AIMLCZG519 - Group 74

Yathisha A S	2022ac05237@wilp.bits-pilani.ac.in
Rudradityo Saha	2022aa05017@wilp.bits-pilani.ac.in
Nilanjana Roy	2022ac05393@wilp.bits-pilani.ac.in
Kowsalya S	2022ac05701@wilp.bits-pilani.ac.in

Design Choices

Post extracting the uploaded zip folder namely *AIMLCZG519_Group74_Problem18* and changing into the unzipped folder, you will find the flask app folder namely *nlpa-assignment-2*.

The flask app folder namely *nlpa-assignment-2* contains

- *nmt-app.py*
    - Python file containing application logic
- *requirements.txt*
    - Text file listing required dependencies
- *Sample Files to Upload* directory
    - Contains sample input text files to check flask app flow
- *static/css/site.css*
    - CSS file to style flask app webpage
- *templates/index.html*
    - HTML file to create flask app webpage
- *venv* directory
    - Contains the Python Virtual Environment
- *__pycache__* directory
    - Contains compiled bytecode file

Challenges Faced during Implementation

- Lemmatization did not happen as expected during English text preprocessing as POS Tagging was not done before Lemmatization leading to technical jargons not being removed from the English text unless that exact same word is listed in the technical jargon list. This led to more challenging and potentially inaccurate translation.

- The implementation of dynamic aspects of flask webpage using JavaScript was quite challenging due to the intricate logic involved in allowing users either to type text in textbox or upload text file but not both and the connecting logic carefully placed in Python file to capture the text from either textbox or text file for preprocessing and translation

- The CSS code had to be meticulously implemented to ensure that all the selectors are able to identify the intended elements so as to ensure that the webpage presents an aesthetic look.

- User can only upload one text file at a time in a session. The scenario where a user can upload multiple text files in a session is not implemented.

- Input Validation is not implemented

Application Frontend Flow

Assuming that the flask web app is running on http://localhost:5000, on opening the same, you will view the following webpage:

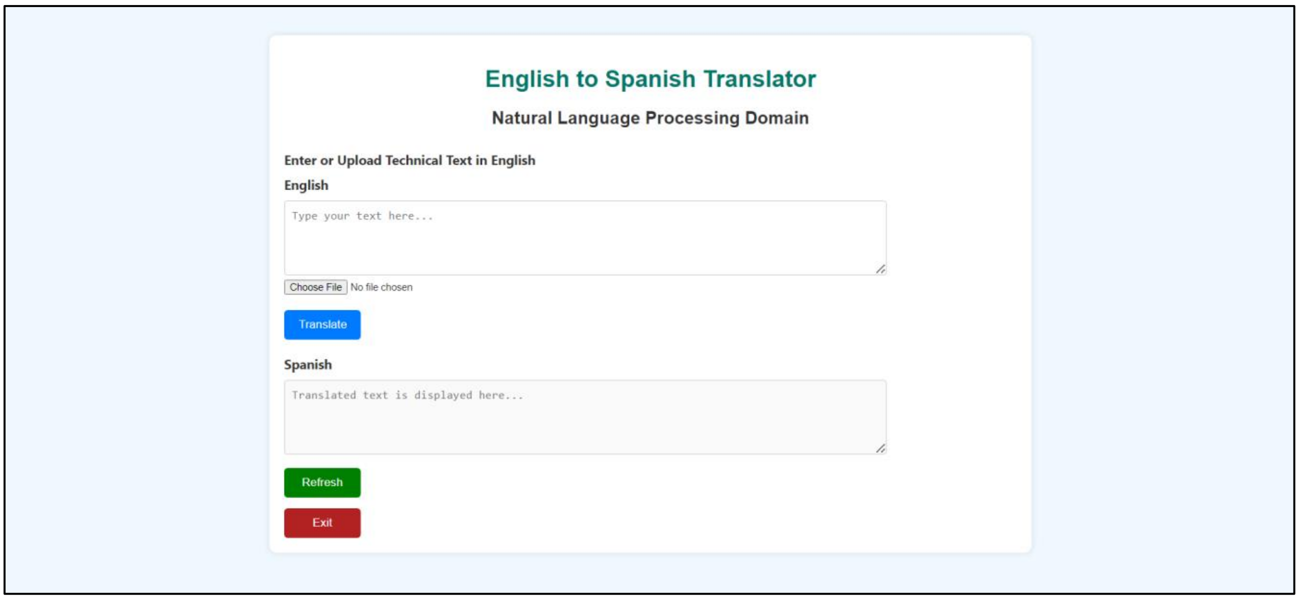

English to Spanish Translator
Natural Language Processing Domain
Enter or Upload Technical Text in English
English
Type your text here...
Choose File
No file chosen
Translate
Spanish
Translated text is displayed here...
Refresh
Exit

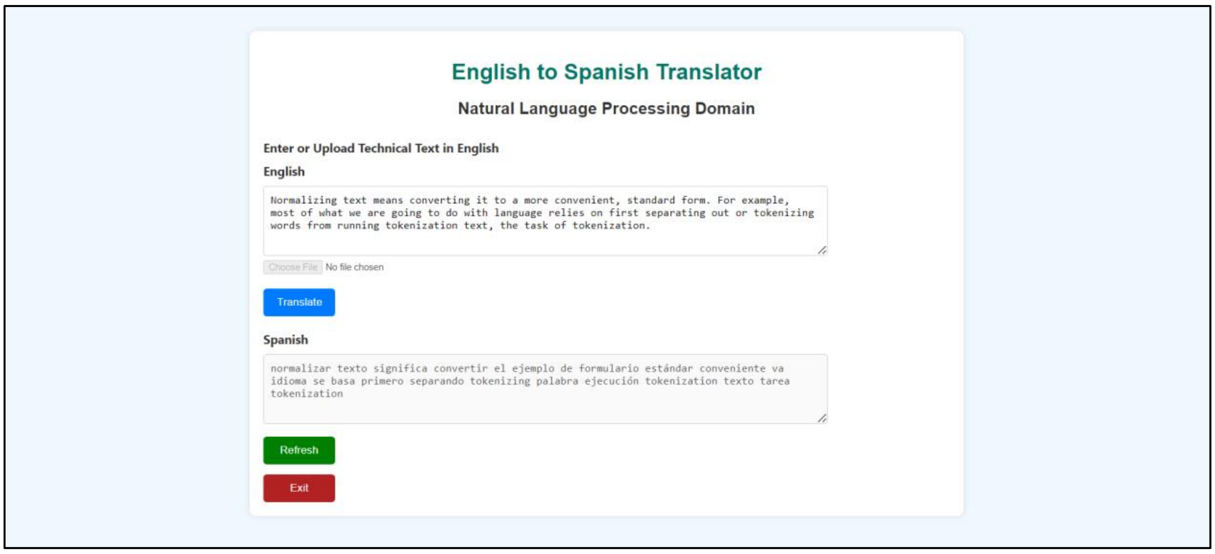
Figure 1: English to Spanish Translator Webpage

Summary

The *English to Spanish Translator System* Webpage translates English text given as input by user, to Spanish text which gets displayed in *Spanish* textbox.

Steps

1. The user can either type English text in *English* textbox or upload a text file with English content by clicking on the *Choose File* button
2. Once the input is provided, the user needs to click on the *Translate* button to obtain translated Spanish text
3. After waiting for a few seconds, the translated Spanish text becomes visible in the *Spanish* textbox



English to Spanish Translator

Natural Language Processing Domain

Enter or Upload Technical Text in English

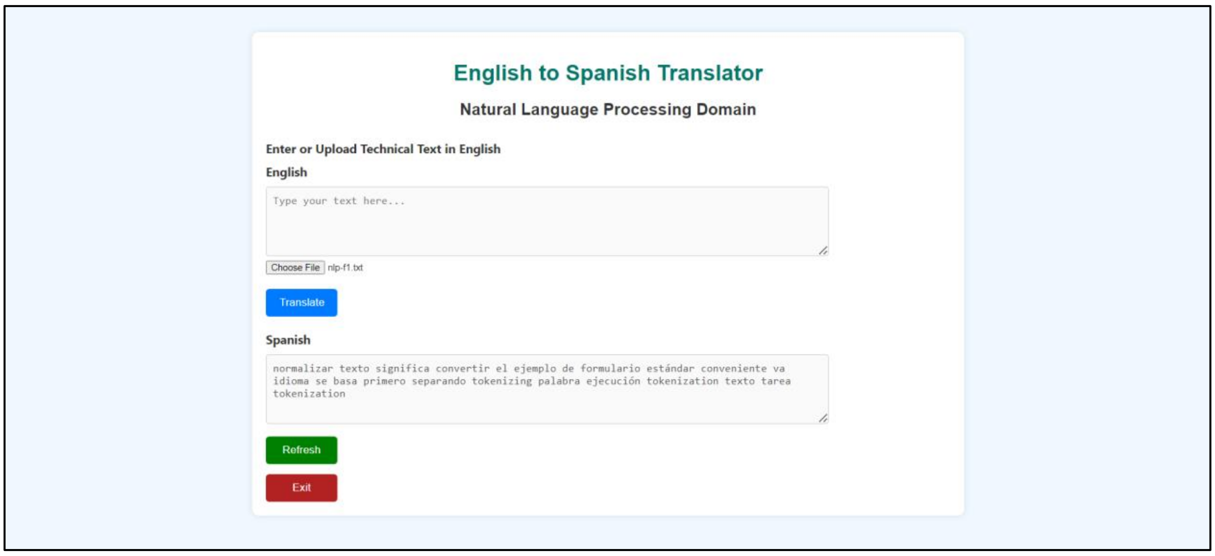
English

```
Normalizing text means converting it to a more convenient, standard form. For example,
most of what we are going to do with language relies on first separating out or tokenizing
words from running tokenization text, the task of tokenization.
```

Spanish

```
normalizar texto significa convertir el ejemplo de formulario estándar conveniente va
idioma se basa primero separando tokenizing palabra ejecución tokenization texto tarea
tokenization
```

Figure 2: Input via Textbox and Output



English to Spanish Translator

Natural Language Processing Domain

Enter or Upload Technical Text in English

English

```
Type your text here...
```

Spanish

```
normalizar texto significa convertir el ejemplo de formulario estándar conveniente va
idioma se basa primero separando tokenizing palabra ejecución tokenization texto tarea
tokenization
```

Figure 3: Input via Text File Upload and Output

Note

- The Heading *Natural Language Processing Domain* indicates that the *English to Spanish Translator System* is expecting technical English texts in Natural Language Processing Domain.
- User can either type English text in *English* textbox or upload a text file with English content by clicking on the *Choose File* button but not both
  - When a user types anything on the textbox, the *Choose File* button gets faded out indicating that the same is disabled and thereby unclickable

    - When a user uploads a file via *Choose File* button
        - The name of the uploaded file along with the file extension, gets visible on the right-hand side of the *Choose File* button
        - The *English textbox* gets faded out indicating that the same is disabled and thereby unclickable
- Clicking on the *Refresh* button reloads the webpage thereby clearing out the typed text or uploaded file in webpage form
- Clicking on The *Exit* button pops up an alert stating *You have exited the form.*
    - Post clicking on *OK*, the webpage turns blank, displaying only the background colour as shown in Figure 4

Figure 4: Blank Webpage