ITCS 6112 - Software Systems Design and Implementation

Final Project

Documentation and ReadMe

<u>TEAM NAME : THE BRAIN STORMERS</u>

Document number 1
April 2023
Version 1

Kanchan Chandnani (801293770)
Prachi Shende (801306878)
Ramakanth Ayalasomajula (801310383)
Sourabh Chaudhari (801286005)

UNIVERSITY OF NORTH CAROLINA AT CHARLOTTE

Overview

The Language Translator is an efficient tool to translate, detect, break sentence and a few more functionalities that are useful to overcome language barrier. Parts of Natural Language Processing and Post Processing of text/corpus implements the translations and detection techniques before parsing. This project covers the major language processing functionalities in the form of APIs. We have created a visual representation of this APIs for the users to utilize the features. The features are self explanatory and simple to use.

Installation

System Requirements :

Chrome Browser : Version 110.0.5481.177 (Official Build) (arm64) or higher
NodeJS : v16.17.1 or higher
Visual Studio Code : version 1.72 or higher

Mac Users and Windows Users

Steps to runt he project :
Step 1 : Download the "SSDIFinalProject.zip" file on the system
Step 2 : Unzip the file into "SSDIFinalProject" folder.
Step 3 : Open the "SSDIFinalProject" on VSCode.
Step 4 : Open terminal
Step 5 : Run command "node server.js"
It will throw package to found error. This means Node.js needs to be install in your system.
Install node.js on your system and enable environment variables for node. Follow the link to for installation

https://radixweb.com/blog/installing-npm-and-nodejs-on-windows-and-mac

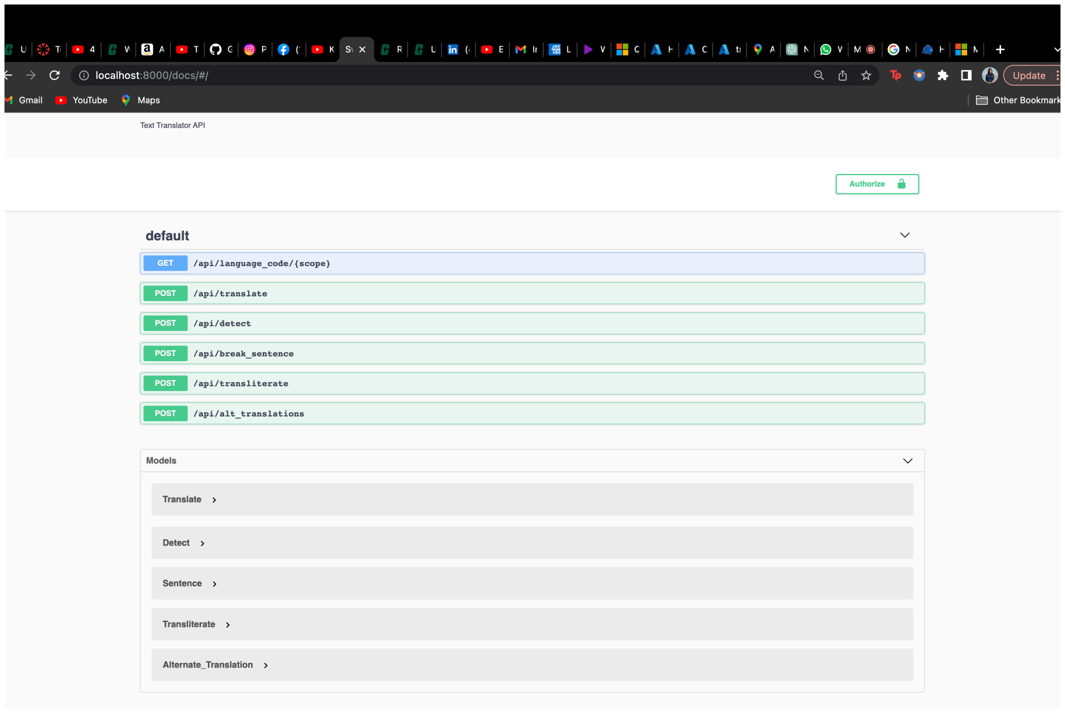
https://stackoverflow.com/questions/23412938/node-is-not-recognized-as-an-internal-or-an-external-command-operable-program

Step 6 : Run "npm install" to install all the package.json file packages into the project.
Once all the packages are installed from package.json file, run step 5. This time the project should run on localhost you should be able to hit the "http://localhost:8000/docs/#/" on the chrome web browser.

Step 7 : "http://localhost:8000/docs/#/" will show the UI.
Step 8 : Refer to the below API documentation and use the sample request and response provided to test the application.

Guidelines to use the UI

UI Screenshot



default

GET /api/language_code/{scope}
POST /api/translate
POST /api/detect
POST /api/break_sentence
POST /api/transliterate
POST /api/alt_translations

Models

Translate
Detect
Sentence
Transliterate
Alternate_Translation

We have 5 language APIs on the UI each with a functionality of its own.

1. Translator

POST api/translate
Url : http://localhost:8000/api/translate

Description:

- This REST endpoint is used to translate the provided input 'text' to the desired language which is provided as an input body parameter in 'to'.
- /api/translate will auto-detect the language of the text provided in the body and indicates the confidence score of the detected language.

Request body:

- The body of the request is a JSON object.
- The JSON object has a string property named ‘**text’**, which represents the string to translate.
- The other string property which is an input param in the JSON request body is ‘**to’** which specifies the language of the output text.
- The target language must be one of the supported language codes or set of languages supported to translate text
- included in the translation scope.
- For example, use "to": "es" to translate the input text to Spanish.
- It is possible to translate to multiple languages simultaneously by passing an array to the ‘to’ input param.

  For example, use "to": ["it", "de"] to translate the same input text to Italian and German simultaneously.

Example JSON Request Body:

JSON Request

```
{
  "text": "hello",
  "to": ["it", "de"]
}
```

Response body:

- A successful response is a JSON array with a 200 Success status.
- A result object includes the following properties [1]:
- detectedLanguage: An object that describes the detected language with the following properties:
  - ▪ language: A string representing the code or codes of the detected language.
  - ▪ score: A score is a float value that indicates the confidence in the result of the language detected. This score is between zero(low) and one(high). A lower score indicates lower confidence in the result of the detected language.

- translations: The JSON array represents the result of each target language translation specified through the 'to' request body parameter. Each element in the array includes the following properties:

  - ▪ text: A string output which gives the translation of the input text to the target language code.
  - ▪ to: A string that represents the language code of the output target language.

- An unsuccessful request returns the error with a 400 Bad Request status.

JSON Response

```
[
 {
  "detectedLanguage": {
   "language": "en",
   "score": 1
  },
  "translations": [
   {
    "text": "Ciao",
    "to": "it"
   },
   {
    "text": "Hallo",
    "to": “de"
   },
  ]
 }
```

2. LANGUAGE DETECT

POST api/detect
Url : http://localhost:8000/api/detect

Description:

- This REST endpoint identifies the language code of a piece of text passes to it as a request input parameter.
- /api/detect will auto-detect the language code of the text, provide a confidence score for the detected language, and list alternative languages with their confidence score.

Request body:

- The body of the request is a JSON object.
- The JSON object has a string property names '**text**', which represents the string whose language code is to be detected.
- The language auto-detection works better with longer input text.

JSON Request

```
{
   "text": “Good to see you”

}
```

Response body:

- A successful response is a JSON array with a 200 Success status.
- The result object includes the following properties:

  language : A string representing the code or output code of the detected language.
  score : A score is a float value that indicates the confidence in the result of the language detected. This score is between zero(low) and one(high). A lower score indicates lower

confidence in the result of the detected language.

alternatives : This is a JSON array of other possible languages detected from the input 'text' body parameter. Each element in the array includes the following properties:

language : Same as listed above

score : Same as listed above

- An unsuccessful request returns the error with a 400 Bad Request status.

JSON Response

```
[
  {
     "language": "en",
     "score": 1
     "alternatives": [
     {
       "language":"",
       “score": ""
     }
    ]
  }
]
```

3. Sentence Length

POST api/break_sentence
Url : http://localhost:8000/api/break_sentence

Description:

- This REST endpoint is used to identify the positioning of sentence boundaries for the input text provided.
- /api/break_sentence auto-detects the language code of the provided input text and outputs each sentence's length from the given input text.

Request body:

- The body of the request is a JSON Object.
- The JSON object has a string property named '**text**', which represents the string whose value is used to compute the sentence boundaries.

JSON Request

```
{
   "text": "Hello, how are you? Hope you are doing great!"
}
```

Response body:

- A successful response is a JSON array with a 200 Success status.
- A result object includes the following properties :

sentLen : An array of representing the sentence number and length of each sentence present in the text element. Also, array length represents the number of sentences present in the given input text.
detectedLanguage : A JSON object which describes the detected language with the

following properties:

<u>language</u> : A string representing the code of the detected language.

<u>score</u> : A score is a float value that indicates the confidence in the result of the language detected. This score is between zero(low) and one(high). A lower score indicates lower confidence in the result of the detected language.

• When an input text consists of mixed languages, the /api/break_sentence will detect the language that has a greater number of words and state it's confidence score.

• An unsuccessful request returns the error with a 400 Bad Request status.

JSON Response

```
[
  {
    "sentLen": [
      "Sentence 1 : 20",
      "Sentence 2 : 26",
      "Sentence 3 : 16"
    ],
     "detectedLanguage": {
     "language": "en",
     "score": 1
    }
  }
]
```

4. Transliterate

POST : api/transliterate

Url : http://localhost:8000/api/transliterate

Request body:

- The body of the request is a JSON object.
- The JSON object has a string property named ‘**text**’, which represents the string which needs a phonetic translation, that is, needs transliteration.
- Another required string property is ‘**language**’ which indicates the language code mentioned in the transliteration table or supported codes.
- The request body takes two other string properties ‘**fromScipt’** and ‘**toScript’** which indicates the name of the script supported by the transliteration table or supported scripts.

JSON Request

```
{
"text": "สวสั ด"ี ,

   "language": "th",
   "fromScript": "Thai",
   "toScript": "Latn"
}
```

Response body:

- A successful response is a JSON array with a 200 Success status.
- The result object includes the following properties :
    - text : The output string which is a phonetic translation or transliteration of the input text in the script passed in the ‘toScript’ property of the request body.
    - script : The script which is used to transliterate the input text to, that is, the value of the script mentioned in the ‘toScript’ property. This property is used to verify the script of the transliterated output text.

- An unsuccessful request returns the error with a 400 Bad Request status

JSON Response

```
[
  {
   “text” : "sawasat",
   “script" : "Latn"
  }
]
```

5. Alternative Translations

POST : api/alt_translations
Url : http://localhost:8000/api/alt_translations

Description:

- This REST endpoint provides alternate translations for a word or certain idiomatic phrases in the input language provided.
- /api/alt_translations responses with a list of alternate translations for the provided input text, in the input language provided and also lists the part-of-speech of the listed text in the translated language along with a list of back-translations in the original language of the text to provide context.

Request body:

- The body of the request is a JSON object.
- The JSON object has a string property names '**text',** which represents the sting which is to be translated.
- The '**from**' string property specifies the language code of the input text. It must be one of the language codes included in the dictionary scope.
- The '**to**' string property specifies the language code of the target output text i.e the language to which the input text is to be translated. It must be one of It must be one of the language codes included in the dictionary scope.

JSON Request

```
{
"text": "shark",
"from": "en",
“to": "es"
}
```

Response body:

- A successful response is a JSON array with a 200 Success status.
- If the term in the input text is not defined in the dictionary, the response is 200 (OK) but the result array will be empty.
- The result object includes the following properties :
    - normalizedTarget: A string with the normalized form of the output text translated in the provided input language.
    - displayTarget: A string which is a better representation of the normalizedTarget term for best end-user display.
    - postTag: This property indicates the part-of-speech of the input text in the translated language.

The following Tags represent the given part-of-speech:

TAG NAME	DESCRIPTION
ADJ	Adjective
ADV	Adverb
CONJ	Conjuctions
DET	Determiners
MODAL	Verbs
NOUN	Noun
PREP	Prepositions
PRON	Pronouns
VERB	Verbs
OTHER	Othera

- confidence : A float value that indicates the confidence in the result of the translated output text. This score is between zero(low) and one(high). A lower score indicates lower confidence in the translation pair.

- prefixWord : A string value, which is a gendered determination of nouns, in the provided language for translation if it has any gendered determinations. If there is no prefix, the output for this property is an empty string.

- backTranslations : This output a list of backtranslations of the text in context to the normazliedTarget value in the original language of the input text. Each element of the backTranslations list is an object that has the following properties:

  - normalizedText : A string with the normalized form of the source input text term which is backtranslated in the original language of the input text.

  - displayText : A string which is a better representation of the normalizedText term for best end-user display.

- numExamples : An integer value that represents the number of examples that are available, for the given translation pair of the input text and the target translated text, in the training model.

- frequencyCount : An integer value which represents the frequency of the given translation pair of the input text and the output translated text. This count can be used to sort the back-translations to get the most frequent pairs.

- An unsuccessful request returns the error with a 400 Bad Request status.

JSON Response

```
[
  {
   "normalizedTarget": “tiburón",
   “displayTarget” : "tiburón",
   "posTag": "OTHER",
   "confidence": 0.8182,
   "prefixWord": "",
   "backTranslations":
   [
     {
      "normalizedText": "shark",
      "displayText": “shark",
      "numExamples": 0,
      "frequencyCount": 45
     }
   ]
},
{
  "normalizedTarget": "escualo",
  "displayTarget": "escualo",
  "posTag": "NOUN",
  “confidence": 0.1818,
  "prefixWord": "",
  "backTranslations":
   [
     {
      "normalizedText": “shark",
      "displayText": "shark",
      "numExamples": 1,
      "frequencyCount": 10
     }
   ]
  }
]
```