A Project/Dissertation Report

on

English and Hindi Mixed Language Transliteration

Submitted in partial fulfilment of the

requirement for the award of the degree of

Bachelor of Technology

Under The Supervision of

Name of Supervisor: Dr. D Rajesh Kumar

Designation: Associate Professor

Submitted By

Pratham Mittal

18021180061/18SCSE1180063

Kartik Garg

18021180015/18SCSE1180016

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"ENGLISH AND HINDI MIXED LANGUAGE TRANSLITERATION"** in partial fulfillment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name… Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Pratham Mittal, 18SCSE1180063

Kartik Garg, 18SCSE1180016

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name

Designation

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Pratham Mittal: 18SCSE1180063 has been held on _____ and his/her work is recommended for the award of Bachelor of Technology.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Abstract

It is a fact that humans are social animals and, in their life, they love to contact their family and friends. Communication is a very necessary thing in today's lifestyle. Communication can be of various types, like chatting, news, notifications, etc. Whenever we talk on Facebook, WhatsApp, or Twitter we have seen that people are not very particular about expressing themselves in pure Hindi and pure English, but they tend to mix them up which troubles to a lot of people. Also, sometimes happens that people cannot figure out what to write in a formal way. So, we here are making a platform which will convert their mixed English and Hindi language to pure Devanagri form. Which will help them to make a better conversation and help them to increase their bonding. To make this happen we have made a dataset of English and Hinglish words each defined whether its english or hindi in front of them and their devanagri conversion of each. In this way, people can express themselves more freely and accurately.

List of Figures

Table of Contents

Chapter 1

Introduction

People often tend to write text in an unstructured manner. They write text in mixed language as per their comfort. But, sometimes it gets awkward while sending this text to some formal person. They need something that can solve this problem. Here our conversion software comes into place. It gets the input from the user in the forms of different languages and here our software converts it into Hindi in Devnagari syntax and in English with roman syntax.

And this process is important as we can see in India that majority of the people speak these languages that is Hindi and English and hence it makes itself very handy during the conversation. And here comes our software in place which will help the user to translate itself in these two languages.

Our software is more advanced than the existing technologies as we can see that the existing technology can only translate the languages when they know which language they want to convert from the available lot to . The language can then be translated but in our software the language is automatically detected and later it got translated.

Chapter 2

Literature Survey

M.M. Phadke et al., Multilingual machine translation: An analytical study, the paper presents a system that uses machine translation in an efficient manner. It takes many languages as input and can efficiently produce many language outputs. It uses various types of machine translations. Statistical machine translation, it works on statistical methods based on analysis of large bilingual datasets. It determines correspondence between words from source and output language. Rule based machine translation works on grammar. It analyses the sentences based on grammar. [1]

Melvin Johnson et al., Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, uses neural machine translation system. It uses a multilingual NMT model which can translate various types of languages with a single model improving the accuracy. It uses an encoder decoder model which contains of three parts, encoder, encoder vector and decoder. It also has zero-shot translation which happens during the multiple conversion of languages. [2]

Middi Venkata Sai Rishita et al., Machine translation using natural language processing, paper talks of using machine translation. It also considers various neural network architectures like simple recurrent neural networks, simple RNN with embedding, bidirectional RNN, and encoder decoder RNN. It forms a new neural network model that includes all of them and makes it more functional.[3]

Byung-Ju Kang et al., a bi-directional transliteration system is presented. It uses automatic english/korean bidirectional transliteration methodology. It also uses back transliteration technology. Uses a decision tree based approach. A character alignment algorithm based on the same is in use and it basically aligns two words in a desirable optimized way across languages. [4]

Krishna Regmi et al., paper focuses on research articles to be translated or transliterated for easy understanding. It determines the relevance of the context first, then it gives a forward translation. It performs backward translation if needed. It then examines the translated text's meaning in both the source and the target languages and finally revising the whole process for other other texts. [5]

Chapter 3

Technology Used

Natural language processing is a part of artificial intelligence which can give the ability of understanding text in the human form. It teaches a computer how to understand the human way. A Neural network is a set of algorithms that uses a network of functions to understand and translate the data from one form into desired output. Machine translation is the automatic translation of one language to another. It is used to convert large data into other language using a faster way. Each and every technology we use, we need to train them extensively so that the accuracy is near to 100%. Neural machine translation is the most important part of machine translation. It uses a large dataset to train the model to translate between two languages using deep learning. Also, LSTM, i.e., long short-term memory, is a type of recurrent neural network which has feedback connections. It makes small modifications to input information. The information flows through cell states and it selectively remember and forget information according to the desired output. This type of RNN is used in deep learning where a system needs to learn from experience. LSTM networks are commonly used in NLP tasks because they can learn the context required for processing sequences of data. To learn long-term dependencies, LSTM networks use a gating mechanism to limit the number of previous steps that can affect the current step.

Language Modeling (LM) is one of the most important parts of modern Natural Language Processing (NLP). Language model is required to represent the text to a form understandable from the machine point of view. The goal of a language model is to compute a probability of a token (e.g. a sentence or a sequence of words). Language Model (LM) actually a grammar of a language as it gives the probability of word that will follow.

o **Corpus** - Body of text, singular. Corpora is the plural of this.

Example: A collection of medical journals.

o **Token**- Each "entity" that is a part of whatever was split up based on rules. For examples, each word is a token when a sentence is "tokenized" into words. Each sentence can also be a token, if you tokenized the sentences out of a paragraph.

o **Lexicon** - Words and their meanings.

Example: English dictionary. Consider, however, that various fields will have different lexicons. For example: To a financial investor, the first meaning for thef word "Bull" is someone who is confident about the market, as compared to the common English lexicon, where the first meaning for the word "Bull" is an animal. As such, there is a special lexicon for financial investors, doctors, children, mechanics, and so on. A language model learns to predict the probability of a sequence of words. This ability to model the rules of a language as a probability gives great power for NLP related tasks. Language models are used in speech recognition, machine translation, part-of-speech tagging, parsing, Optical Character Recognition, handwriting recognition, information retrieval, and many other daily tasks.

• **Machine translation:** translating a sentence saying about height it would probably state that P(tall man)>P(large man) as the '*large*' might also refer to weight or general appearance thus, not as probable as '*tall*'

• **Spelling Correction:** Spell correcting sentence: "Put you name into form", so that P(name into **form**)>P(name into **from**)

• **Speech Recognition:** Call my nurse: P(Call my nurse.)≫P(coal miners), I have no idea. P(no idea.)≫P(No eye deer.)

• **Summarization, question answering, sentiment analysis** etc.

Tokenization is a common task in Natural Language Processing (NLP). It's a fundamental step in both traditional NLP methods like Count Vectorizer and Advanced Deep Learning-based architectures like Transformers. Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords. Hence, tokenization can be broadly classified into 3 types – word, character, and subword (n-gram characters) tokenization.

Word Tokenization is the most commonly used tokenization algorithm. It splits a piece of text into individual words based on a certain delimiter. Depending upon delimiters, different word-level tokens are formed. Pretrained Word Embeddings such as Word2Vec and GloVe comes under word tokenization.

Character Tokenization splits apiece of text into a set of characters. It overcomes the drawbacks we saw above about Word Tokenization. Character Tokenizers handles OOV words coherently by preserving the information of the word. It breaks down the OOV word into characters and represents the word in terms of these characters. It also limits the size of the vocabulary. Want to talk a guess on the size of the vocabulary? 26 since the vocabulary contains a unique set of characters. Subword Tokenization splits the piece of text into subwords (or n-gram characters). For example, words like lower can be segmented as low-er, smartest as smart-est, and so on.

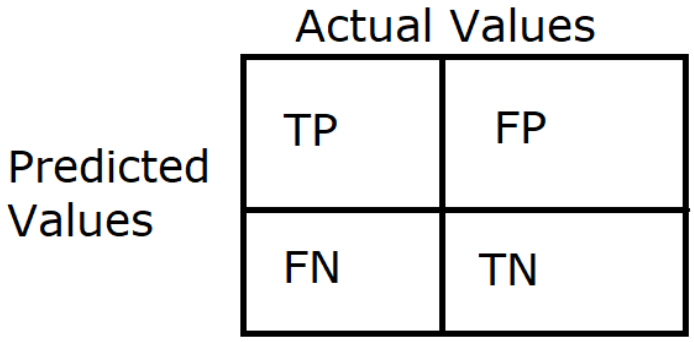
Word Embeddings are the texts converted into numbers and there may be different numerical representations of the same text. As we know that many Machine Learning algorithms and almost all Deep Learning Architectures are not capable of processing strings or plain text in their raw form. In a broad sense, they require numerical numbers as inputs to perform any sort of task, such as

classification, regression, clustering, etc. Also, from the huge amount of data that is present in the text format, it is imperative to extract some knowledge out of it and build any useful applications.

In short, we can say that to build any model in machine learning or deep learning, the final level data has to be in numerical form because models don't understand text or image data directly as humans do.

Tokenization is the process of dividing each sentence into words or smaller parts, which are known as tokens. After the completion of tokenization, we will extract all the unique words from the corpus. Here corpus represents the tokens we get from all the documents and used for the bag of words creation.

Confusion Matrix:



A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

- **true positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives (TN):** We predicted no, and they don't have the disease.
- **false positives (FP):** We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **false negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

Lemmatization & Stemming

When we speak or write, we tend to use inflected forms of a word (words in their different grammatical forms). To make these words easier for computers to understand, NLP uses lemmatization and stemming to transform them back to their root form.

The word as it appears in the dictionary – its root form – is called a lemma. For example, the terms "is, are, am, were, and been," are grouped under the lemma 'be.' So, if we apply this lemmatization to "African elephants have four nails on their front feet," the result will look something like this:

African elephants have four nails on their front feet = "African," "elephant," "have," "4", "nail," "on," "their," "foot"]

This example is useful to see how the lemmatization changes the sentence using its base form (e.g., the word "feet"" was changed to "foot").

When we refer to stemming, the root form of a word is called a stem. Stemming "trims" words, so word stems may not always be semantically correct.

For example, stemming the words “consult,” “consultant,” “consulting,” and “consultants” would result in the root form “consult.”

While lemmatization is dictionary-based and chooses the appropriate lemma based on context, stemming operates on single words without considering the context. For example, in the sentence:

“This is better”

The word “better” is transformed into the word “good” by a lemmatizer but is unchanged by stemming. Even though stemmers can lead to less-accurate results, they are easier to build and perform faster than lemmatizers. But lemmatizers are recommended if you're seeking more precise linguistic rules.

Stopword Removal

Removing stop words is an essential step in NLP text processing. It involves filtering out high-frequency words that add little or no semantic value to a sentence, for example, which, to, at, for, is, etc.

You can even customize lists of stopwords to include words that you want to ignore.

Let’s say you want to classify customer service tickets based on their topics. In this example: “Hello, I’m having trouble logging in with my new password”, it may be useful to remove stop words like “hello”, “I”, “am”, “with”, “my”, so you’re left with the words that help you understand the topic of the ticket: “trouble”, “logging in”, “new”, “password”.

Word Sense Disambiguation

Depending on their context, words can have different meanings. Take the word “book”, for example:

- You should read this **book**; it’s a great novel!
- You should **book** the flights as soon as possible.
- You should close the **books** by the end of the year.
- You should do everything by the **book** to avoid potential complications.

There are two main techniques that can be used for word sense disambiguation (WSD): knowledge-based (or dictionary approach) or supervised approach. The first one tries to infer meaning by observing the dictionary definitions of ambiguous terms within a text, while the latter is based on natural language processing algorithms that learn from training data.

Named Entity Recognition (NER)

Named entity recognition is one of the most popular tasks in semantic analysis and involves extracting entities from within a text. Entities can be names, places, organizations, email addresses, and more.

Relationship extraction, another sub-task of NLP, goes one step further and finds relationships between two nouns. For example, in the phrase “Susan lives in Los Angeles,” a person (Susan) is related to a place (Los Angeles) by the semantic category “lives in.”

Text Classification

Text classification is the process of understanding the meaning of unstructured text and organizing it into predefined categories (tags). One of the most popular text classification tasks is sentiment analysis, which aims to categorize unstructured data by sentiment.

3.1 Difference Between Classical NLP and Deep Learning based NLP

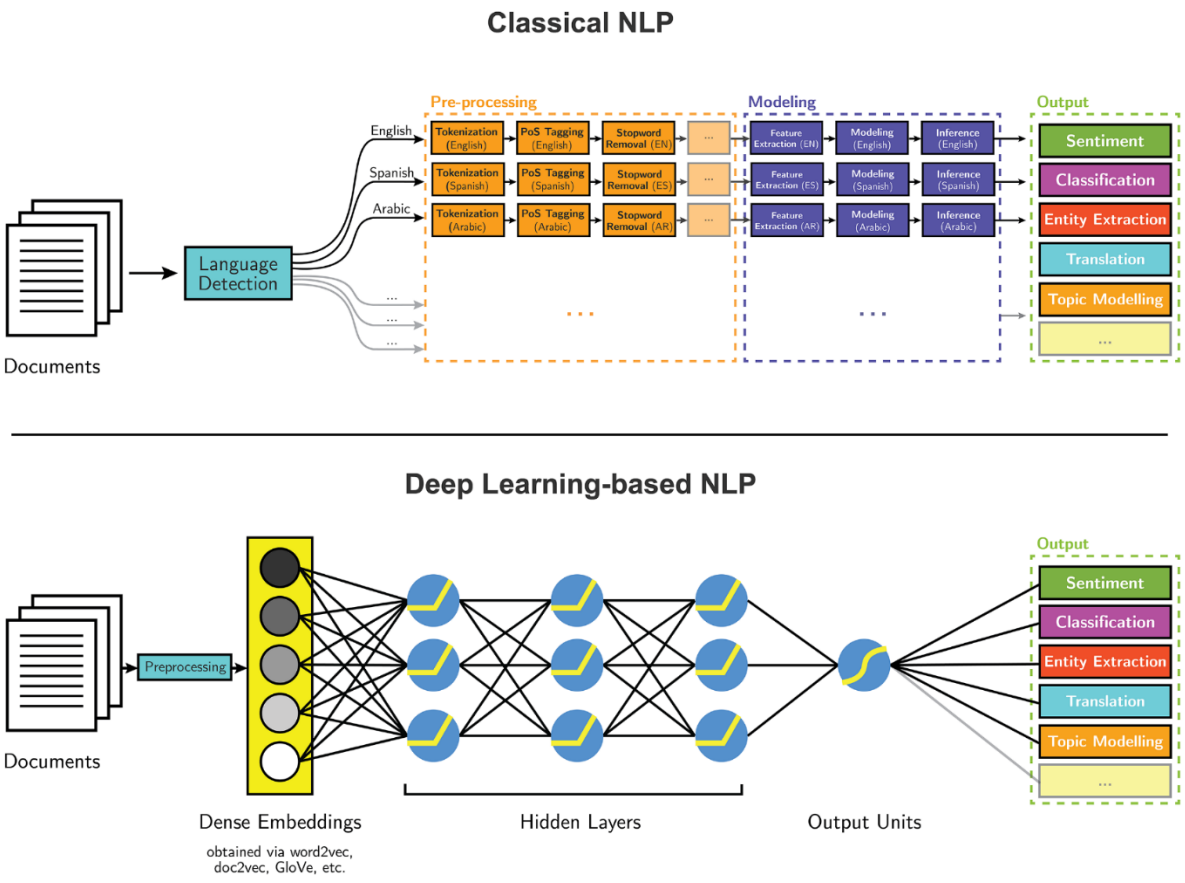


Fig 1: Comparison between NLP [4]

As we compare classical NLP and Deep learning NLP, classic NLP processes a small dataset form which it first detects the language and then performs preprocessing of the data. Tokenization, PoS tagging and stopwords removal occurs. Then the modeling occurs with feature extraction and inference. Then the output is displayed. A deep learning NLP processes a large dataset. It also preprocesses the data. Instead, it prepares a training model with deep neural network model with hidden layers to increase the accuracy.

3.2 NLG System Architecture

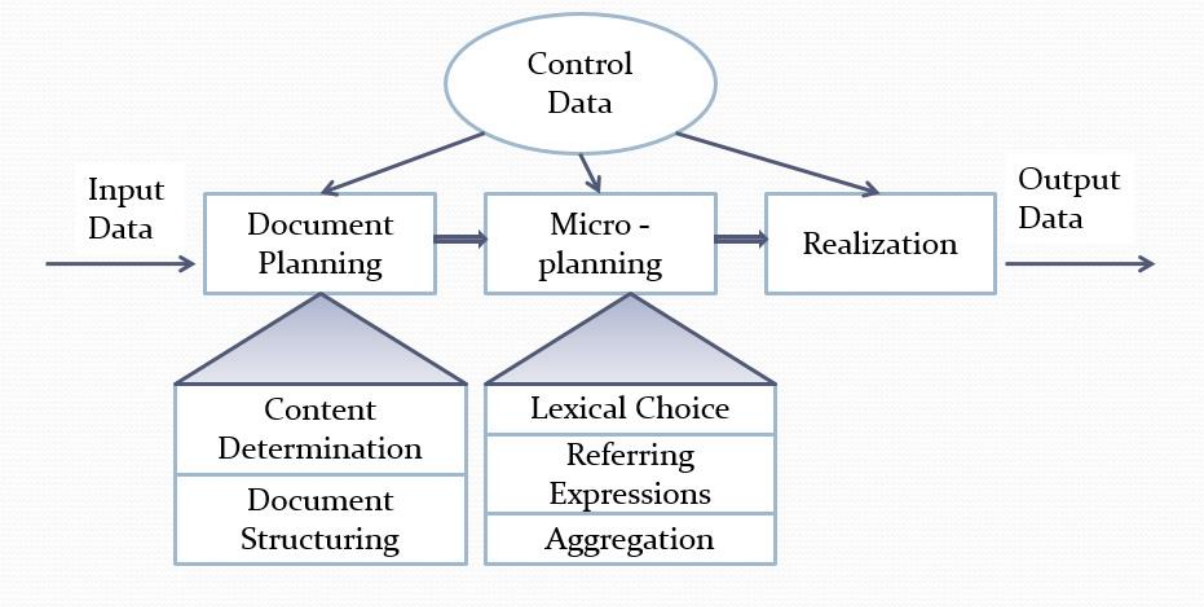


Fig 2: NLG system architecture.

Natural language generation (NLG) is the use of artificial intelligence (AI) programming to produce written or spoken narratives from a data set. NLG is related to human-to-machine and machine-to-human interaction, including computational linguistics, natural language processing (NLP) and natural language understanding (NLU).

NLG is a multi-stage process, with each step further refining the data being used to produce content with natural-sounding language. The six stages of NLG are as follows:

1. **Content analysis.** Data is filtered to determine what should be included in the content produced at the end of the process. This stage includes identifying the main topics in the source document and the relationships between them.

2. **Data understanding.** The data is interpreted, patterns are identified and it's put into context. Machine learning is often used at this stage.
3. **Document structuring.** A document plan is created and a narrative structure chosen based on the type of data being interpreted.
4. **Sentence aggregation.** Relevant sentences or parts of sentences are combined in ways that accurately summarize the topic.
5. **Grammatical structuring.** Grammatical rules are applied to generate natural-sounding text. The program deduces the syntactical structure of the sentence. It then uses this information to rewrite the sentence in a grammatically correct manner.
6. **Language presentation.** The final output is generated based on a template or format the user or programmer has selected.

Machine Translation

Machine translation (MT) involves the use of sophisticated computer programs to translate from one language into another language without the involvement of human translators. It is not just about substituting words in one language for those in another. MT involves the use of computational linguistic techniques to recognize and translate whole phrases while also considering the complex structure and figures of speech that are unique to each language. Machine translation (MT), process of translating one source language or text into another language, is one of the most important applications of NLP.

Chapter 4

Phase Plan Layout

- Language detection
    a) Preparing system for importing libraries like nltk, pandas, numpy, matplotlib, etc.
    b) Getting a dataset which is in the form of a CSV file.
    c) Working on dataset.
- Translation: focusing on translation of the text into languages where one output will be in English with roman text and other would be Hindi with Devanagari text
- Creating UI for user input and output: it will focus on creating an UI for our user from where they can interact here. The functions from phase 1 and phase 2 will called as user will interact using this layer and hence get the translation

Chapter 5

Project Design

Our software can be divided in three phases

Phase –I

It is the phase where we will train our model to detect the language which is entered by user it completely made by using python libraries which have been mentioned below with their full functioning and the tool we used in the process is Microsoft visual studio code as our IDLE.

Dataset – Our dataset is in csv form where the string of lines is assigned with their respective languages and it will be used to create the model on which will be used by our code to detect language.

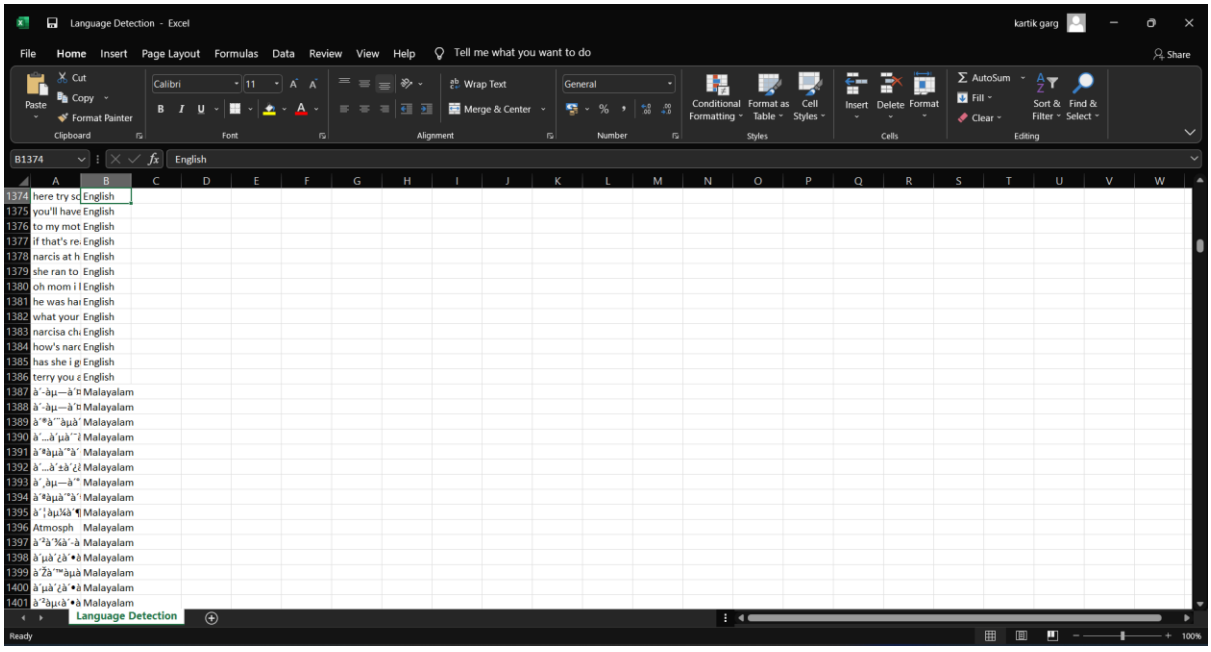


Fig 3: A sample of the dataset.

Now the libraries we have used are –

1) Pandas - this is used to manipulate data in our csv file. this library is required to read the csv file.

```
# Loading the dataset
data = pd.read_csv("C:\\Users\\gargk\\Downloads\\Language Detection.csv")
```

2) re – We will use the function of sub from it which will remove the symbols such as @#$% etc. so that the detector can work on only text part.

```
for text in X:
    # removing the symbols and numbers
    text = re.sub(r'[!@#$(),n"%^*?:;~`0-9]', ' ', text)
    text = re.sub(r'[[]]', ' ', text)
```

2)sklearn – this is the library which will help us to create the model by using its various functions.

a) labelEncoder - it is used to convert the variables into numerical which are used to fit in the model having respective language name

```
X = data["Text"]
y = data["Language"]
# converting categorical variables to numerical
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

b) Countervectoriztion – It is the part of library of sklearn and is used to create a bag of words which will be allotted to their respective language.

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X = cv.fit_transform(data_list).toarray()
```

c) Train_test_split - The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

d) multinomialNB – it is used to create the model

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x_train, y_train)
# prediction
y_pred = model.predict(x_test)
```

Now next two functions are used to test the model efficiency that are –

Accuracy score and confusion matrix

```
0.9796905222437138
[[118   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0  74   0   1   2   0   0   0   0   0   0   1   0   0   2   0   0]
 [  0   0 119   1   0   1   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0 271   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   2 203   0   0   0   0   0   0   0   0   1   0   0   0]
 [  0   0   1   0   1  90   0   0   0   0   0   1   0   0   1   0   0]
 [  0   0   0   2   0   0  72   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0  16   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0 121   0   0   0   0   2   0   0   0]
 [  0   0   0   2   0   0   0   0   0  65   0   0   0   0   0   0   0]
 [  0   0   0   2   0   0   0   0   0   0 106   0   0   0   0   0   0]
 [  0   0   0   1   0   0   0   0   1   0   0 140   0   0   0   0   0]
 [  0   0   0   3   0   0   0   0   0   0   0   0 139   0   0   0   0]
 [  0   0   0   2   0   0   0   0   0   0   0   2   0 163   0   0   0]
 [  0   0   0   1   1   0   0   0   0   0   0   0   0   0 130   0   0]
 [  0   0   0   3   0   0   0   0   0   0   0   0   0   0   0 109   0]
 [  0   0   0   1   0   1   0   0   0   0   0   0   0   1   0   0  90]]
```

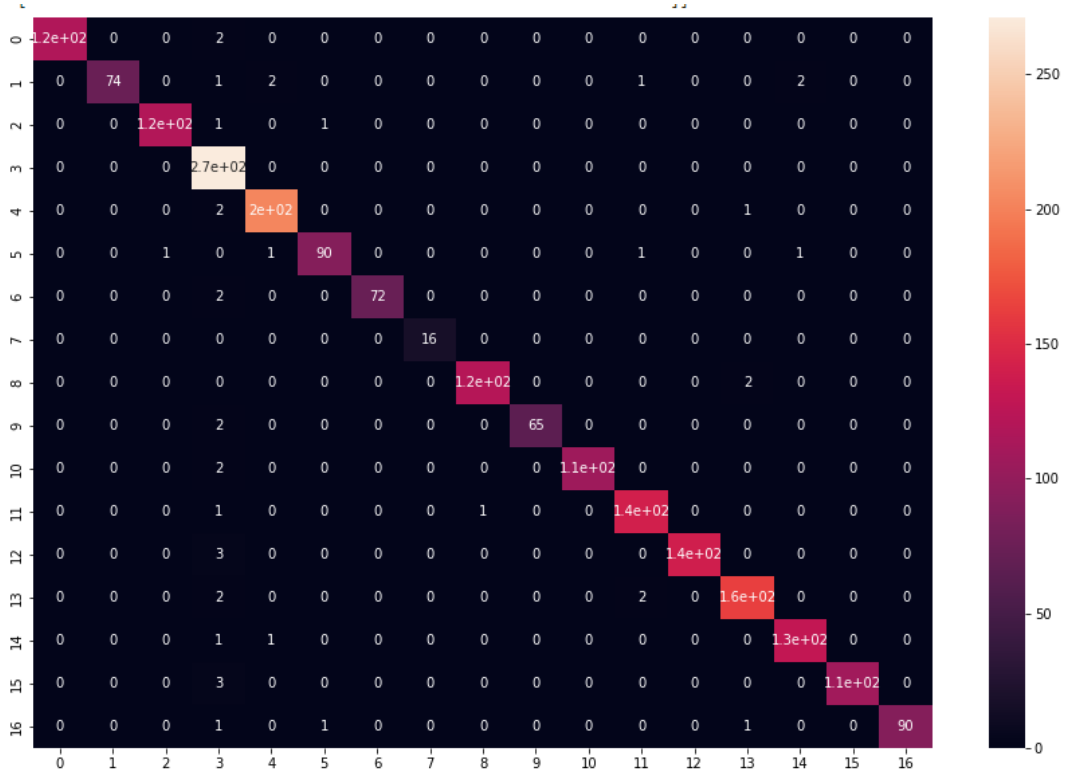


Fig 4: Confusion Matrix.

Phase -2

This phase will be used to translate the text from the detected language to English and Hindi

The library which will used here is –

1)Translate - it is the library where the text is translated from one language to another

```
from translate import Translator
Translator = Translator(from_lang= 'English',to_lang='Hindi')
Translation = Translator.translate("Hey how are you")
print(Translation)
```

Output –

```
roject\test.py
अरे आप कैसे हैं
```

The things which are need to be done here is the calling of previous phase file and using its function for prediction and later using that value in above method

```
import detector as dt
import translate as tl
print("Enter the Text")
text=input()
k=dt.predict(text)
print(k)
```

Phase -3

It will be started using tkinter library which will help us to create the UI of the software and here the file of both the phases will be called to make changes.

This UI will be the face of the web application.

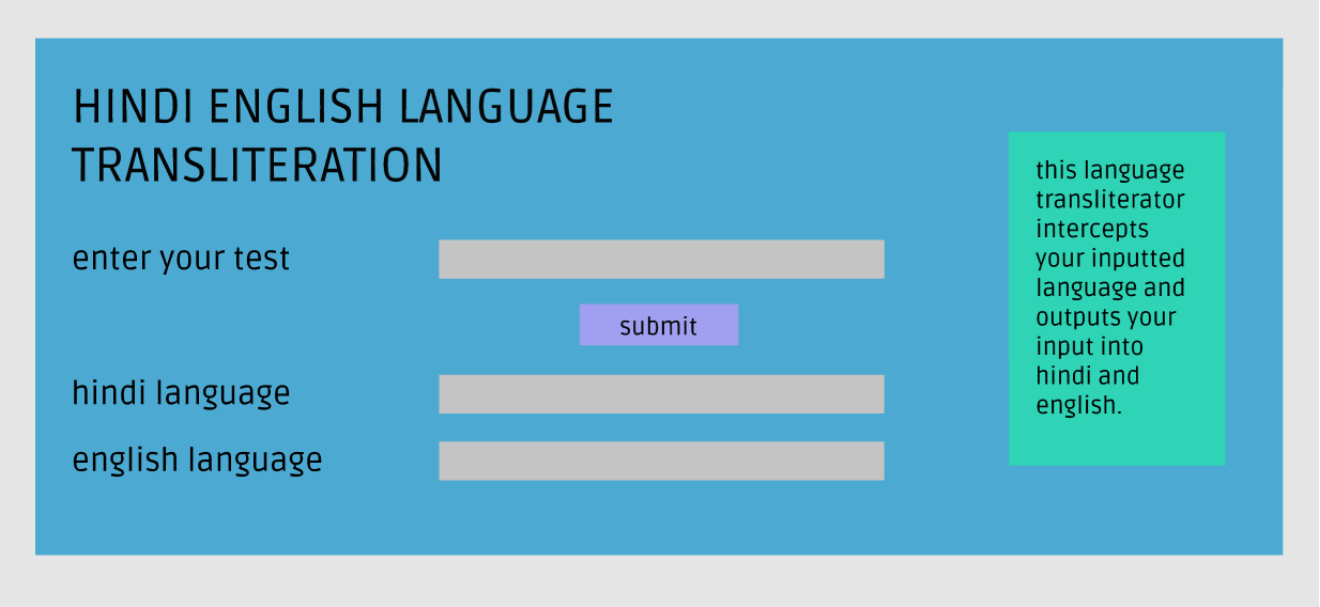


Fig 5: UI Design.

Chapter 6

SOURCE CODE

This is the code for prediction model which is created using NLP.

```
import pandas as pd
import numpy as np
import re
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import sklearn
warnings.simplefilter("ignore")
# Loading the dataset
data = pd.read_csv("C:\\Users\\gargk\\Downloads\\Language Detection.csv")
# value count for each language
data["Language"].value_counts()
# separating the independent and dependant features
X = data["Text"]
y = data["Language"]
# converting categorical variables to numerical
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
# creating a list for appending the preprocessed text
data_list = []
# iterating through all the text
```

```
for text in X:
    # removing the symbols and numbers
    text = re.sub(r'[!@#$(),n"%^*?:;~`0-9]', ' ', text)
    text = re.sub(r'[[]]', ' ', text)
    # converting the text to lower case
    text = text.lower()
    # appending to data_list
    data_list.append(text)
# creating bag of words using countvectorizer
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X = cv.fit_transform(data_list).toarray()
#train test splitting
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
#model creation and prediction
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x_train, y_train)
# prediction
y_pred = model.predict(x_test)
# model evaluation
from sklearn.metrics import accuracy_score, confusion_matrix
ac = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
# visualising the confusion matrix
```

```
plt.figure(figsize=(15,10))
sns.heatmap(cm, annot = True)
# function for predicting language
def predict(text):
    x = cv.transform([text]).toarray()
    lang = model.predict(x)
    lang = le.inverse_transform(lang)
    return(lang[0])
```

This is the language translator page

```
import detector as dt
import translate as tl
print("Enter the Text")
text=input()
k=dt.predict(text)
print(k)
def Translate():
    translator = tl.Translator(from_lang= k,to_lang='English')
    Translation = translator.translate("")
    print(Translation)
    return(Translation)

print(Translate)
```

Output:

Prediction model

```
:\Users\gargk\.vscode\extensions\ms-python.python-2021
roject\predictor.py'
Enter the Text
[illegible]
Malayalam
<function Translate at 0x000001C833964C10>
```

```
predict("എന്താ സുഖമല്ലെ")
```

```
[2]: 'Malayalam'
```

+ Code + Markdown

```
predict("உங்கள் நாள் எப்படி இருக்கிறது")
```

```
[3]: 'Tamil'
```

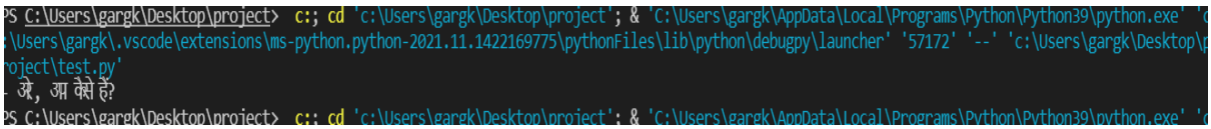
+ Code + Markdown

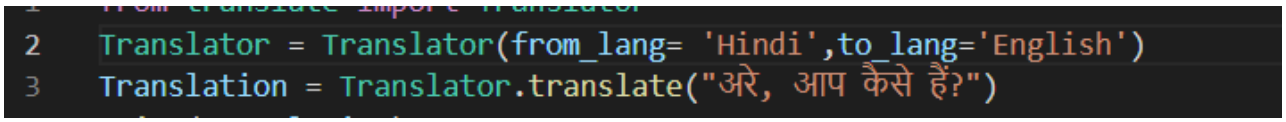
```
predict("Hej, hur mår du")
```

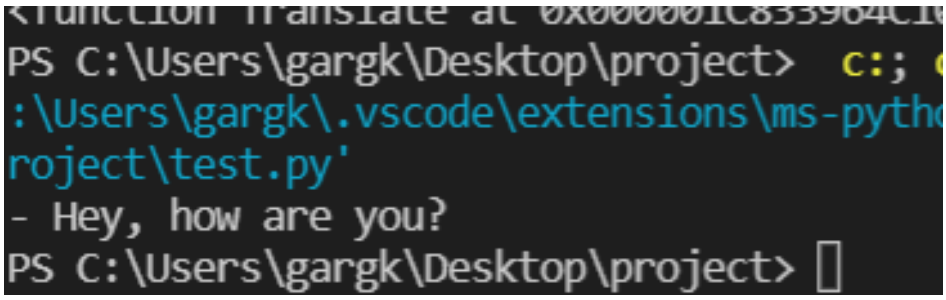
```
[4]: 'Sweedish'
```

+ Code + Markdown

Translation part







Databases

	A	B	C	D	E	F	G	H
2032	à¤•à¤² à¤•à¥‡ à¤²à¤¿à¤ à¤•à¥‹à¤ˆ à¤¨à¤¹à¥€à¤‚ à¤¥à¤¾	Hindi						
2033	à¤•à¥à¤¯à¥‹à¤‚ à¤†à¤ª à¤…à¤—à¤²à¥‡ à¤¶à¤¨à¤¿à¤µà¤¾à¤°	Hindi						
2034	à¤•à¥à¤¯à¥‹à¤‚ à¤†à¤ª à¤‡à¤¸ à¤¸à¥à¤•à¥‚à¤² à¤•à¥‡	Hindi						
2035	à¤¯à¤•à¥€à¤¨ à¤¹à¥ˆ, à¤†à¤ªà¤•à¥‡ à¤®à¤¨ à¤®à¥‡à¤‚, à	Hindi						
2036	à¤®à¥ˆà¤‚, à¤¸à¤à¥€ à¤¸à¥à¤¤à¤¹à¤¾à¤¸à¤¿à¤•à¤¤ à¤®à¥‡	Hindi						
2037	à¤…à¤—à¤° à¤®à¥ˆà¤‚, à¤•à¤² à¤•à¥‡ à¤¦à¤¿à¤¨ à¤¨à¤¿à¤¯ à¤¨à¤¿	Hindi						
2038	à¤¹à¤¾à¤‚, à¤•à¥fà¤ªà¤¯à¤¾ à¤¸à¥à¤¨à¤¿à¤¶à¥à¤šà¤¿à¤¤	Hindi						
2039	à¤•à¥à¤¯à¤¾ à¤†à¤ª à¤®à¥à¤¤à¥‡ à¤•à¥à¤› à¤ªà¥ˆà¤¸à¥	Hindi						
2040	à¤®à¥ˆà¤‚, à¤¸à¥‹à¤š à¤°à¤¹à¤¾ à¤¥à¤¾ à¤•à¤¿ à¤•à¥à¤	Hindi						
2041	à¤•à¥ˆà¤¸à¥‡ à¤•à¥à¤› à¤†à¤‡à¤•à¥à¤°à¥€à¤® à¤•à¥	Hindi						
2042	à¤•à¥à¤¯à¤¾ à¤†à¤ª à¤®à¥à¤¤à¥‡ à¤¤à¤• à¤¸à¤µà¤¾à¤°à	Hindi						
2043	à¤œà¥€ à¤¬à¥‹à¤²à¤¿à¤¯à¥‡ à¤¯à¤¹ à¤¬à¤¹à¥à¤¤ à¤.	Hindi						
2044	à¤à¤• à¤¸à¥‡à¤•à¤‚à¤¡ à¤ªà¤° à¤²à¤—à¤•à¤¾à¤“ à¤–	Hindi						
2045	à®µà®¿à®•à¯à®•à®¿à®ªà¯à®€à®Ÿà®¿à®¯à¯ (Wikiped	Tamil						
2046	à®¤à®®à®¿à®´à¯ à®µà®¿à®•à¯à®•à®¿à®ªà¯à®€à®Ÿà®¿	Tamil						
2047	à®ªà¯†à®°à¯à®ªà®¾à®²à¯à®•à¯ à®‡à®¤à¯	Tamil						
2048	[6] à®®à¯‡à®²à¯à®¨à¯ à®‡à®¤à¯ à®•à®¿à®Ÿà¯à®¤à¯à®©	Tamil						
2049	[7][8] à®šà®©à®µà®°à¿ 2021 à®µà®°à¯ˆà®¯à¯, à®	Tamil						
2050	à®‡à®¤à¯ à®‡à®£à¯ˆà®¯à¯à®¤à¯à®³à®¿à®¤à¯ à	Tamil						
2051	[9][10][11][12] à®®à¯‡à®²à¯à®¨, à®‡à®¤à¯ à®…à®²à¯†à	Tamil						

	A	B	C	D	E	F	G	H	I
1	naar mijn moeder laat me naar huis gaan melissa en terre	Dutch							
2	als dat echt is wat je wilt, dan neem ik je mee naar huis.	Dutch							
3	Terrence nam het huilen en ongelukkig op.	Dutch							
4	narcis thuis in de koets begon ze diep na te denken over d	Dutch							
5	ze rende om haar moeder te omhelzen die in het huis zat,	Dutch							
6	oh mam ik hou zo veel van je en je had gelijk uiterlijk en rij	Dutch							
7	hij was knap hij gaf helemaal niet om me maar mijn liefste	Dutch							
8	Ik weet dat het me zo spijt van de manier waarop ik me ge	Dutch							
9	narcisa veranderde haar manier van doen ze worstelde ee	Dutch							
0	hoe' Marian vertelde hun nu allebei alles wat er was gebe	Dutch							
1	heeft ze, denk ik dat ze nu geen goudbrood meer zou wille	Dutch							
2	Terry je lijkt eigenlijk een beetje op die engel, maar wat zie	Dutch							
3	Nature es una de las mÃ¡s prestigiosas revistas cientÃ-fica	Spanish							
4	Su primer nÃºmero fue publicado el 4 de noviembre de 18	Spanish							
5	[2]â€‹ Con una periodicidad semanal, la revista es publica	Spanish							
6	Nature tiene oficinas en Londres, Nueva York, San Francis	Spanish							
7	La editorial publica tambiÃ©n varias revistas especializad	Spanish							
8	La revista es leÃ-da por cientÃ-ficos e investigadores de t	Spanish							
9	Sin embargo tambiÃ©n incluye editoriales y noticias cient	Spanish							
0	SegÃºn los Journal Citation Reports, la revista tiene un fac	Spanish							

4	Î¼ÎµÎ´ÏŽ Î´Î¿ÎºÎ¹Î¼Î¬ÏƒÏ„Îµ Î¼ÎµÏÎ¹ÎºÎÏ‚ Ï‡ÏÏ…ÏƒÎÏ‚ Ï€Î±Ï„Î¬Ï„ÎµÏ‚, Ï€	Greek			
5	Î¸Î± Ï€ÏÎÏ€ÎµÎ¹ Î½Î± Ï†Î¬Ï„Îµ ÏŒÎ»Î± Î±Ï…Ï„Î¬, Î½ÏŒÎ¼Î¹Î¶Î± Ï	Greek			
6	ÏƒÏ„Î· Î¼Î·Ï„ÎÏÎ± Î¼Î¿Ï… Î¬ÏÎ·ÏƒÎ Î¼Îµ Î½Î± Ï€Î¬ÎµÎ¹ ÏƒÏ€Î¯Ï„Î¹,	Greek			
7	Î±Î½ Î±Ï…Ï„ÏŒ ÎµÎ¯Î½Î±Î¹ Ï€ÏÎ±Î³Î¼Î±Ï„Î¹ÎºÎ¬ Î±Ï…Ï„ÏŒ Ï€Î¿Ï…	Greek			
8	narcis ÏƒÏ„Î¿ ÏƒÏ€Î¯Ï„Î¹ Ï„Î·Ï‚ Î¬Î¼Î±Î¾Î±Ï‚ Î¬ÏÏ‡Î¹ÏƒÎµ Î½Î± ÏƒÎºÎ	Greek			
9	ÎÏ„ÏÎµÎ¾Îµ Î½Î± Î±Î³ÎºÎ±Î»Î¹Î¬ÏƒÎµÎ¹ Ï„Î· Î¼Î·Ï„ÎÏÎ± Ï„Î·Ï‚ Ï€Î¿Ï..	Greek			
0	Ï‰ Î¼Î±Î¼Î¬ Ïƒ 'Î±Î³Î±Ï€ÏŽ Ï„ÏŒÏƒÎ¿ Ï€Î¿Î»Ï ÎºÎ±Î¹ ÎµÎ¯ÏƒÎ±Î¹ Ïƒ	Greek			
1	Î®Ï„Î±Î½ ÏŒÎ¼Î¿ÏÏ†Î¿Ï„ Î´ÎµÎ½ Î¼Îµ Î½Î¿Î¹Î¬Î¶ÎµÎ¹ ÎºÎ±Î¸ÏŒÎ»Î	Greek			
2	Ï„Î¹ Î®ÏÎ¸Îµ Ï„Î¿ Î»Î¬Î¸Î¿Ï‚ ÏƒÎ¿Ï…, Î±Ï‚ ÏƒÎ¿Ï… Î´ÏŽÏƒÎ¿Ï…Î¼Îµ ÎºÎ¬	Greek			
3	Î— Î½Î±ÏÎºÎ¯ÏƒÏƒÎ± Î¬Î»Î»Î±Î¾Îµ Ï„Î¿Ï…Ï‚ Ï„ÏÏŒÏ€Î¿Ï…Ï‚ Ï„Î·Ï‚ Ï€	Greek			
4	Ï€Ï‰Ï‚' ÎŸ Î½Î±ÏÎºÎ¹ÏƒÏƒÎ¹ÏƒÎ¼ÏŒÏ„ Ï„ÏŽÏÎ± Î· Î¼Î±ÏÎ¯Î±, Ï„Î¿Ï…Ï‚	Greek			
5	ÎœÎ®Ï€Ï‰Ï‚ Ï…Ï€Î¿Î¸ÎÏ„Ï‰ ÏŒÏ„Î¹ Î´ÎµÎ½ Î¸Î± Î®Î¸ÎµÎ»Îµ Î¬Î»Î	Greek			
6	Terry, ÏƒÏ„Î·Î½ Ï€ÏÎ±Î³Î¼Î±Ï„Î¹ÎºÏŒÏ„Î·Ï„Î± Î¼Î¿Î¹Î¬Î¶ÎµÎ¹Ï‚, Î»Î¯	Greek			
7	Nature (Ð²Â Ð¿ÐµÑ€ÐµÐ²Ð¾Ð´Ðµ ÑÂ Ð°Ð½Ð³Ð».	Russian			
8	â€”â€‰Â«ÐŸÑ€Ð¸Ñ€Ð¾Ð´Ð°Â»)Â â€” Ð¾Ð´Ð¸Ð½ Ð¸Ð· Ñ	Russian			
9	ÐŸÑƒÐ±Ð»Ð¸ÐºÑƒÐµÑ‚ Ð¸ÑÑÐ»ÐµÐ´Ð¾Ð²Ð°Ð½Ð¸Ñ, Ð¿Ð	Russian			
0	ÐŸÐµÑ€Ð²Ñ‹Ð¹ Ð½Ð¾Ð¼ÐµÑ€ Ð¶ÑƒÑ€Ð½Ð°Ð»Ð° Nature	Russian			
1	Nature Ñ€ÐµÐ´Ð°ÐºÑ‚Ð¸Ñ€ÑƒÐµÑ‚ÑÑ Ð² Ð'ÐµÐ»Ð¸ÐºÐ¾	Russian			
2	Ð£ Ð¶ÑƒÑ€Ð½Ð°Ð»Ð° ÐµÑÑ‚ÑŒ Ñ‚Ð°ÐºÐ¶Ðµ Ð¾Ñ„Ð¸Ñ	Russian			
3	Ð–ÑƒÑ€Ð½Ð°Ð» Ð¾Ñ€Ð¸ÐµÐ½Ñ‚Ð¸Ñ€Ð¾Ð²Ð°Ð½ Ð½Ð° Ð	Russian			
4	Ð' ÐºÐ¾Ð»Ð¾Ð½ÐºÐµ Ñ€ÐµÐ´Ð°ÐºÑ‚Ð¾Ñ€Ð° Ð¸ Ñ€Ð°Ð	Russian			
5	News Articles) ÑÐ¾Ð¾Ð±Ñ‰Ð°ÐµÑ‚ÑÑ Ð¾ ÑÐ¾Ð±Ñ‹Ñ‚	Russian			
6	ÐžÑÑ‚Ð°Ð»ÑŒÐ½Ð°Ñ Ñ‡Ð°ÑÑ‚ÑŒ Ð¶ÑƒÑ€Ð½Ð°Ð»Ð° Ñ	Russian			
7	Ð¡ 2005 Ð³Ð¾Ð´Ð° Ð¶ÑƒÑ€Ð½Ð°Ð» Ð¿ÑƒÐ±Ð»Ð¸ÐºÑƒÐµ	Russian			

3	de rubis as saladas de esmeraldas e o pÃ£o ali Ã© feito d	Portugeese
9	outra coisa, mas eu quero pÃ£o normal, pÃ£o que eu pos	Portugeese
0	mas estava com o coraÃ§Ã£o partido por tudo que era in	Portugeese
1	experimente aqui algumas batatas douradas, elas sÃ£o m	Portugeese
2	vocÃª vai ter que comer tudo isso Eu pensei que vocÃª go	Portugeese
3	para minha mÃ£e deixe-me ir para casa melissa e terrenc	Portugeese
4	se isso Ã© mesmo o que vocÃª quer, entÃ£o eu vou te lev	Portugeese
5	narcis em casa, na carruagem, ela comeÃ§ou a pensar pr	Portugeese
6	ela correu para abraÃ§ar a mÃ£e que estava sentada em	Portugeese
7	oh mÃ£e eu te amo tanto e vocÃª estava certa aparÃªnci	Portugeese
8	qual foi o seu erro, vamos alimentar vocÃª com algo legal	Portugeese
9	narcisa mudou seus hÃ¡bitos, ela lutou no inÃ-cio, mas ler	Portugeese
0	Como' s narcisismo agora marian contou a ambos tudo o	Portugeese
1	tem ela, eu acho que ela nÃ£o iria querer mais pÃ£o de o	Portugeese
2	Terry, vocÃª realmente se parece um pouco com aquele a	Portugeese
3	Si vous disposez d'ouvrages ou d'articles de rÃ©fÃ©rence	French
4	Comment ajouter mes sourcesÂ ?	French
5	Cette page ou section est en train d'Ãªtre traduite en fran	French
6	Vous pouvez aider au dÃ©veloppement de WikipÃ©dia e	French
7	Le mot nature est un terme polysÃ©mique (câ€™est-Ã -d	French
8	Au sens commun, la nature peut regrouperÂ : Face au con	French
9	Les principes de l'Ã©thique environnementale, de nouvell	French
0	Si l'Ã©tymologie du terme Â«Â natureÂ Â» est relativeme	French
1	Le mot nature est attestÃ© en franÃ§ais depuis 1119[2].	French
2	Il vient du latin natura, qui dÃ©signait Â«Â le cours des ch	French
3	Le terme vient lui-mÃªme du verbe nascor (Â«Â naÃ®treÂ	French
4	Si ce terme signifie essentiellement le Â«Â caractÃ¨re innÃ	French

Language Detection

Chapter 7

Conclusion

The project will be able to translate the user input with mixed languages and produce the output as the user wants it. The use of Artificial intelligence and its components are used in full manner to get maximum possible accuracy. This project is a next step towards easy machine translation. Language detection, translation and creating UI for the users is a planned phase which is executed in a sequential manner so that the internal errors are kept minimum and the software gets to its full extent.

References

1. M. M. Phadke and S. R. Devane, "Multilingual machine translation : An analytical study," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 881-884, doi: 10.1109/ICCONS.2017.8250590.
2. Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean, “Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation.” Transactions of the Association for Computational Linguistics 2017; 5 339–351.
3. Middi Venkata Sai Rishita, Middi Appala Raju, and Tanvir Ahmed Harris, “Machine translation using natural language processing,” International Joint Conference on Metallurgical and Materials Engineering, 2018, MATEC Web Conf. Volume 277, 2019, pp. 1-6.
4. https://s3.amazonaws.com/aylien-main/misc/blog/images/nlp-language-dependence-small.png