Type': 'application/json' },

        body: JSON.stringify({ text: inputText, language: targetLanguage })
```

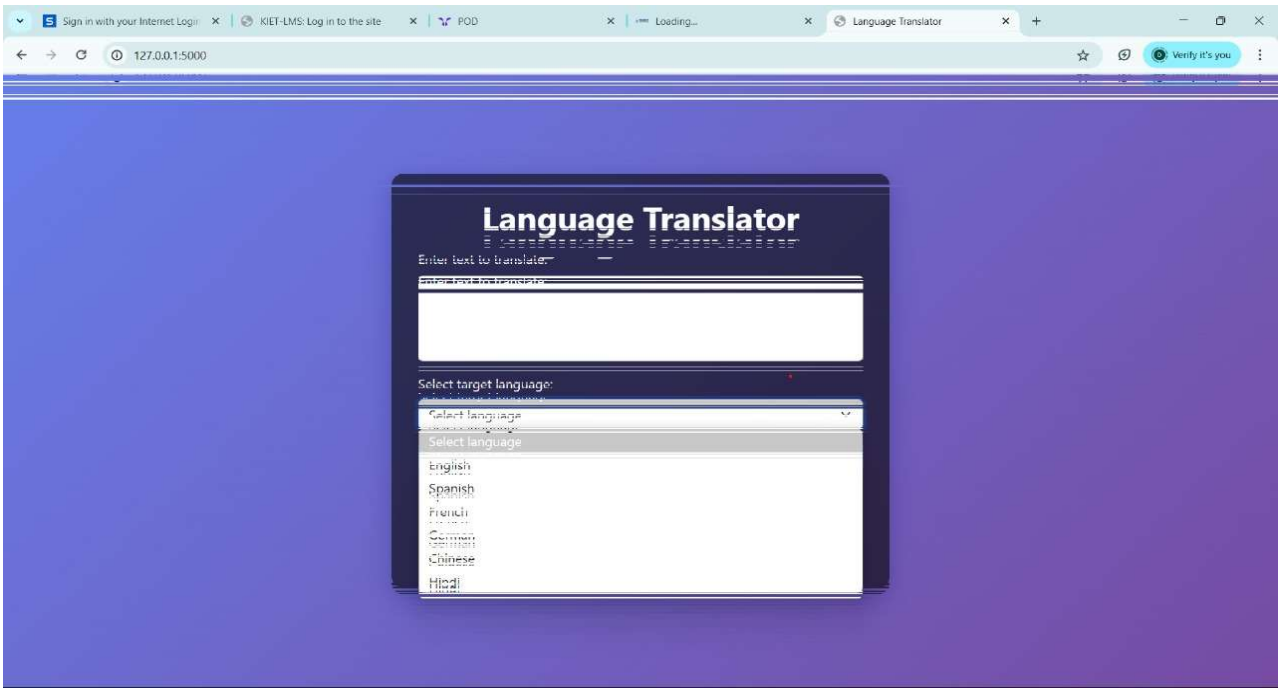
```
            });

            const data = await response.json();

            loadingSpinner.style.display = 'none';



            if (data.translated_text) {

                resultBox.textContent = data.translated_text;

                sourceLanguageBox.textContent = `Detected Source Language:
${data.source_language}`;

            } else {

                resultBox.textContent = 'Translation failed.';

            }

        } catch (error) {

            loadingSpinner.style.display = 'none';

            resultBox.textContent = 'Error occurred during translation.';

        }

    });

  </script>

</body>

</html>
```

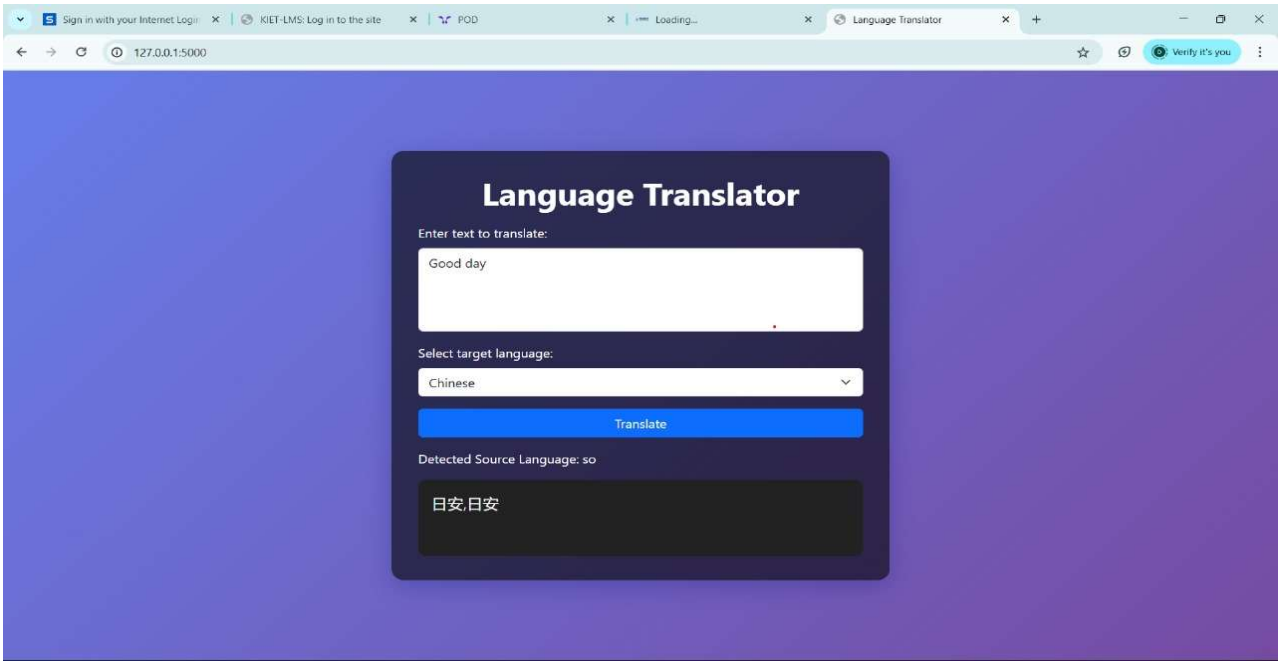
```
"""

@app.route('/')
def index():
    return render_template_string(html_template, languages=languages)

@app.route('/translate', methods=['POST'])
def translate():
    data = request.get_json()
    input_text = data.get('text', '')
    target_language = data.get('language', '')
    if not input_text or not target_language:
        return jsonify({'error': 'Invalid input'}), 400

    # Perform translation
    translated_text = translate_text(input_text, target_language)
    return jsonify({'source_language': detect(input_text), 'translated_text': translated_text})
```

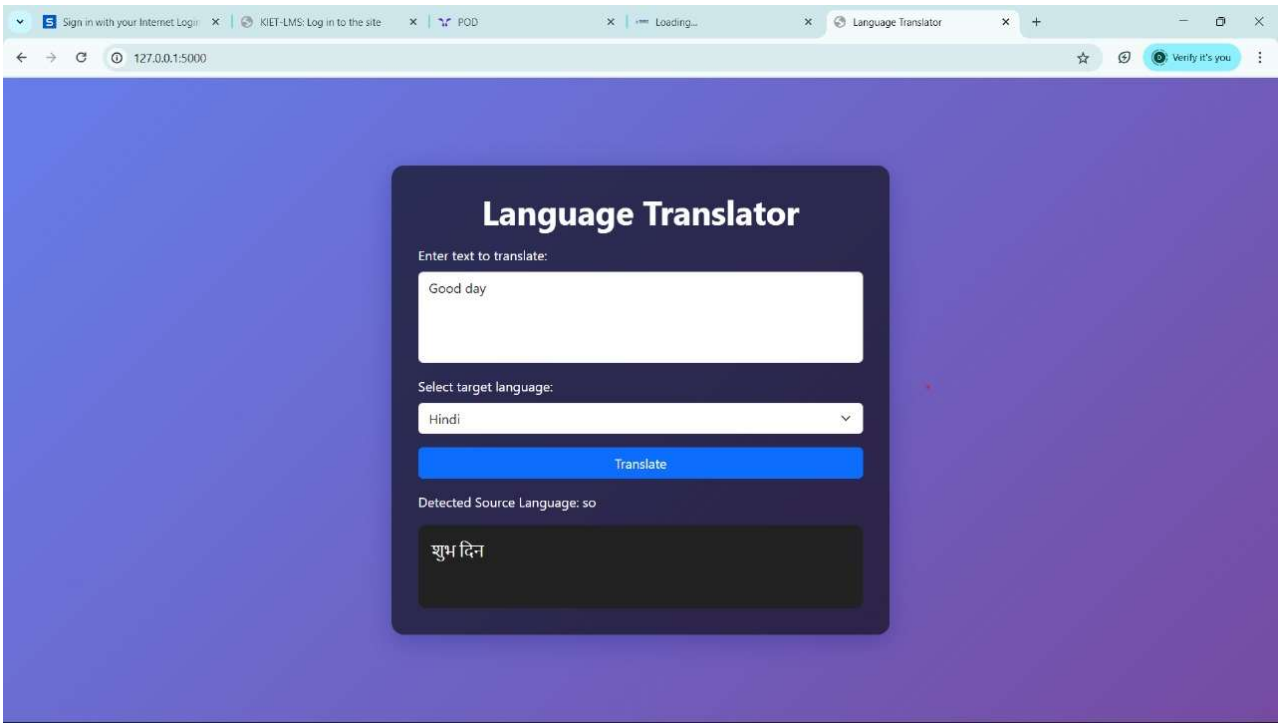
```
if __name__ == '__main__':

    app.run(debug=True)
```

5. OUTPUT




Sign in with your Internet Login
KIET-LMS: Log in to the site
POD
Loading...
Language Translator
127.0.0.1:5000
Verify it's you
Language Translator
Enter text to translate:
Select target language:
Select language
Select language
English
Spanish
French
German
Chinese
Hindi


Sign in with your Internet Login
KIET-LMS: Log in to the site
POD
Loading...
Language Translator
127.0.0.1:5000
Verify it's you
Language Translator
Enter text to translate:
Good day
Select target language:
Chinese
Translate
Detected Source Language: so
日安,日安



6. CONCLUSION

Language translation using sequence models has made tremendous advancements over the past few years, driven by deep learning techniques like **Encoder-Decoder** architectures, **Attention Mechanisms**, and **Transformers**. These models have transformed machine translation from rule-based systems to data-driven, context-aware solutions. With the introduction of **Neural Machine Translation (NMT)** and especially **Transformer-based models**, translation quality has reached new heights in terms of fluency, accuracy, and context understanding.

The adoption of sequence models has made language translation more accessible and efficient. Models like **Google Translate** and **DeepL** are now widely used, facilitating global communication and knowledge exchange. These models have proven invaluable in various industries, including education, healthcare, e-commerce, and customer service, by allowing businesses and individuals to communicate across language barriers in real-time. Additionally, they play a crucial role in **content localization**, enabling companies to reach international markets more effectively.

However, challenges remain, particularly around low-resource languages and the potential for errors in highly specialized or context-dependent translations. Despite these limitations, the continual improvement of sequence models and the integration of **pretrained models** such as **MarianMT** and **mBART** are pushing the boundaries of what is possible.

In conclusion, language translation with sequence models is not only enhancing global communication but also paving the way for future AI applications that can understand and process human language with unprecedented accuracy and fluency. The future of machine translation

promises even more sophisticated, adaptable, and context-aware systems that will further bridge the language divide.

7. REFERENCES

1. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017).
"Attention Is All You Need"
Advances in Neural Information Processing Systems (NeurIPS)
https://arxiv.org/abs/1706.03762
2. Bahdanau, D., Cho, K., & Bengio, Y. (2014).
"Neural Machine Translation by Jointly Learning to Align and Translate"
arXiv preprint
https://arxiv.org/abs/1409.0473
3. Sutskever, I., Vinyals, O., & Le, Q. V. (2014).
"Sequence to Sequence Learning with Neural Networks"
NeurIPS
https://arxiv.org/abs/1409.3215
4. Hugging Face Transformers Library
https://huggingface.co/transformers/
5. MarianMT - Multilingual Translation Models
https://huggingface.co/Helsinki-NLP
6. Langdetect – Language Detection Library
https://pypi.org/project/langdetect/