Natural Language Processing

Mini Project

Submitted By
Tagadghar Shailesh Ashok

Roll No : 31031523034
MSc CS – Sem II. Part I
Academic Year : 2023 – 24

Department Of Computer Science
Somaiya Vidyavihar University
SK Somaiya college

Title : Text Translator Service

Introduction :

This documentation provides an overview of a Flask-based Translator Service, which utilizes Google Translate API for text translation. The service allows users to input text in one language and obtain its translation in a desired target language.

The Translator Service offers a seamless solution for translating text across multiple languages. Leveraging the power of Google Translate API, users can effortlessly convert text from one language to another with accuracy and speed. This service caters to individuals and organizations seeking efficient language translation capabilities for various purposes, from communication to content localization. With its user-friendly interface and robust functionality, the Translator Service simplifies the process of multilingual communication.

Scope :

The Translator Service aims to offer a simple yet effective solution for translating text between various languages. It caters to users who require quick and accurate translations for personal or professional purposes.

The Translator Service is designed to meet the needs of users requiring swift and reliable text translation services. It provides a convenient platform for translating text between numerous languages, catering to both personal and professional use cases. From individuals seeking to communicate across language barriers to businesses aiming to localize content for global audiences, the Translator Service offers a versatile solution for seamless language translation needs.

Software :

1. Python 3.12
2. Flask
3. Googletrans
4. HTML/CSS
5. JavaScript
6. VS Code

Algorithm Implemented :

The primary algorithm implemented in this service involves sending text to Google Translate API and receiving the translated text in response. This is facilitated by the translate_text() function, which utilizes the Googletrans library.

The Translator Service utilizes a straightforward algorithm that interacts with the Google Translate API. When a user submits text for translation, the service sends a request to the API with the provided text and the target language. The API processes the request and returns the translated text, which is then displayed to the user. This simple yet effective approach ensures accurate and efficient language translation within the application.

Library :

1. Flask: A micro web framework for Python.

2. Googletrans: A Python library for Google Translate API.

Dataset :

The Translator Service does not rely on a specific dataset as it dynamically translates user-provided text using Google Translate API.

The Translator Service does not rely on a specific dataset. Instead, it dynamically interacts with the Google Translate API, which internally utilizes vast language models and datasets to perform accurate translations. Users input text directly into the service, which then sends requests to the API for translation. This approach ensures that the Translator Service can handle translations for a wide range of languages without requiring its own dataset.

Code :

Translator_server.py :

```
#pip install Flask SpeechRecognition googletrans==4.0.0-rc1


from flask import Flask, request, jsonify
from googletrans import Translator

app = Flask(__name__)

def translate_text(text, target_language):
    translator = Translator()
    translation = translator.translate(text, dest=target_language)
    return translation.text

@app.route('/')
def index():
    # return open('index.html').read()
    return open('index.html', encoding='utf-8').read()

@app.route('/translate', methods=['POST'])
def translate():
    data = request.json
    text_to_translate = data['text_to_translate']
    target_language = data['target_language']
    translated_text = translate_text(text_to_translate, target_language)
    return jsonify({'translated_text': translated_text})
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

index.html :

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Translator</title>
</head>

<body>
    <h1>Translator</h1>
    <div id="content-wrapper">
        <div id="form-wrapper">
            <form id="translation-form">
                <label for="text-to-translate">Enter text to translate :
</label><br><br>
                <textarea id="text-to-translate" name="text-to-translate"
rows="4" cols="50"></textarea><br><br>
                <label for="target-language">Select target language :
</label><br><br>
                <select id="target-language" name="target-language">
                    <option value="mr">Marathi</option>
                    <option value="hi">Hindi</option>
                    <option value="af">Afrikaans</option>
                    <option value="ak">Akan</option>
                    <option value="sq">Albanian</option>
                    <option value="am">Amharic</option>
                    <option value="ar">Arabic</option>
                    <option value="hy">Armenian</option>
                    <option value="as">Assamese</option>
                    <option value="ay">Aymara</option>
                    <option value="az">Azerbaijani</option>
                    <option value="bm">Bambara</option>
                    <option value="bn">Bangla</option>
                    <option value="eu">Basque</option>
                    <option value="be">Belarusian</option>
                    <option value="bh">Bhojpuri</option>
                    <option value="bs">Bosnian</option>
                    <option value="bg">Bulgarian</option>
```

```
                    <option value="my">Burmese</option>
                    <option value="ku">Kurdish</option>
                    <option value="ky">Kyrgyz</option>
                    <option value="lo">Lao</option>
                    <option value="la">Latin</option>
                    <option value="lv">Latvian</option>
                    <option value="ln">Lingala</option>
                    <option value="lt">Lithuanian</option>
                    <option value="lb">Luxembourgish</option>
                    <option value="mk">Macedonian</option>
                    <option value="mai">Maithili</option>
                    <option value="mg">Malagasy</option>
                    <option value="ms">Malay</option>
                    <option value="ug">Uyghur</option>
                    <option value="uz">Uzbek</option>
                    <option value="vi">Vietnamese</option>
                    <option value="cy">Welsh</option>
                    <option value="fy">Western Frisian</option>
                    <option value="xh">Xhosa</option>
                    <option value="yi">Yiddish</option>
                    <option value="yo">Yoruba</option>
                    <option value="zu">Zulu</option>
                </select><br><br>
                <button type="button" onclick="translateText()"
id="button">Translate</button>
            </form>
        </div>

        <div id="translated-text-wrapper">
            <label for="translated-text">Translated Text : </label><br><br>
            <textarea id="translated-text"></textarea>
        </div>
    </div>

    <script>
        function translateText() {

            var button = document.getElementById("button");
            button.innerHTML = "Translating..."; // Change button text to
indicate translation is in progress

            var textToTranslate = document.getElementById("text-to-
translate").value;
            var targetLanguage = document.getElementById("target-
language").value;

            fetch('/translate', {
                method: 'POST',
```

```
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify({
                    text_to_translate: textToTranslate,
                    target_language: targetLanguage
                }),
            })
                .then(response => response.json())
                .then(data => {
                    document.getElementById("translated-text").innerText =
data.translated_text;
                    button.innerHTML = "Translate";
                })
                .catch((error) => {
                    console.error('Error:', error);
                    button.innerHTML = "Translate";
                });
        }
    </script>

</body>

</html>
```

Index.css :

```
body {
    text-align: center;
    background-color: azure;
    display: flex;
    flex-direction: column;
    align-items: center;
}

h1 {
    color: blueviolet;
    font-size: 50px;
}

label {
    color: rgb(103, 86, 86);
    font-size: 30px;
}

#button {
    height: 40px;
```

```
    width: 180px;
    background-color: lightskyblue;
    font-size: 20px;
    color: black;
    border-radius: 5px;
    border-style: hidden;
    display: inline-flex;
    align-items: center;
    justify-content: center;
    cursor: pointer;
}

#button:hover {
    background-color: bisque;
}

#target-language {
    height: 40px;
    width: 190px;
}

select {
    font-size: 20px;
    text-align: center;
    /* background-color: lightskyblue; */
    border-style: hidden;
    border-radius: 5px;
}

#text-to-translate {
    font-size: 20px;
    width: 600px;
    height: 200px;
    overflow: auto;
    border: 1px solid black;
    border-style: hidden;
}

#translated-text {
    color: black;
    border: 1px solid black;
    border-style: hidden;
    font-size: 20px;
    width: 600px;
    height: 200px;
    overflow: auto;
    background-color: white;
}
```

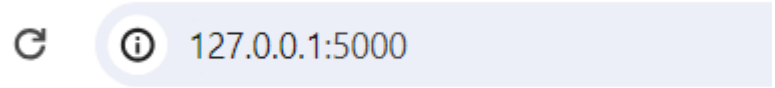
```css
#content-wrapper {
    display: flex;
    justify-content: space-around;
    width: 100%;
    position: relative;
}

#form-wrapper {
    flex: 1;
}

#translated-text-wrapper {
    flex: 1;
    border-left: 2px solid red;
    padding-left: 20px;
    margin-left: 20px;
}

.refresh-icon {
    width: 20px;
    height: 20px;
    margin-left: 5px;
}
```

To run this file :

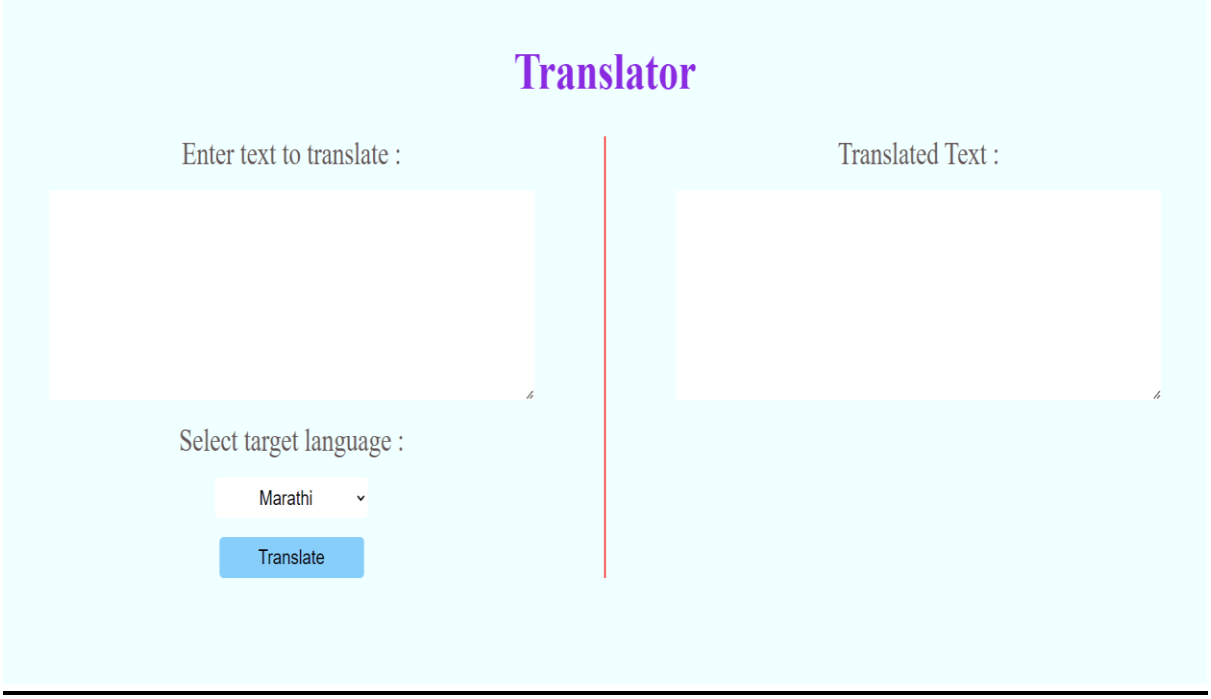
- Open new terminal
- Type this command given below,
- python translator_server.py

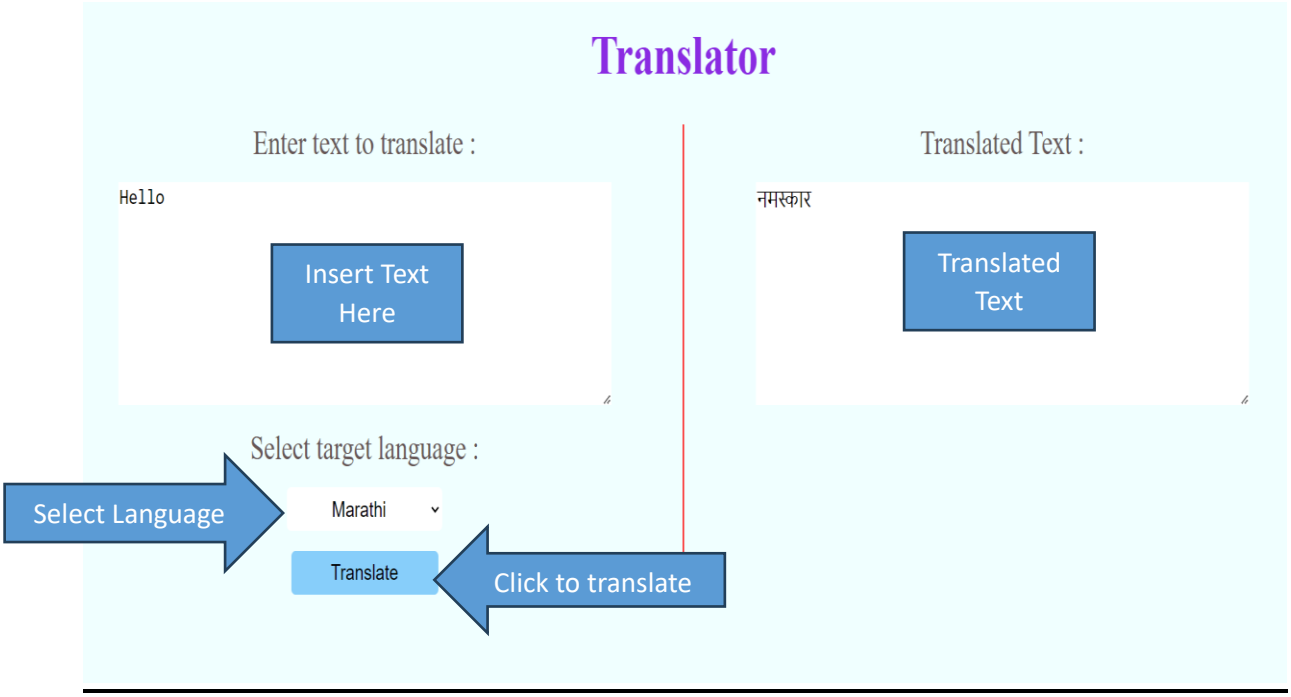
```
PS D:\Somaiya\NLP\Translator_Project\NewP> python translator_server.py
 * Serving Flask app 'translator_server'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 786-770-011
```

- It will run on http://127.0.0.1:5000/



Outcome Screenshot :





Further Enhancement :

Possible enhancements for the Translator Service include:

- Adding support for additional languages.
- Implementing user authentication for personalized translation history.
- Enhancing the user interface for better user experience.
- Adding Voice module to Convert Text into specified language.

Conclusion :

The Translator Service provides a convenient solution for text translation needs. With its simple interface and integration with Google Translate API, it offers users a straightforward way to translate text between various languages.

The Translator Service offers a user-friendly and efficient solution for language translation needs. Leveraging the power of Google Translate API, it provides accurate translations across multiple languages, catering to a wide range of users. With its intuitive interface and potential for further enhancements, the Translator Service stands as a valuable tool for facilitating global communication and bridging language barriers in today's interconnected world.

References :

1. Google
2. Geeks for Geeks
3. Chat GPT
4. GitHub