***K.T.S.P Mandal’s***

***K.M.C College Khopoli***

***DEPARTMENT OF COMPUTER SCIENCE***

***KHOPOLI–410203***

*A Project Report*

*On*

***Language Detection And Translation***

*Submitted To*

***University of Mumbai***

*By*

***Miss Neha Narendra Ghonge***

*Under Guidance Of*

***Prof. Dhanashree Pawar***

*2021-2022*

***K.T.S.P MANDAL’S***

***KMC COLLEGE KHOPOLI***

***DEPARTMENT OF COMPUTER SCIENCE***

***CERTIFICATE***

*This is to certify that* ***Neha Narendra Ghonge***  *has successfully completed the project on the topic of*

*“****Language Detection And Translation****” in Sem-II.*

*During the academic year 2021-2022 as per the guidelines issued by* ***University of Mumbai****.*

***Teacher’s HOD’s Examiner’s***

***Signature Signature Signature***

***Date: Date:***

***Acknowledgment***

*In the accomplishment of this project successfully, many people have best owned upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with this project.*

*Primarily, I would thank god for being able to complete this project with success. Then I would like to thank my principal* ***Prof****,****Dr.Pratap Patil*** *and my project teacher* ***Prof. Dhanashree Pawar*** *whose valuable guidance has been the ones that helped me patch this project and make it full proof success. Her suggestions and her instructions have served as the major contributor towards the completion of the project. I am also thankful to my head of department* ***Prof. Dhanashree Pawar*** *who encourage me and gave me moral support during my project.*

*Technologies Used*

***Software requirement:***

*Software requirements for this system are as listed follows:*

* Frontend : Python
* Backend : Python
* Operating System : Windows

***Hardware requirement:***

*Minimum Hardware requirements for these system are listed below:*

* C.P.U:- RMD Ryzen.
* R.A.M:- 8 Giga Bytes.
* Hard Disk:- 40 Giga Bytes.
* Type Of System : Single User

*Description Of Project*

**Language Detection & Translation:-**

The language detection service is used to identify the language of business texts, such as emails and chats. The service identifies the language of a text and the parts of that text where the language changes, down to the word level. Using the language detection service, Surveillance Insights can highlight and annotate the languages that are used in a text and help to identify potential suspicious activity.

For Language Detection and Translation in Python We require modules such as

From “googletrans” we import “Translator and Constants “ and For Structured printing we use “pprint” i.e preety print

For installing this Modules:-

* pip install googletrans
* pip install pprint

Also For reading the “dataset.csv” we used “pandas” and in the dataset we have “Text Languages” .We used “Count Vectorizer” and “MultinomialNB” from sklearn and also used “numpy”. For getting input from user and performing operations on it and translating to any language.

Modules installation required for this are:-

* pip install numpy
* pip install pandas
* pip install sklearn

Translator () function is used from googletrans for Translation & Detection.

* translator.detect() is used for detection
* translator.translate() is used for Translation.

The pprint(constants.LANGUAGES) is used for printing all the Languages supported by googletrans.

*Coding*

import pandas as pd

import numpy as np

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

from googletrans import Translator, constants

from pprint import pprint

data = pd.read\_csv("https://raw.githubusercontent.com//amankharwal//Website-data//master//dataset.csv")

print(data.head())

print(data.isnull().sum())

print(data["language"].value\_counts())

x = np.array(data["Text"])

y = np.array(data["language"])

cv = CountVectorizer()

X = cv.fit\_transform(x)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=42)

model = MultinomialNB()

model.fit(X\_train,y\_train)

model.score(X\_test,y\_test)

print(model.score(X\_test,y\_test))

user = input("Enter a Text: ")

data = cv.transform([user]).toarray()

output = model.predict(data)

print(output)

#For translating the language

translator = Translator()

print("\nTranslating to arabic")

translation = translator.translate(user,dest="ar")

print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")

print("\nTranslating to swahili")

translation = translator.translate(user,dest="sw")

print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")

print("\nTranslating to chinese (traditional)")

translation = translator.translate(user,dest="zh-tw")

print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")

print("\nTranslating to marathi")

translation = translator.translate(user,dest="mr")

print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")

print("\nTranslating to korean")

translation = translator.translate(user,dest="ko")

print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")

#Detecting a language by using google trans

# detect a language

detection = translator.detect(user)

print("Language code:", detection.lang)

print("Confidence:", detection.confidence)

# print all available languages

print("\n Language supported by google trans")

print("Total supported languages:", len(constants.LANGUAGES))

print("Languages:")

pprint(constants.LANGUAGES)

*Output*

Text language

0 klement gottwaldi surnukeha palsameeriti ning ... Estonian

1 sebes joseph pereira thomas på eng the jesuit... Swedish

2 ถนนเจริญกรุง อักษรโรมัน thanon charoen krung เ... Thai

3 விசாகப்பட்டினம் தமிழ்ச்சங்கத்தை இந்துப் பத்திர... Tamil

4 de spons behoort tot het geslacht haliclona en... Dutch

Text 0

language 0

dtype: int64

Arabic 1000

Korean 1000

Estonian 1000

English 1000

French 1000

Chinese 1000

Russian 1000

Urdu 1000

Persian 1000

Indonesian 1000

Tamil 1000

Romanian 1000

Turkish 1000

Spanish 1000

Japanese 1000

Pushto 1000

Latin 1000

Swedish 1000

Thai 1000

Portugese 1000

Hindi 1000

Dutch 1000

Name: language, dtype: int64

0.953168044077135

Enter a Text: 멈추지 말고 계속 해나가기만 한다면 늦어도 상관없다

['Korean']

Translating to arabic

멈추지 말고 계속 해나가기만 한다면 늦어도 상관없다 (ko) --> لا يهم إذا واصلت القيام بذلك ، لا تتوقف (ar)

Translating to swahili

멈추지 말고 계속 해나가기만 한다면 늦어도 상관없다 (ko) --> Haijalishi ikiwa utaendelea kuifanya, usiache (sw)

Translating to chinese (traditional)

멈추지 말고 계속 해나가기만 한다면 늦어도 상관없다 (ko) --> 無論您繼續這樣做都沒關係，不要停止 (zh-tw)

Translating to marathi

멈추지 말고 계속 해나가기만 한다면 늦어도 상관없다 (ko) --> आपण हे करत राहिल्यास काही फरक पडत नाही, थांबवू नका (mr)

Translating to korean

멈추지 말고 계속 해나가기만 한다면 늦어도 상관없다 (ko) --> 멈추지 말고 계속 해나가기만 한다면 늦어도 상관없다 (ko)

Language code: ko

Confidence: None

Language supported by google trans

Total supported languages: 107

Languages:

{'af': 'afrikaans',

'am': 'amharic',

'ar': 'arabic',

'az': 'azerbaijani',

'be': 'belarusian',

'bg': 'bulgarian',

'bn': 'bengali',

'bs': 'bosnian',

'ca': 'catalan',

'ceb': 'cebuano',

'co': 'corsican',

'cs': 'czech',

'cy': 'welsh',

'da': 'danish',

'de': 'german',

'el': 'greek',

'en': 'english',

'eo': 'esperanto',

'es': 'spanish',

'et': 'estonian',

'eu': 'basque',

'fa': 'persian',

'fi': 'finnish',

'fr': 'french',

'fy': 'frisian',

'ga': 'irish',

'gd': 'scots gaelic',

'gl': 'galician',

'gu': 'gujarati',

'ha': 'hausa',

'haw': 'hawaiian',

'he': 'hebrew',

'hi': 'hindi',

'hmn': 'hmong',

'hr': 'croatian',

'ht': 'haitian creole',

'hu': 'hungarian',

'hy': 'armenian',

'id': 'indonesian',

'ig': 'igbo',

'is': 'icelandic',

'it': 'italian',

'iw': 'hebrew',

'ja': 'japanese',

'jw': 'javanese',

'ka': 'georgian',

'kk': 'kazakh',

'km': 'khmer',

'kn': 'kannada',

'ko': 'korean',

'ku': 'kurdish (kurmanji)',

'ky': 'kyrgyz',

'la': 'latin',

'lb': 'luxembourgish',

'lo': 'lao',

'lt': 'lithuanian',

'lv': 'latvian',

'mg': 'malagasy',

'mi': 'maori',

'mk': 'macedonian',

'ml': 'malayalam',

'mn': 'mongolian',

'mr': 'marathi',

'ms': 'malay',

'mt': 'maltese',

'my': 'myanmar (burmese)',

'ne': 'nepali',

'nl': 'dutch',

'no': 'norwegian',

'ny': 'chichewa',

'or': 'odia',

'pa': 'punjabi',

'pl': 'polish',

'ps': 'pashto',

'pt': 'portuguese',

'ro': 'romanian',

'ru': 'russian',

'sd': 'sindhi',

'si': 'sinhala',

'sk': 'slovak',

'sl': 'slovenian',

'sm': 'samoan',

'sn': 'shona',

'so': 'somali',

'sq': 'albanian',

'sr': 'serbian',

'st': 'sesotho',

'su': 'sundanese',

'sv': 'swedish',

'sw': 'swahili',

'ta': 'tamil',

'te': 'telugu',

'tg': 'tajik',

'th': 'thai',

'tl': 'filipino',

'tr': 'turkish',

'ug': 'uyghur',

'uk': 'ukrainian',

'ur': 'urdu',

'uz': 'uzbek',

'vi': 'vietnamese',

'xh': 'xhosa',

'yi': 'yiddish',

'yo': 'yoruba',

'zh-cn': 'chinese (simplified)',

'zh-tw': 'chinese (traditional)',

'zu': 'zulu'}

*Conclusion And Future Use*

**Conclusion:-**

**Language identification** can be an important step in a Natural Language Processing (NLP) problem. It involves trying to predict the natural language of a piece of text. It is important to know the language of text before other. By **Language Detection & Translation** we can:-

* Find out bias in text data based on the languages.
* You can classify the article based on the different languages.
* Language is generally associated with the region. This method helps you to classify the article based on languages.
* You can use this method in the language translation model.
* You can use it in data cleaning and data manipulation processes.

We should consider language detection as one of the data cleaning processes, for textual data. Internet text data are not always present in the English language.

**Future use** :-

**Automatic language detection** is a technology for determining the language of texts. It is a great use case for artificial intelligence. The detection model can analyze the given text and identify the language. In this way, it can help you to sort information quickly and apply additional layers of language-specific workflows effectively.

*Refrences*

* Multilingual Natural Language Processing Applications: From Theory to Practice by Bikel (Author).
* Handbook of Computational Linguistics and Natural Language

Processing, Martin Whitehead, Clanrye International, 2020.

* <https://towardsdatascience.com/develop-a-text-language-detector-and-translator-system-in-python-6537d6244b10>.
* <https://zetcode.com/python/googletrans/>.