COMSATS UNIVERSITY ISLAMABAD,

LAHORE CAMPUS

Instructor: Dr. Allah Bux Sargana

Course Title: CSC354- Machine Learning

Project Title: Language detection Model **Group: IV**

Name	Reg# #	Email	Phone #	CGPA	Credit (for office use)				
					5	15	15	15	10
Hasnain Ali	SP20-BCS-122	SP20-BCS-122@gmail.com	+92 313 5085477	3.24					
Azeem Manzoor	SP20-BCS-046	SP20-BCS-046@gmail.com	+92 334 7207291	2.62					
Adnan Safdar	SP20-BCS-116	SP20-BCS-116@gmail.com	+92 301 4717085	2.30					
Ayesha Tariq	SP20-BCS-020	ayesha981606@gmail.com	+92 309 5424856	2.73					

Abstract:

Identifying the language of a text is a key step of a reading system. We often see some electronic text where text is unconstrained and mix variable writing types. Language identification system is the task of automatically detecting the language present in any electronic form for example, in a document, a text message, a social media post etc. . In this work, we address the problem of detecting electronic text that contain text from more than one language (multilingual documents). We introduce a method that is able to detect the languages present, and estimate their relative proportions. We demonstrate the effectiveness of our method over synthetic data, as well as real-world multilingual documents collected from the web. This is a solution for many artificial intelligence applications and computational linguists. It helps in tracking and identifying multilingual documents too.

Introduction:

Now a days, a lot of textual information is stored in electronic form such as legal documents, text messages, web documents etc. Sometimes, a text is written in another language which everyone can't detect that in which language a text is written. So, language detection is becoming important increasingly. This system requires a supervised learning technique that classifies every piece of text by assigning one or more class labels from a fixed or predefined class. It treats the language like it's just a bag full of words and each message is a random handful of them**.** Large documents have a lot of words that are generally characterized by very high dimensionality feature space with thousands of features. Hence, the learning algorithm requires to tackle high dimensional problems, both in terms of classification performance and computational speed.

Automatic language detection is the first step towards achieving a variety of tasks like detecting the source language for machine translation, improving the search relevancy by personalizing the search results according to the query language, providing uniform search box for a multilingual dictionary, tagging data stream from social media with appropriate language etc. While classifying languages belonging to disjoint groups is not hard, disambiguation of languages originating from the same source and dialects still pose a considerable challenge in the area of natural language processing.

Methodology:

1. Dataset description:

The data for this project work was obtained from: "https://raw.githubusercontent.com/amankharwal/website-data/master/dataset.csv". A set of 1000 instances per language was provided for 22 different world languages. The languages are grouped as shown in Table. Each entry in the dataset is a full sentence written in one of the languages and tagged with the language group and country of origin. A similar set of mixed language instance was also provided to add noise to the data.

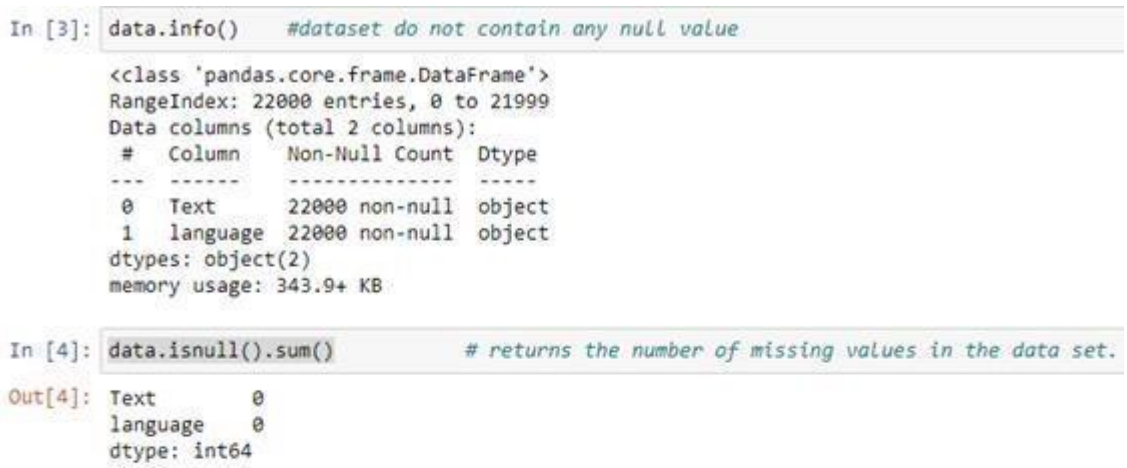
Sr#	Language
1.	Estonian
2.	Swedish

3.	English
4.	Russian
5.	Romanian
6.	Persian
7.	Pushto
8.	Spanish
9.	Hindi
10.	Korean
11.	Chinese
12.	French
13.	Portuguese
14.	Indonesian
15.	Urdu
16.	Latin
17.	Turkish
18.	Japanese
19.	Dutch
20.	Tamil
21.	Thai
22	Arabic

2. Experiments and Methods:

Language Identification is a supervised learning task, particularly a plain single label multi class classification. Given some historical or training data in which for each text *t* there exist a label *l*, the language in which the text is written, the goal is to learn a model such that the given some previously unseen text, it can identify, as accurately as possible, in which this text is written.

First, we separated our independent and dependent variables from the data set. Then we preprocessed our input and output variables. Our data set do not contain any null value as

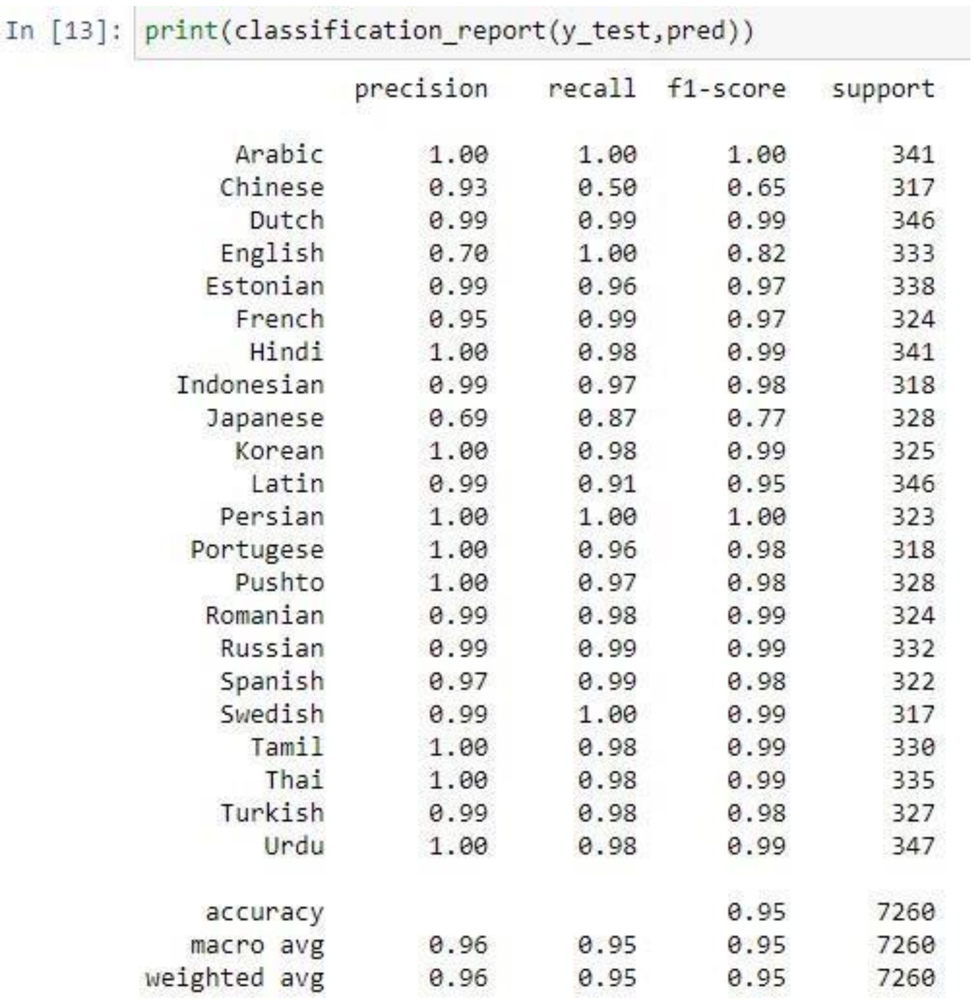


```
In [3]: data.info()    #dataset do not contain any null value

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 22000 entries, 0 to 21999
        Data columns (total 2 columns):
         #   Column    Non-Null Count  Dtype
        ---  ------    --------------  -----
         0   Text      22000 non-null  object
         1   language  22000 non-null  object
        dtypes: object(2)
        memory usage: 343.9+ KB

In [4]: data.isnull().sum()          # returns the number of missing values in the data set.

Out[4]: Text        0
        language    0
        dtype: int64
```

we had checked using functions data.info() and data.isnull().sum(). The output is attached below:

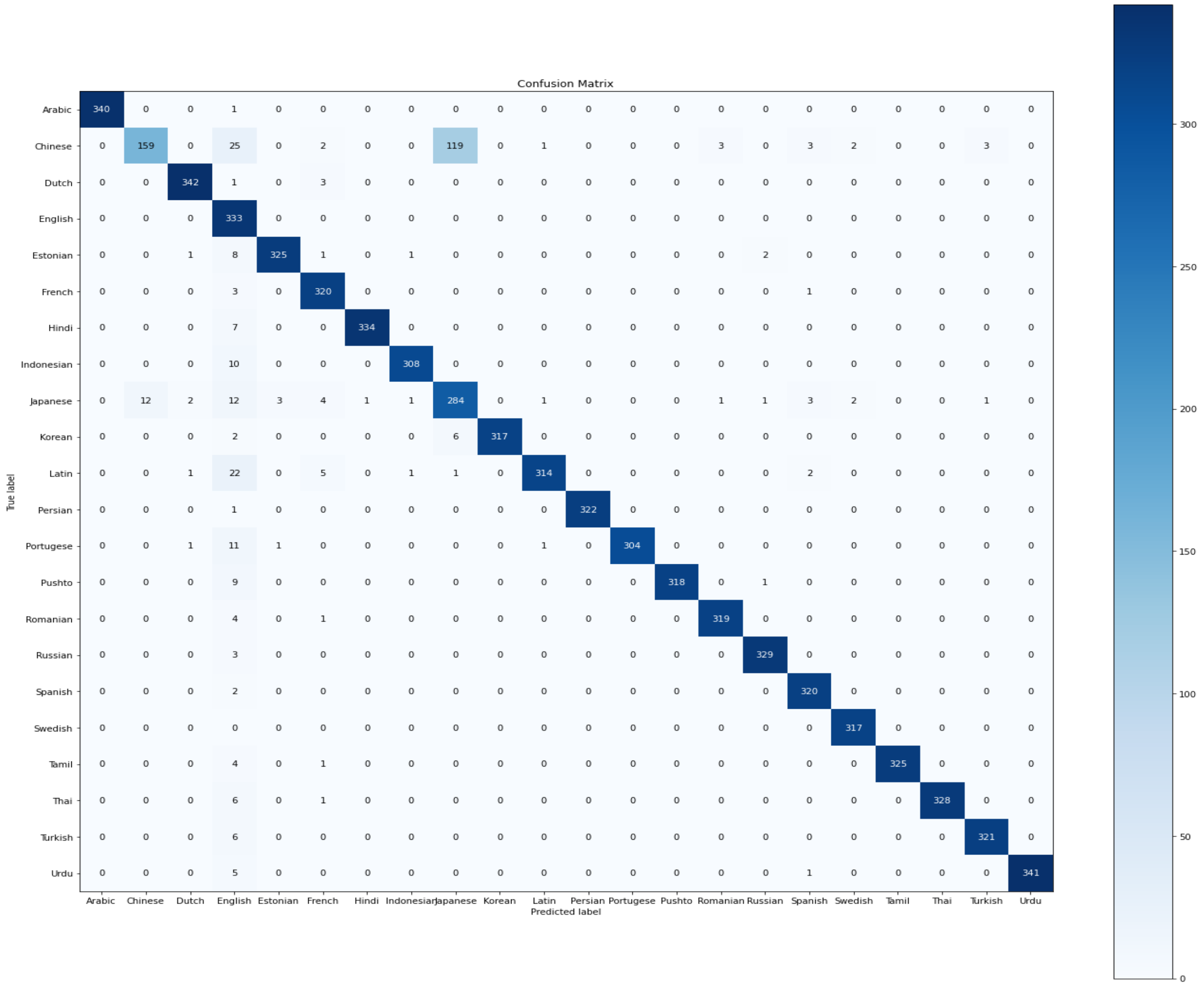
We had also used count vectorizer function. Count Vectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. This is helpful when we have multiple such texts, and we wish to convert each word in each text into vectors (for using in further text analysis). Count Vectorizer creates a matrix in which each unique word is represented by a column of the matrix, and each text sample from the document is a row in the matrix. The value of each cell is nothing but the count of the word in that particular text sample.

The next step is to create the training set, for training the model and test set, for evaluating the training set. For this process, we are using a train test split. We used 67% of data for training and 33% of data for testing. We had used different machine Learning algorithms like Gaussian Naïve Bayes, Bernoulli Naïve Bayes, Multinomial Naïve Bayes, SVC, Linear SVC, Logistic Regression, Random Forest Classifier, Decision Tree Classifier, MLP Classifier, K Neighbors Classifier(n neighbors =4). We tested all algorithms one by one. But we found that Multinomial Naïve Bayes Classifier gives an accuracy of 95% at a test time. MLP classifier had an accuracy of 0.97 but it is good at training time but not at testing time which leads to overfitting. Multinomial Naïve Bayes Classifier model most of the time gives correct results as we can see their precision, recall, f1-score and support measure.



```
In [13]: print(classification_report(y_test,pred))
              precision    recall  f1-score   support

      Arabic       1.00      1.00      1.00       341
     Chinese       0.93      0.50      0.65       317
       Dutch       0.99      0.99      0.99       346
     English       0.70      1.00      0.82       333
    Estonian       0.99      0.96      0.97       338
      French       0.95      0.99      0.97       324
       Hindi       1.00      0.98      0.99       341
  Indonesian       0.99      0.97      0.98       318
    Japanese       0.69      0.87      0.77       328
      Korean       1.00      0.98      0.99       325
       Latin       0.99      0.91      0.95       346
     Persian       1.00      1.00      1.00       323
   Portugese       1.00      0.96      0.98       318
      Pushto       1.00      0.97      0.98       328
    Romanian       0.99      0.98      0.99       324
     Russian       0.99      0.99      0.99       332
     Spanish       0.97      0.99      0.98       322
     Swedish       0.99      1.00      0.99       317
       Tamil       1.00      0.98      0.99       330
        Thai       1.00      0.98      0.99       335
     Turkish       0.99      0.98      0.98       327
        Urdu       1.00      0.98      0.99       347

    accuracy                           0.95      7260
   macro avg       0.96      0.95      0.95      7260
weighted avg       0.96      0.95      0.95      7260
```

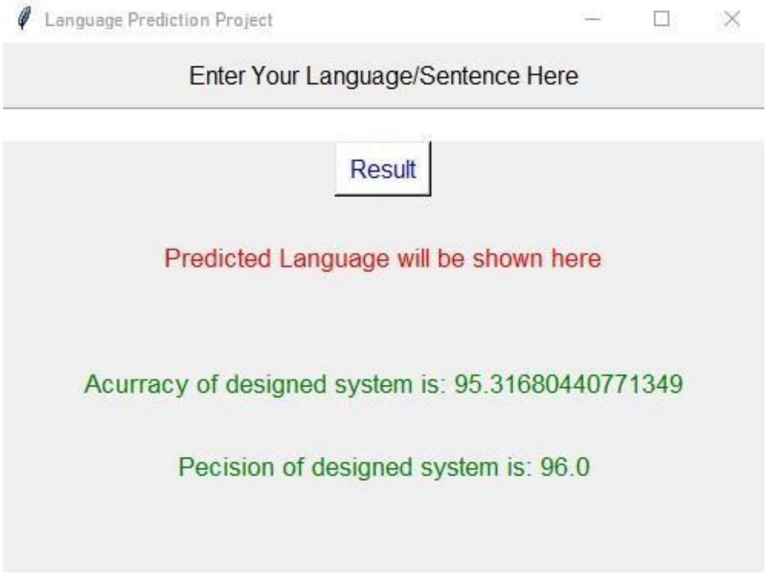
The final classification for each language group is captured in the confusion matrix. A confusion matrix is a table that is used to define the performance of a classification algorithm. It presents a table layout of the different outcomes of the prediction and results of a classification problem and helps visualize its outcomes. It visualizes and summarizes the performance of a classification algorithm. All the True Positives will be along the diagonal.
The other values will be False Positives or False Negatives.



	Arabic	Chinese	Dutch	English	Estonian	French	Hindi	Indonesian	Japanese	Korean	Latin	Persian	Portugese	Pushto	Romanian	Russian	Spanish	Swedish	Tamil	Thai	Turkish	Urdu
Arabic	340	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chinese	0	159	0	25	0	2	0	0	119	0	1	0	0	0	3	0	3	2	0	0	3	0
Dutch	0	0	342	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
English	0	0	0	333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Estonian	0	0	1	8	325	1	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0
French	0	0	0	3	0	320	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Hindi	0	0	0	7	0	0	334	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Indonesian	0	0	0	10	0	0	0	308	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Japanese	0	12	2	12	3	4	1	1	284	0	1	0	0	0	1	1	3	2	0	0	1	0
Korean	0	0	0	2	0	0	0	0	6	317	0	0	0	0	0	0	0	0	0	0	0	0
Latin	0	0	1	22	0	5	0	1	1	0	314	0	0	0	0	0	2	0	0	0	0	0
Persian	0	0	0	1	0	0	0	0	0	0	0	322	0	0	0	0	0	0	0	0	0	0
Portugese	0	0	1	11	1	0	0	0	0	0	1	0	304	0	0	0	0	0	0	0	0	0
Pushto	0	0	0	9	0	0	0	0	0	0	0	0	0	318	0	1	0	0	0	0	0	0
Romanian	0	0	0	4	0	1	0	0	0	0	0	0	0	0	319	0	0	0	0	0	0	0
Russian	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	329	0	0	0	0	0	0
Spanish	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0
Swedish	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0
Tamil	0	0	0	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	325	0	0	0
Thai	0	0	0	6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	328	0	0
Turkish	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	321	0
Urdu	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	341

Results:

We have presented a machine learning based language identification scheme that achieves near perfect accuracy in classifying languages with about 95% accuracy. For final result, we had made a front end in which another screen appears after running a code. In that screen, a text box is present in which we have to enter a text to get our desired result. After entering a text in any language among the above mentioned languages present in data, this model predict that language with the accuracy of 95.32% and precision of 96%. Some results are shown in the figures below:

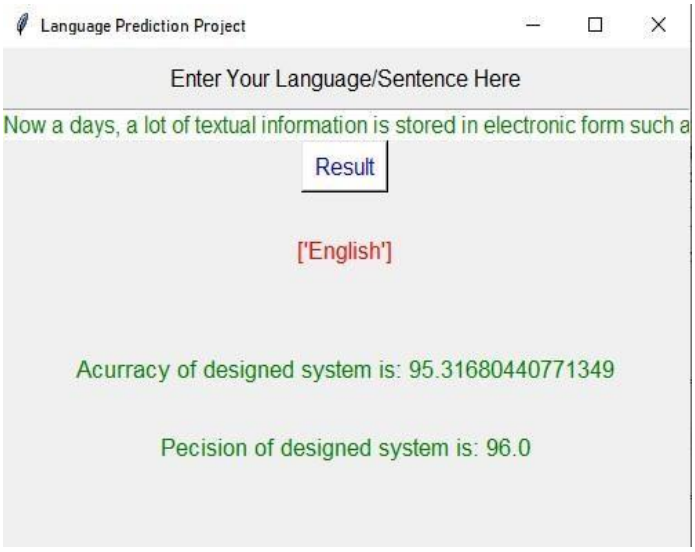


Enter Your Language/Sentence Here

Predicted Language will be shown here

Acurracy of designed system is: 95.31680440771349

Pecision of designed system is: 96.0



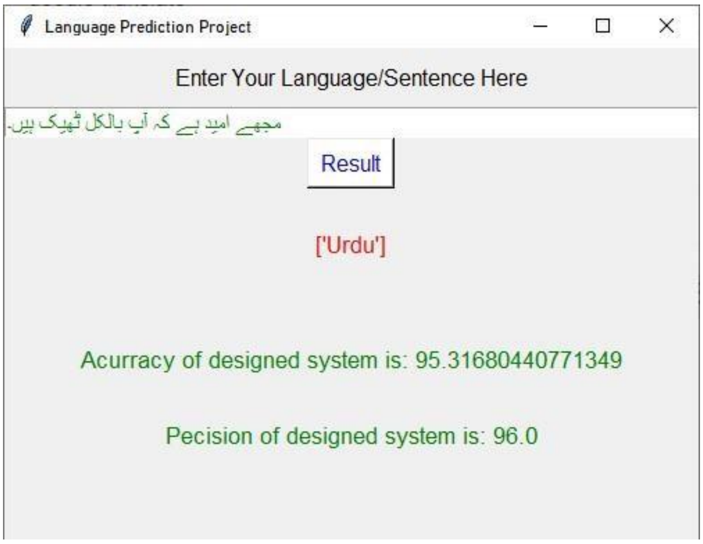
Enter Your Language/Sentence Here

Now a days, a lot of textual information is stored in electronic form such a

['English']

Acurracy of designed system is: 95.31680440771349

Pecision of designed system is: 96.0



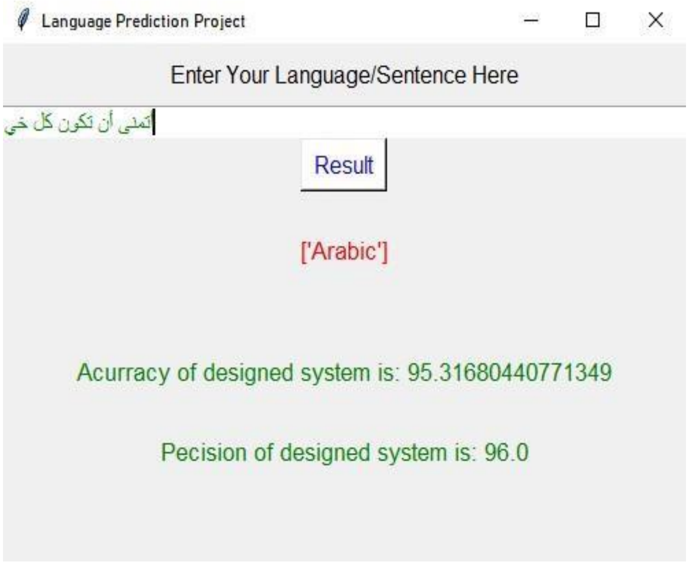
Enter Your Language/Sentence Here

مجھے امید ہے کہ آپ بالکل ٹھیک ہیں۔

['Urdu']

Acurracy of designed system is: 95.31680440771349

Pecision of designed system is: 96.0



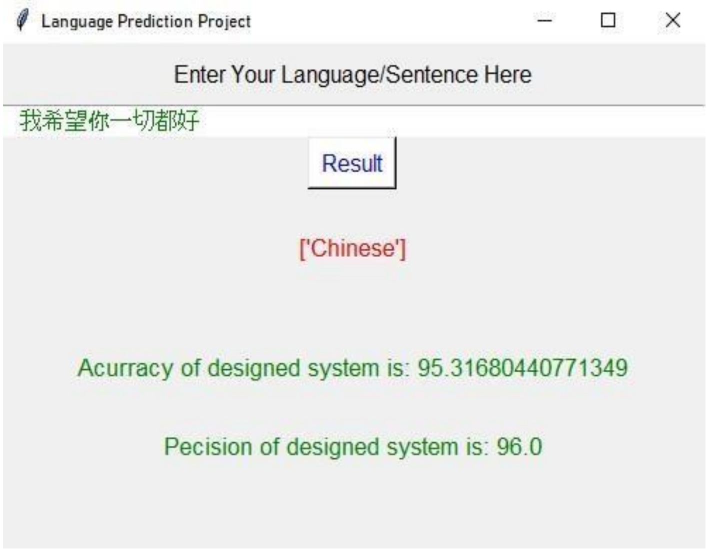
Enter Your Language/Sentence Here

أتمنى أن تكون كل خي

['Arabic']

Acurracy of designed system is: 95.31680440771349

Pecision of designed system is: 96.0



Enter Your Language/Sentence Here

我希望你一切都好

['Chinese']

Acurracy of designed system is: 95.31680440771349

Pecision of designed system is: 96.0

Conclusion:

We have presented a system for language identification in multilingual electronic text using a Multinomial Naïve Bayes Classifier inspired by supervised topic classification algorithms, combined with a document representation based on previous research in language identification for text. We showed that the system outperforms alternative approaches from the literature on synthetic data, as well as on real-world. We also showed that our system is able to accurately identify the language of the text with graphical user interface.

Future Work:

Living in such a modern era we're still facing the **problem that on call both sides individuals must speak in the same language only then they'll be able to communicate**. This project will **solve this problem by converting the language of one side individual to the desired language of the listener on another side**. Like if on one side the speaker speaks in Pashto so on the other side listener will listen in Punjabi and when the Punjabi speaks so on the other side the listener will listen in Pashto. The same scenario can be applied to all languages.

Also, this concept can be generalized for other programs like video listening directly in your desired language, automatically listening to someone in your desired language, etc.