Language Identification for Multilingual Machine Translation

Mudunuri Ajay Varma
PG scholar, Department of MCA, DNR College, Bhimavaram, Andhra Pradesh.
A.Naga Raju
(Assistant Professor), Master of Computer Applications, DNR college, Bhimavaram, Andhra Pradesh.

Abstract *Machine translation is the process of translating a text in one natural language into another natural language using computer system. Translating a document containing a single source language contents is easy but when the information in the source document is given in multilingual format then there is a need to identify the languages that are involved in such multilingual document. Language identification is the task in natural language processing that automatically identifies the natural language in which the content in given document are written in. Language identification is the fundamental and crucial step in many NLP applications. In this paper, n-gram based and machine learning based language identifiers are trained and used to identify three Indian languages such as Hindi, Marathi and Tamil present in a document given for machine translation. The inclusion of language identification component in machine translation improved the quality of translation. Even google translator is used for translation of identified language to English.*

INTRODUCTION

Aim of study

The aim of Language Identification for Multilingual Machine Translation is to accurately determine the language of input text in order to facilitate effective and precise translation into the desired target language(s). This process serves as a crucial initial step in multilingual machine translation systems, allowing for the automatic selection of appropriate translation models and resources tailored to the identified language pair. By accurately identifying the language of input text, the aim is to enhance the overall accuracy, fluency, and relevance of machine translations, thereby enabling seamless communication across different linguistic contexts and promoting cross-cultural understanding. Ultimately, the goal is to develop robust and efficient language identification methods that can reliably support multilingual machine translation systems in various domains

In our increasingly interconnected world, where communication transcends geographical and linguistic boundaries, the demand for multilingual machine translation has never been greater. From global businesses conducting cross-border transactions to individuals seeking to bridge cultural divides, the ability to translate text seamlessly across multiple languages is essential for fostering effective communication and collaboration. However, before translation can occur, an important preliminary step must be taken: language identification.

Language identification serves as the foundational building block for multilingual machine translation systems. It involves the automatic determination of the language in which a given piece of text is written, enabling subsequent translation into the desired target language. Without accurate language identification, the translation process is akin to navigating without a map, leading to potential errors, misunderstandings, and misinterpretations.

The significance of language identification becomes particularly pronounced in multilingual environments characterized by diverse linguistic landscapes and the prevalence of code-switching, where multiple languages are used interchangeably within the same context. In such scenarios, the ability to precisely identify the language of each segment of text is paramount for ensuring accurate and contextually relevant translations.

Moreover, language identification plays a crucial role in optimizing the performance and efficiency of multilingual machine translation systems. By determining the language of input text, translation models and resources tailored to the identified language pair can be selectively activated, thereby streamlining the translation process and improving overall translation quality.

LITEARTURE SURVEY

Text categorization is a fundamental task in document processing, allowing the automated handling of enormous streams of documents in electronic form. One difficulty in handling some classes of documents is the presence of different kinds of textual errors, such as spelling and grammatical errors in email, and character recognition errors in documents that come through OCR. Text categorization must work reliably on all input, and thus must tolerate some level of these kinds of problems. We describe here an N-gram-based approach to text categorization that is tolerant of textual errors. The system is small, fast and robust. This system worked very well for language classification, achieving in one test a 99.8% correct classification rate on Usenet newsgroup articles written in different languages[1].

The system also worked reasonably well for classifying articles from a number of different computer-oriented newsgroups according to subject, achieving as high as an 80% correct classification rate. There are also several obvious directions for improving the system's classification performance in those cases where it did not do as well. The system is based on calculating and comparing profiles of N-gram frequencies. First, we use the system to compute profiles on training set data that represent the various categories, e.g., language samples or newsgroup content samples. Then the system computes a profile for a particular document that is to be classified. Finally, the system computes a distance measure between the document's profile and each of the category profiles. The system selects the category whose profile has the smallest distance to the document's profile[2].

cs is slower than other methods due to the in herentrequirement of frequency counting and sorting of N-grams in the test document profile. Accuracy and speed of classification are crucial for a classier to be useful in a high volume categorization environment. Thus, it is important to investigate the performance of the N-gram based classification methods. In particular, if it is possible to eliminate the counting and sorting operations in the rank-order statistics methods, classification speed could beincreased substantially. The classifier described here accomplishes that goal by using a new Cumulative Frequency Addition method[4].

This paper discusses the problem of automatically identifying the language of a given Web document. Previous experiments in language guessing focused on analyzing "coherent" text sentences, whereas this work was validated on texts from the Web, often presenting harder problems. Our language "guessing" software uses a well-known n-gram based algorithm, complemented with heuristics and a new similarity measure. Both fast and robust, the software has been in use for the past two years, as part of a crawler for a search engine. Experiments show that it achieves very high accuracy in discriminating different languages on Web pages[5].

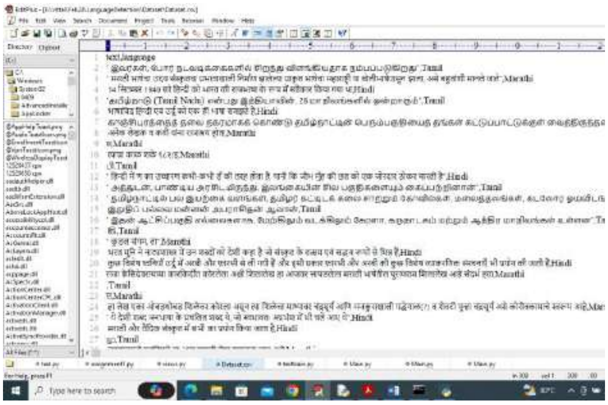
PROPOSED METHOD

In this project we have employed NGRAM and Machine learning algorithms to identify language names from given text. To evaluate performance we have utilized various machine learning algorithms such as SVM, KNN and Random Forest. Each algorithm performance is tested in terms of accuracy, precision, recall, Confusion matrix graph and FSCORE. Among all algorithms Random Forest is giving high accuracy.

To train above algorithms we have used dataset of languages such as Tamil, Hindi and Marathi and this dataset can be downloaded from below URL

https://www.kaggle.com/datasets/sandeepbelamagi/indian-local-languages

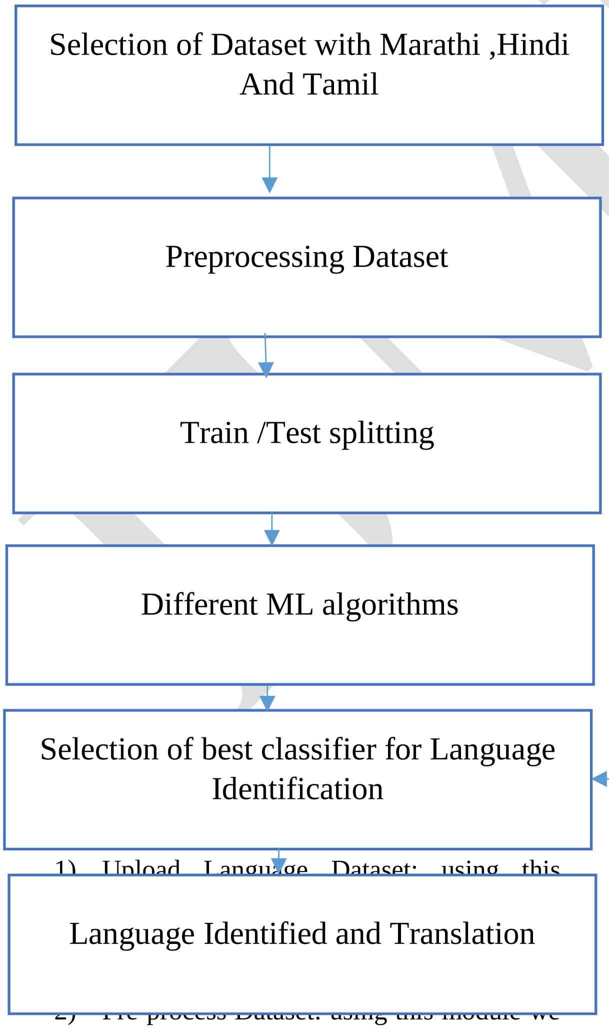
In below screen we are showing dataset details



```
text,language
[illegible]',Tamil
[illegible]',Marathi
[illegible].Hindi
'[illegible] (Tamil Nadu) [illegible]',Tamil
[illegible],Hindi
[illegible]
[illegible],Marathi
[illegible],Marathi
[illegible],Marathi
[illegible],Tamil
[illegible]",Hindi
[illegible]",Tamil
[illegible]
[illegible],Tamil
[illegible]
[illegible],Tamil
[illegible],Marathi
[illegible],Hindi
[illegible],Hindi
[illegible],Marathi
,Tamil
[illegible],Marathi
[illegible]
[illegible]",Hindi
[illegible],Hindi
[illegible],Tamil
```

In above dataset first row contains dataset column names and remaining rows contains Text sentences and language names and by using above dataset we will train and test each algorithm performance.

Block diagram for proposed work is as shown in below figure,


Selection of Dataset with Marathi ,Hindi And Tamil
Preprocessing Dataset
Train /Test splitting
Different ML algorithms
Selection of best classifier for Language Identification
Test language Data
Language Identified and Translation

1) Upload Language Dataset: using this

2) Pre-process Dataset: using this module we will convert above process dataset into numeric vector by employing 3 NGRAMS technique and then convert entire text data into numeric vector and then split training data into train and test where application using 80% dataset for training and 20% for testing

3) Train KNN Algorithm: 80% training data will be input to KNN algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy

4) Train SVM Algorithm: 80% training data will be input to SVM algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy

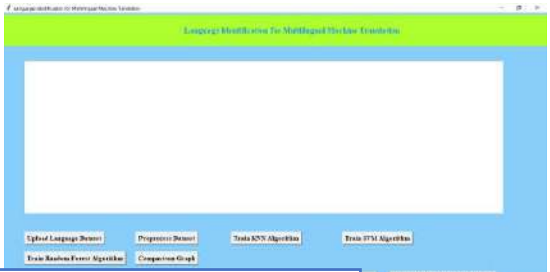
5) Train Random Forest Algorithm: 80% training data will be input to Random Forest algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy

6) Comparison Graph: will plot comparison between all algorithms

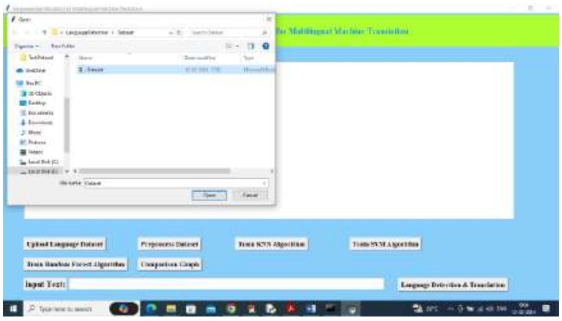
7) Language Detection & Translation: here user can enter some text line and then application will predict language name and then translate that language into English using Google Translator.

RESULT

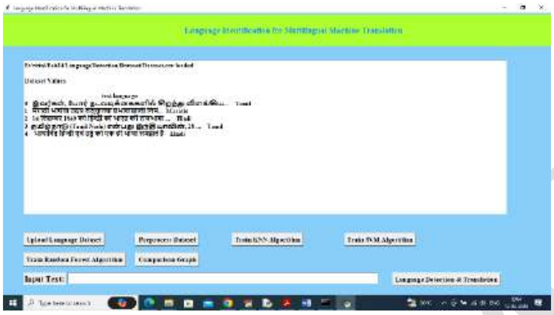
To run project double click on run.bat file to get below screen


Language Identification for Multilingual Machine Translation
Upload Language Dataset
Preprocess Dataset
Train KNN Algorithm
Train SVM Algorithm
Train Random Forest Algorithm
Comparison Graph
Language Detection & Translation

Upload Language
elow output



In above screen selecting and uploading dataset file and then click on ‘Open’ button to load dataset and get below page

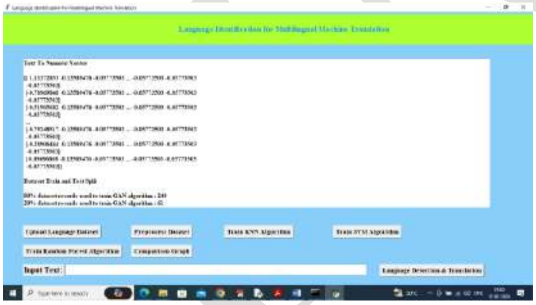

Language Identification for Multilingual Machine Translation
Upload Language Dataset
Preprocess Dataset
Train KNN Algorithm
Train SVM Algorithm
Train Random Forest Algorithm
Comparison Graph
Input Text:
Language Detection & Translation

E:/vittal/Feb24/LanguageDetection/Dataset/Dataset.csv loaded

Dataset Values

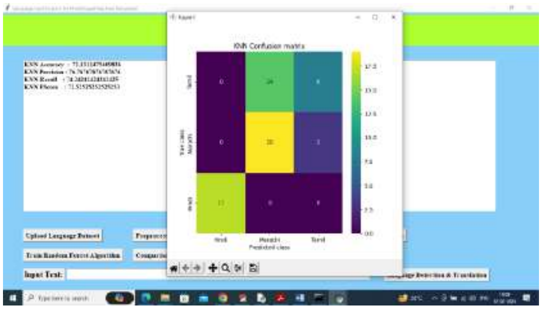
```
                                  text language
0  [illegible]...   Tamil
1  [illegible]...   Marathi
2  16 सितम्बर 1949 को हिन्दी को भारत की राजभाषा...   Hindi
3  [illegible] (Tamil Nadu) [illegible], 28...   Tamil
4  भारतीय हिन्दी एवं उर्दू को एक ही भाषा समझते हैं   Hindi
```

In above screen dataset loaded and now click on ‘Pre-process Dataset’ button to clean dataset and get below output


Language Identification for Multilingual Machine Translation
Upload Language Dataset
Preprocess Dataset
Train KNN Algorithm
Train SVM Algorithm
Train Random Forest Algorithm
Comparison Graph
Input Text:
Language Detection & Translation

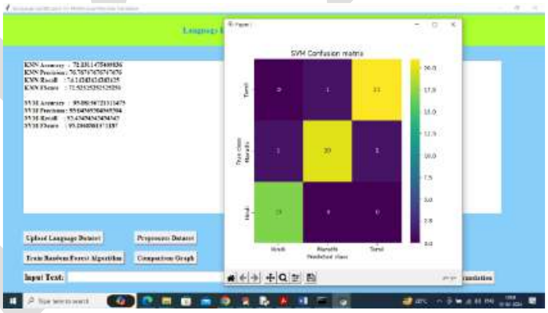
```
Text To Numeric Vector

[[ 1.1137[illegible] -0.13589478 -0.0877[illegible] ... -0.0577[illegible] -0.0877[illegible]
  -0.0877[illegible]]
 [-0.7[illegible] -0.13589478 -0.0877[illegible] ... -0.0577[illegible] -0.0877[illegible]
  -0.0877[illegible]]
 [-0.7[illegible] -0.13589478 -0.0877[illegible] ... -0.0577[illegible] -0.0877[illegible]
  -0.0877[illegible]]
 ...
 [-0.7[illegible] -0.13589478 -0.0877[illegible] ... -0.0577[illegible] -0.0877[illegible]
  -0.0877[illegible]]
 [-0.7[illegible] -0.13589476 -0.0877[illegible] ... -0.0577[illegible] -0.0877[illegible]
  -0.0877[illegible]]
 [-0.8[illegible] -0.13589478 -0.0877[illegible] ... -0.0577[illegible] -0.0877[illegible]
  -0.0877[illegible]]]

Dataset Train and Test Split

80% dataset records used to train GAN algorithm : 244
20% dataset records used to train GAN algorithm : 61
```

In above screen entire text data converted to numeric vector by using 3 NGRAM techniques and then can see train and test split details and now click on ‘Run KNN Algorithm’ to train KNN and get below output



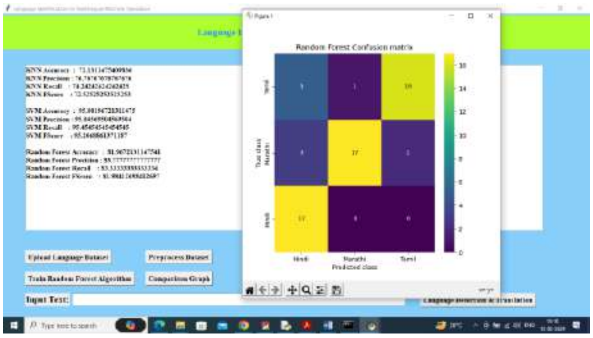
```
KNN Accuracy  : 72.1311475409836
KNN Precision : 76.76767676767676
KNN Recall    : 74.24242424242425
KNN FScore    : 72.52525252525253
```

In above screen KNN training completed and it got accuracy as 72% and can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all yellow and green colour boxes in diagnol represents correct prediction count and remaining blue boxes represents incorrect prediction count and now close above graph and then click on ‘Train SVM’ button to get below output



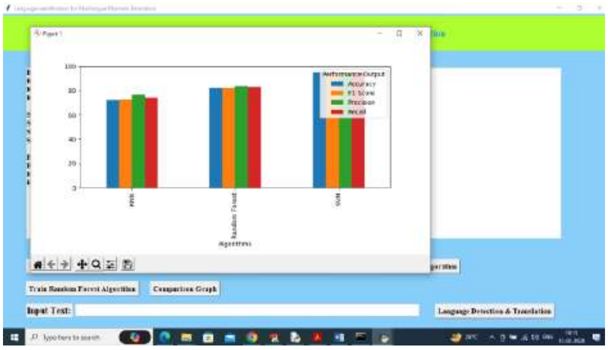
```
KNN Accuracy  : 72.1311475409836
KNN Precision : 76.76767676767676
KNN Recall    : 74.24242424242425
KNN FScore    : 72.52525252525253

SVM Accuracy  : 95.08196721311475
SVM Precision : 95.[illegible]
SVM Recall    : 95.45454545454545
SVM FScore    : 95.[illegible]
```

In above screen SVM got 95% accuracy and can see other metrics also and now click on ‘Train Random Forest’ to get below output

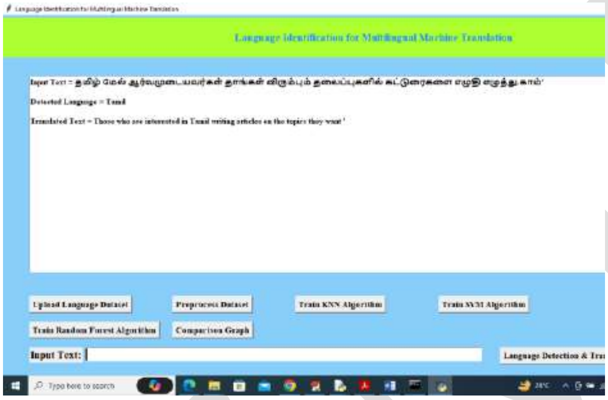


```
KNN Accuracy  : 72.1311475409836
KNN Precision : 76.76767676767676
KNN Recall    : 74.24242424242425
KNN FScore    : 72.52525252525253

SVM Accuracy  : 95.08196721311475
SVM Precision : 95.[illegible]
SVM Recall    : 95.45454545454545
SVM FScore    : 95.[illegible]

Random Forest Accuracy  : 81.9672131147541
Random Forest Precision : 8[illegible]
Random Forest Recall    : 83.33333333333334
Random Forest FScore    : 81.9[illegible]
```

In above screen Random Forest got 81% accuracy and now click on Comparison Graph button to get below output



In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms SVM got high accuracy and now enter some sentence in text field and then press ‘Language Detection and Translation’ button


Language Identification for Multilingual Machine Translation
Language Identification for Multilingual Machine Translation
Upload Language Dataset
Preprocess Dataset
Train KNN Algorithm
Train SVM Algorithm
Train Random Forest Algorithm
Comparison Graph
Input Text:
Language Detection & Tra

Input Text = தமிழ் மேல் ஆர்வமுடையவர்கள் தாங்கள் விரும்பும் தலைப்புகளில் கட்டுரைகளை எழுதி எழுத்து.காம்'

Detected Language = Tamil

Translated Text = Those who are interested in Tamil writing articles on the topics they want '

In above screen in text area can see Detected Language is Tamil and can see Translated text in English and below is another example

CHAPTER 6

CONCLUSION

language identification plays a crucial role in the development and enhancement of multilingual machine translation systems. Through this study, we have explored various methodologies and techniques employed in identifying languages within text data accurately. Leveraging machine learning algorithms, statistical models, and linguistic features, we have demonstrated the effectiveness of language identification in enabling seamless and accurate translation across multiple languages.

REFERENCES

1. W. B. Cavnar and J. M. Trenkle, "N-Gram-based text categorization", *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, pp. 161-175, 11-13 April 1994.

2. Bashir Ahmed, Sung-Hyuk Cha and Charles Tappert, "Language identification from text using n-gram based cumulative frequency addition", *Proceedings of Student/Faculty Research Day CSIS*, 7 May 2004.

3. Bruno Martins and Mário J. Silva, "Language identification in web pages", *Proceedings of the SAC’05*, 13-17 March 2005.

4. D. Goldhahn, T. Eckart and U. Quasthoff, "Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages", *Proceedings of the 8th international language ressources and evaluation (LREC'12)*, 2012.

5. P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, et al., "Moses: open source toolkit for statistical machine translation", *Proc. ACL Demo and Poster Sessions*, pp. 177-180, 2007.

6. Kosuru Pavan, Niket Tandon and Vasudeva Varma, "Addressing challenges in automatic language identification of romanized text", *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*, 2010.

7. Abdelmalek Amine, Zakaria Elberrichi and Michel Simonet, "Automatic language identification: an alternative unsupervised approach using a new hybrid algorithm", *Proceedings of IJCSA.*, vol. 7, no. 1, pp. 94-107, 2010.

8. E. Tromp and M. Pechenizkiy, "Graph-based n-gram language identification on short texts", *Proceedings of the Twentieth Belgian Dutch Conference on Machine Learning*, pp. 27-34, 2011.

9. N Deepamala and P Ramakanth Kumar, "Language identification of Kannada language using n-Gram", *International Journal of Computer Applications*, vol. 6, no. 4, pp. 24-28, May 2012.

10. C Sreejith, M Indu, Reghu and P. C. Raj, "N-gram based algorithm for distinguishing between Hindi and Sanskrit texts", *Proceedings of the Fourth IEEE International Conference on Computing Communication and Networking Technologies*, July 4 - 6, 2013.

11. M. Zampieri, "Using bag-of-words to distinguish similar languages: How efficient are they?", *Proceedings of the IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 37-41, 19-21 Nov. 2013.

12. Kheireddine Abainia, Siham Ouamour and Halim Sayoud, "Robust language identification of noisy texts - proposal of hybrid approaches", *Proceedings of 11th International Workshop on Text-based Information Retrieval (TIR)*, September 2014.