Project Review 3

CSE4022

Natural Language Processing

Slot: E2+TE2

Topic: Indian Language Detection using NLP

Submitted by: Krushn Pathak 20BCE0580

Rahulkumar Ankola 20BCE0530

Jayendra Awasthi 20BCE0536

Submitted to: Dr. Priya G

School of Computer Science and Engineering
Vellore Institute of Technology
Vellore

1. Abstract

In the era of information explosion, the internet is flooded with multilingual content, with a majority being in natural languages. However, the availability of resources for training data poses a challenge, especially for low-resourced languages like Indian regional languages, which are considered to be disadvantaged compared to other languages. Native-language identification (NLI) is the task of determining the native language of a text from a multilingual document. In this project, we propose a hybrid approach that combines rule-based and machine learning approaches with natural language processing, machine learning, and human input to detect the language of given data. This approach is designed to work efficiently even with limited training data, providing increased flexibility, faster iteration, and reduced strain on resources. This approach is well-suited for conversational experiences and task-oriented projects, where language detection is a crucial component.

2. Introduction

Natural language, which evolves through human usage over time, is the primary means of communication for exchanging knowledge, emotions, and feelings. With numerous native languages existing in different parts of the world, each with its unique alphabet, signs, and grammar, processing data in languages other than English poses challenges for computers. Despite the complexity, there have been significant research efforts and applications developed for handling natural languages in various real-time scenarios, such as chatbots, text-to-speech conversion, language identification, hands-free computing, spell-check, summarizing electronic medical records, and sentiment analysis, among others.

In this project, our focus is on Indian language detection using natural language processing (NLP) techniques. India is known for its diverse linguistic landscape, with over 120 major languages spoken across the country. However, Indian regional languages are considered to be low-resourced compared to other languages, which makes language detection a challenging task. To address this, we propose a hybrid approach that combines the best rule-based and machine learning approaches with NLP, machine learning, and human input. We will utilize a language detection dataset containing different types of languages, and our approach aims to recognize the language of given data accurately and efficiently. This project has potential applications in various domains, including conversational experiences and task-oriented projects, where language detection is critical for effective communication and understanding.

3. Literature Review

S. No.	Reference	Method/Algorithm	Challenges	Observations
1	[1]	First, the authors clustered the phones to phonetically motivated clusters such as front vowels, back vowels, consonants, voiced stops, unvoiced stops, etc. Then, they developed single gaussian mono-phone based acoustic models for all the phones in the two languages separately without any mapping. The well-known Bhattacharyya distances are calculated for the phone to be mapped to the all the phones in the same phonetic cluster of the target language. After doing the necessary mapping of phones across the two languages, they are required to develop a triphone based speech recognition system for the multilingual phone set.	One language differs from another language in one or more of the following: Phonology: Phone sets would be different for each of the languages. Morphology: The word roots and the lexicons are different. Syntax: The sentence patterns are different. Prosody: Duration, pitch, and stress patterns	The multilingual speech recognition system was trained using 10 hours of PSTN quality Tamil speech and 8 hours of PSTN quality Hindi speech recorded with 8 kHz sampling frequency. There are 74 speakers enrolled in the Tamil training corpus and 105 speakers for Hindi. For the test set, we used a total of 72 utterances from 16 speakers for Tamil and 85 utterances from 16 speakers for Hindi. It may be noted the system has performed well for both the languages in comparison with the results reported.
2	[2]	Recurrent Neural Network (RNN) based architecture was used for ALI of library catalogue entries. However, they observe that a statistical model based on trigram statistics outperform the proposed RNN based architecture.	ALI for discriminating similar languages ALI for large number of languages ALI in the social media domain Word-level and sentence level ALI	Various methods of ALI over social media text at different levels of granularities have been explored. The features used include text features (e.g. character level features, Wordlevel features, sentence-level features), non-textual features such as POS tags as well as social and conversational features (user features, link features, conversation features etc.). However, the existing approaches still exhibit limitations in terms of their ability to

				handle naturally embedded borrowed words and ambiguous word forms commonly prevalent in social media content.
3	[3]	Machine learning algorithms were used from the scikit-learn machine leaning framework:11 KNeighborsClassifier (KNN), NearestCentroid (NC), ExtraTreesClassifier (EXT), RandomForestClassifier (RF), and GradientBoostingClassifier (GB), and Support Vector Machines (SVM), and report only the results of the best classifiers based on NearestCentroid (NC).	Collecting human annotations of complex words and phrases1 by both native and non-native speakers in three languages (English, German, and Spanish), and for English, for three different text genres; Proposing a language independent set of features to build state-of-the-art automated CWI systems for all three languages Showing that CWI systems using the language-independent feature set can be successfully trained on a dataset in one language and applied on another language, thus reducing the need for compiling CWI datasets for various languages.	Complex word identification (CWI) task is an important task in text accessibility and text simplification. So far, however, this task has only been addressed on the Wikipedia sentences and taking into account mostly the needs of nonnative English speakers. Moreover, languages other than English did not receive any attention with regard to building either the CWI datasets or automated CWI systems.

4	[4]	In this paper, supervised classification approach was used. We devise and run experiments using several models that capture different types of linguistic information. For each model, features are extracted from the texts and a classifier is rained to predict the L1 labels using the features. We use a linear Support Vector Machine to perform multiclass classification in our experiments.	One objective is to investigate the efficacy of the type of features that have been common to almost all NLI approaches to date for several languages which are significantly different from English. Another issue that arises in this type of multilingual research is the use of multiple part-of speech tag sets developed for different languages. Differences in the granularity of the tags mean that they are often not directly comparable.	The present study has examined a number of different issues from a cross-lingual perspective, making a number of novel contributions to NLI research. Using up to six languages to inform our research, this experiment used evidence from multiple languages to support their results and to identify general patterns that hold across multiple languages.
5	[5]	This model takes source letters as input and provides a language label for each of them. Whenever we need to recognize the language of a document, we take the language assigned by our model to the majority of letters. The method proposed is designed for short text without relying on document boundaries.	There is no established dataset for the novel setting of text partitioning by language, we evaluated our model in several common tasks (monolingual and multilingual language identification for long and short texts) which were previously handled by separate algorithms.	In this paper, authors have developed a language identification algorithm based on bidirectional recurrent neural networks. The approach is designed for identifying languages on a short-texts, allowing to detect code switching including switches to formal markup languages like HTML.

6	[6]	A representation of text is selected A model for each language is derived from corpora where the languages are known A function is defined that determines the similarity between text and each language The highestscoring model determines the language of the text predicted by the system.	Similar Languages, Language Varieties, and Dialects Short Texts Unseen Languages and Unsupervised LI	This article has presented a comprehensive survey on language identification of digitally encoded text. We have shown that LI is a rich, complex, and multi-faceted problem that has engaged a wide variety of research communities. LI accuracy is critical as it is often the first step in longer text processing pipelines, so errors made in LI will propagate and degrade the performance of later stages.
7	[7]	3 basic statistical methods (Character Based Algorithm, Word Based Algorithm, Special Character based Algorithm) to identify language in the text and 2 hybrid approaches to improve the performance and accuracy	On applying the statistical models, these methods work fine on clean or long texts but fails on short and corrupted text. There exist similar approach using KNN and other models for language identification of noisy texts	Hybrid approaches provide high performances and seem to be better than Google Translator and Microsoft Word. In Microsoft Word Persian texts were recognized as Arabic texts. Turkish texts were recognized as French texts. In Google Translator some Malay texts were recognized as Indonesian ones, some
				Latin texts were recognized as Italian
8	[8]	In this paper, i-vector representation of the speech signal was used to detect the native language of an English speaker. Two different i-vector extraction strategies were used: language-independent and language-dependent.	This approach is complex and is similar to an approach that has been already successfully implemented for the problem of language Identification	In the language-independent, a unique language-independent ivector extractor is estimated for all the native language classes. The language-dependent strategy consists of training one native language-dependent ivector for each native language class

9	[9]	In this paper, a hybrid approach for identifying a language which is a combination of Vector Quantization (VQ) and Gaussian Mixture Models (GMM) is used for speech recognition	This method lack accuracy. The hybrid method which is a combination of VQ and GMM method can be used to improve this model accuracy.	This paper discussed the application of hybrid VQ-GMM model which is proved to be better than GMM model.
10	[10]	In this paper language identification based on deep neural networks (DNN), i-vector paradigm and convolutional neural networks (CNN) is discussed.	Other studies focus on deep neural networks or other conventional approaches such as Gaussian mixture models and supervector-based identification methods shows more promising results	In this study, for the identification of the 50 inset languages, EERs of 3.6% and 3.5% were obtained using DNN and CNN, respectively.
11	[11]	The considerable task is to recognize the features that can distinguish between languages clearly and efficiently. The model uses audio files and converts those files into spectrogram images. It applies the convolutional neural network (CNN) to bring out main attributes or features to	Various languages in the world belong to the Indo Persian and European families. In this group, the languages are separated into three subparts: Germanic, Romance, and Slavic. Our model confuses those languages	The presented work discusses various methods which attain state-of-the-art results using four different datasets with audio, and the first dataset contains three languages, the second dataset includes 22 languages, and the third dataset

		detect output easily. The main objective is to detect languages out of English, French, Spanish, and German, Estonian, Tamil, Mandarin, Turkish, Chinese, Arabic, Hindi, Indonesian, Portuguese, Japanese, Latin, Dutch, Portuguese, Pushto, Romanian, Korean, Russian, Swedish, Tamil, Thai, and Urdu. An experiment was conducted on different audio files using the Kaggle dataset named spoken language identification. These audio files are comprised of utterances, each of them spanning over a fixed duration of 10 seconds. The whole dataset is split into training and test sets. Preparatory results give an overall accuracy of 98%. Extensive and accurate testing show an overall accuracy of 88%.	with the same words; for example, “Cat” word in English, “Chatte” word in French, “Kat” word in Dutch, and “Katze” in German all have the same sound and pronunciation; hence, it is very difficult for a model to understand. Our model confuses Russian (Ru) and French (Fr) because they have similar accents; many words are adopted from French to Russian, so it is very difficult to give accurate results.	includes 16 languages. All are available on the Kaggle and fourth Mozilla common voice dataset contains four languages and is available on the Mozilla website. In the image domain, 2D convolutional neural networks obtained an accuracy of 98%. In another dataset of CSV file, word embedding using the pretrained model obtained an accuracy of 95%. With Bernoulli Naïve Bayes approach, we obtained an accuracy of 93% on a 22- language dataset. Using the SVM and random forest classifier model achieved 82.88% and 72.42% accuracy on the 16language dataset.

12	[12]	The main objective of this chapter is to examine the automatic techniques that can be applied to learner corpora to identify learners' mother tongue backgrounds from their patterns of production in the target language. The chapter focuses particularly on aspects of learner corpus design that have a direct bearing on the results (size, topic homogeneity, type and level of annotation).Special attention is also paid to the contribution of NLI research to SLA, more particularly in the form of the detection-based approach to transfer. NLI is a relatively new line of enquiry: research so far has focused on identifying the	research would be one where all of the texts in the corpus are written on the same topic or at least in the same genre with a symmetrical distribution of topics across L1 groups, the texts are all of similar lengths or have similar length means and standard deviations across L1 groups, all of the learners are at precisely the same level of L2 proficiency or at least the levels of proficiency are evenly balanced across L1groups, the learners within and across groups have similar educational, socioeconomic and psychological profiles, and the learners within and	Results showed that combining the predictions of each individual type of feature in an ensemble model performed best in the four datasets. On the ICLE , the best ensemble model achieved an accuracy of 90.1% and a close look at the results for the individual types of features revealed that single words and bigrams, syntactically-based features and perplexity scores from 5-gramlanguage models were the most powerful predictors
		native language of writers using English as a foreign language and the chapter will thus deal with English as a target language.	across groups have had comparable amounts and types of instruction in and exposure to the target language.	

13	[13]	By keeping the objective of comprehension, the work officially created in the defined area, we directed a methodical writing survey. This paragraph portrays the ways received and the accomplished outcomes comprehensively. In this specific circumstance, we utilize the name report as an equivalent word for study, theory, or some other kind of content original copy. The most widely recognized methodology found in our orderly writing audit comprises of structure an algorithm based on machine learning for hate discourse characterisation. We additionally discovered that the most widely recognised calculations utilised are SVM, R.F (Random forest), and D.T (Decision Tree)	Finding the right highlights for a grouping issue could be one of the all the more requesting assignments while utilizing AI. Subsequently, we apportion this particular segment to portray the highlights officially utilized by other writers. We partition the highlights into two classifications: general highlights utilised in content mining, which are regular in other content mining areas; and particular detest discourse recognition highlights, which we found in loathe discourse discovery reports and are characteristically identified with the attributes of this issue	A methodical writing survey is directed to comprehend the cutting edge and openings in the field of programmed detest discourse recognition. This demonstrated to be a difficult assignment, for the most part in light of the fact that this subject has been generally examined in different fields, for example, sociologies and law, and along these lines we found an enormous number of archives that must need higher assets to process.
14	[14]	The paper evaluates the performance of various multilingual offensive language identification methods for the languages of India. The authors of the paper analysed the performance of various machine learning algorithms, including Support Vector Machines (SVM), Naive Bayes (NB), and Deep Neural Networks (DNN) for identifying offensive language in Indian languages.	The performance of the algorithms was affected by the availability and quality of annotated data, and that further efforts were needed to improve the performance of these methods.	

15	[15]	The paper focuses on the problem of language identification and named entity recognition in Hinglish code mixed tweets. Hinglish is a code-mixed language that consists of a mixture of Hindi and English languages. The authors of the paper proposed a language identification and named entity recognition system for Hinglish code mixed tweets that combines traditional NLP techniques and deep learning models. The system was trained on a dataset of Hinglish tweets and evaluated on a test set.	Identifying the language in code-mixed Hinglish tweets Dealing with noisy and inconsistent data in Hinglish tweets Obtaining annotated data for training and evaluating the system Accurately recognizing named entities in Hinglish code mixed tweets Integrating traditional NLP techniques and deep learning models to develop a robust language identification and named entity recognition system.	The results of the evaluation showed that the proposed system performed well in identifying the language and named entities in Hinglish code mixed tweets, with an accuracy of 93.65% for language identification and an F1 score of 83.63% for named entity recognition. The authors also concluded that the system performed well even when dealing with noisy and inconsistent data.

4. Uniqueness

- **Hybrid Approach:** Our project proposes a hybrid approach that combines rule-based and machine learning approaches with natural language processing, machine learning, and human input for native-language identification (NLI). This hybrid approach allows for efficient language detection even with limited training data, providing increased flexibility, faster iteration, and reduced strain on resources. This approach is unique in its combination of different techniques to improve the accuracy and efficiency of language detection.

- **Focus on Low-Resourced Languages**: Our project focuses on Indian regional languages, which are considered to be low-resourced compared to other languages. We address the challenge of limited resources for training data in low-resourced languages and propose a solution that can work effectively in such scenarios. This aspect adds uniqueness to our project as it caters to the specific needs of low-resourced languages.

- **Comprehensive Methodology**: Our project follows a comprehensive methodology that includes various steps such as data import, data processing, text pre-processing, data visualization, data normalization, dataset division, feature extraction, model training, model evaluation, and prediction. This holistic approach ensures that all necessary aspects of language detection using NLP models are covered, making our project robust and reliable.

- **Multiple NLP Models:** Our project utilizes three different NLP models, namely Random Forest Classifier, Linear SVC, and Logistic Regression, for language detection. This diversity of models allows for a comparison of their performance and accuracy, adding uniqueness to our project in terms of experimentation and evaluation of different models.

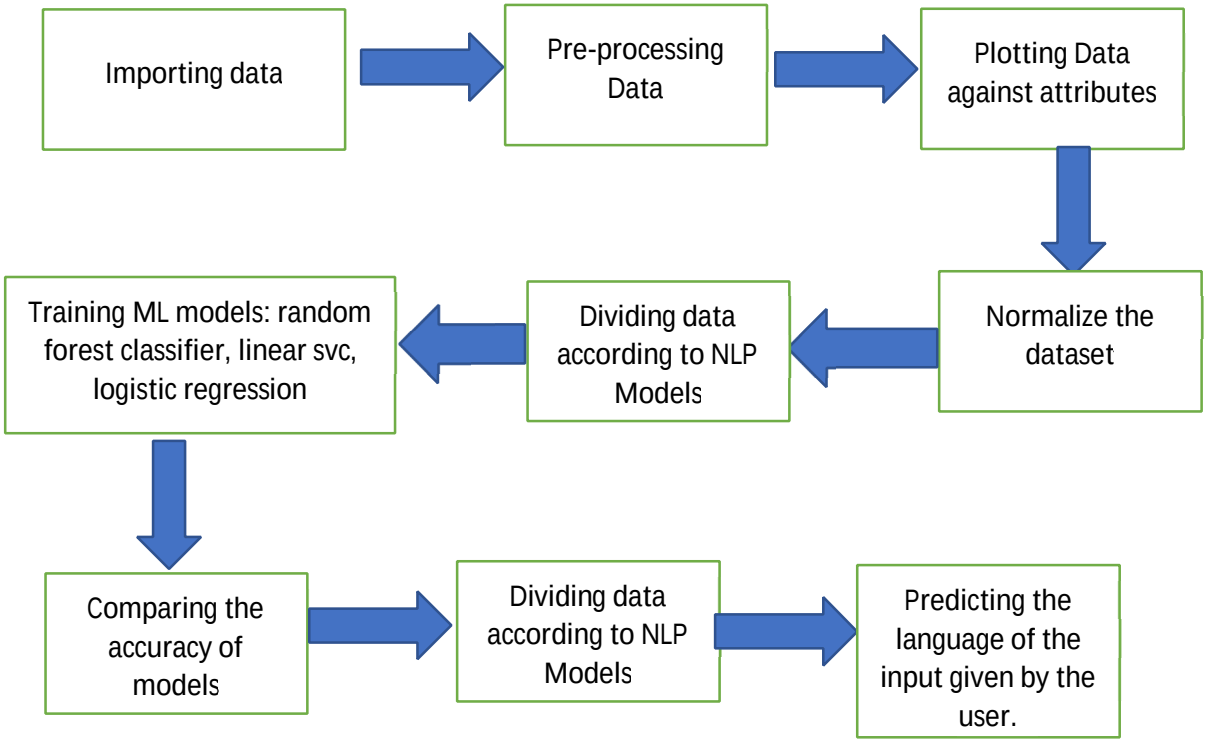
- **Practical Application:** Our project focuses on conversational experiences and task-oriented projects where language detection is a crucial component. This practical application aspect adds uniqueness to our project as it caters to real-world scenarios where language detection plays a significant role, such as chatbots, customer service, and content filtering.

5. Problem Statement and workflow

With approximately 7,000 languages spoken worldwide, language diversity poses a challenge in developing effective natural language processing (NLP) systems. While English is a universal language, there are numerous variations such as Singlish, a creole language spoken in Singapore, which presents unique grammar and syntax. NLP technologies are crucial in enabling billions of people to access and benefit from the internet in their native languages, including lesser-known languages like Singlish. However, existing NLP techniques face challenges in language identification, machine translation, sentiment analysis, POS tagging, and named entity recognition for diverse languages with limited training data.

The first challenge is identifying languages in a multi-lingual resource, as many language names can refer to multiple languages, and thousands of languages may have multiple names. The second challenge is automated language identification, which requires sufficient training data for accurate results. In this project, we aim to address these challenges by using a language detection dataset with various languages and applying a hybrid approach that combines rule-based and machine learning techniques with natural language processing, machine learning, and human input. This approach offers flexibility, scalability, and speed, making it suitable for conversational experiences and task-oriented projects with limited training data. By overcoming these challenges, our project aims to develop an effective language detection system that can facilitate NLP applications for diverse languages and promote inclusive access to digital resources across the globe.

5.1 Workflow



5.2 Pseudocode

1. Begin
2. Import warnings
3. Import libraries
4. Import data
5. Select the required rows
6. Add attributes to all 12 languages
7. Pre-processing the dataset
8. Convert all text to lower case
9. Applying tokenization, sentences are tokenized to words
10. Stemming the words to their root/base form
11. Lemmatizing the data
12. Importing plotting libraries
13. Plot the graph against different attributes
14. Normalize the data
15. Splitting the data into training and test data

16. Divide dataset according to NLP models (unigram, bigram, trigram etc)
17. Train ML models
18. Applying Random Forest Classifier model
19. Applying LinearSVC model
20. Applying Logistic Regression model
21. Compare accuracy of models
22. Predict the language of output given by user.

6. Implementation and Results

6.1 Dataset

Dataset Link: https://downloads.tatoeba.org/exports/sentences.csv This dataset has 3 columns: S.no, Sample, Language. Out of which we only need sample and language. The dataset has around 10million rows and contains data about 150 languages. We have only selected 12 languages which are actively spoken in India. The 12 languages selected are:

- Bengali: ben
- English: eng
- Gujarati: guj
- Hindi: hin
- Kannada: kan
- Malayalam: mal
- Marathi: mar
- Nepali: nep
- Oria: ori
- Punjabi: pan
- Sanskrit: san
- Tamil: tam
- Urdu: urd

6.1.1 Methodology

1. Importing Libraries: The first step in the methodology is to import the necessary libraries for the research. This includes importing the "warnings" library to ignore any warnings generated by Python, as well as importing other libraries such as NumPy, pandas, re, string, math, and nltk.

2. Importing Dataset: The dataset containing data of about 150 languages is imported. This dataset will be used for training and testing the NLP models.

3. Data Processing: The data is processed by removing unnecessary columns that are not needed for the research. Only the rows containing data for the 12 languages of interest are selected.

4. Text Pre-processing: Text pre-processing is performed to prepare the text data for model building. This includes several steps such as removing punctuations using the pre-defined list of punctuations in the string library, converting the text to lower case, tokenization (splitting text into smaller units such as words or sentences), stemming (reducing words to their root/base form), and lemmatization (reducing words to their base form while preserving their meaning).

5. Data Visualization: After pre-processing the data, data visualization techniques such as plotting the data and examining the number of values for each attribute are applied to gain insights into the dataset.

6. Data Normalization: Since the dataset may have an imbalance in the number of samples for each language, data normalization techniques such as RandomOverSampler algorithm are applied to balance the dataset by oversampling the minority class (i.e., the languages with fewer samples) so that all 12 languages have equal representation in the dataset.

7. Dataset Division: The normalized dataset is divided into training and testing datasets according to the requirements of the NLP models to be used for training and evaluation.

8. Feature Extraction: Unigram, bigram, trigram, and up to 10-chargram models are used to extract features from the text data for training the NLP models. All the features obtained from these models are named and used for training the dataset.

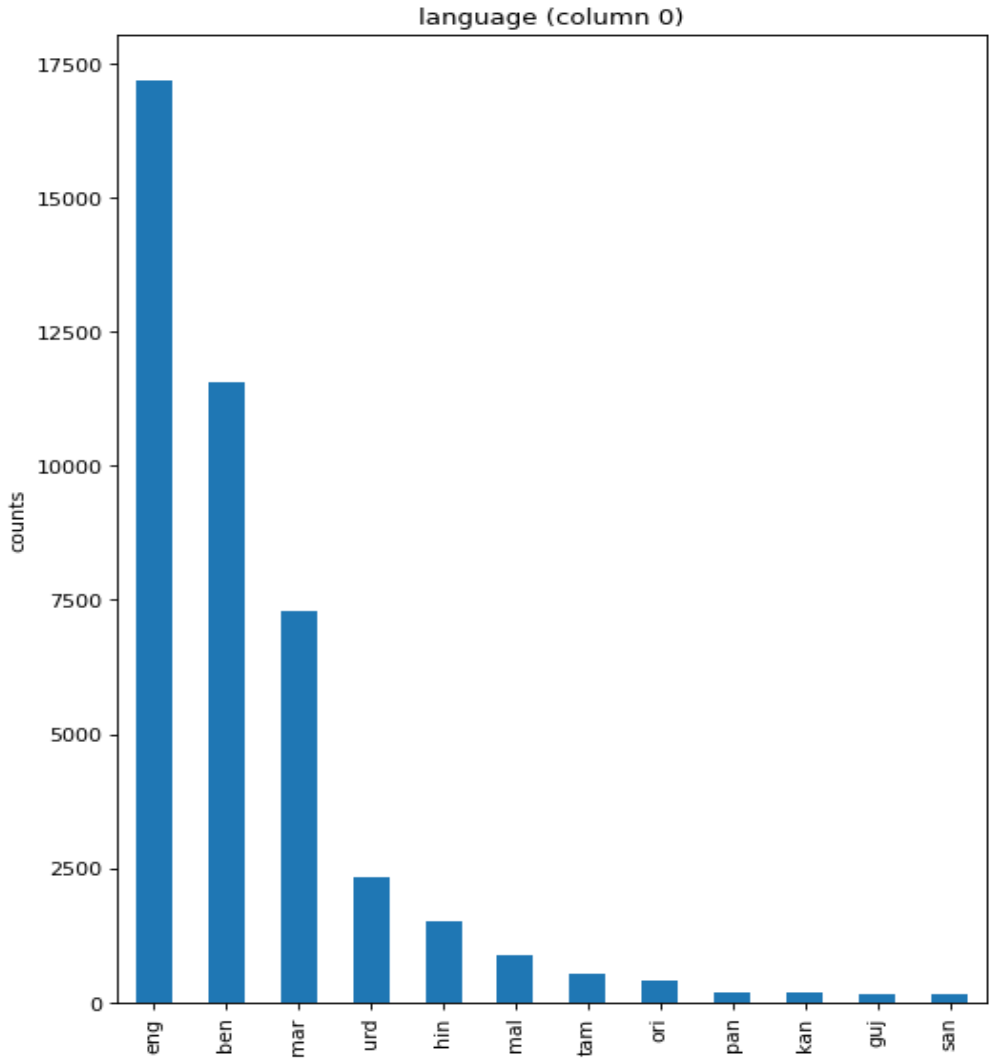
9. Model Training: Three NLP models, namely Random Forest Classifier, Linear SVC, and Logistic Regression, are trained using the training dataset. The Random Forest Classifier is an ensemble method that fits multiple decision tree classifiers on sub-samples of the dataset and uses averaging to improve predictive accuracy. Linear SVC is a support vector classifier that aims to find a hyperplane that best separates the data. Logistic Regression is a supervised classification algorithm that predicts the probability of a data entry belonging to a certain category.

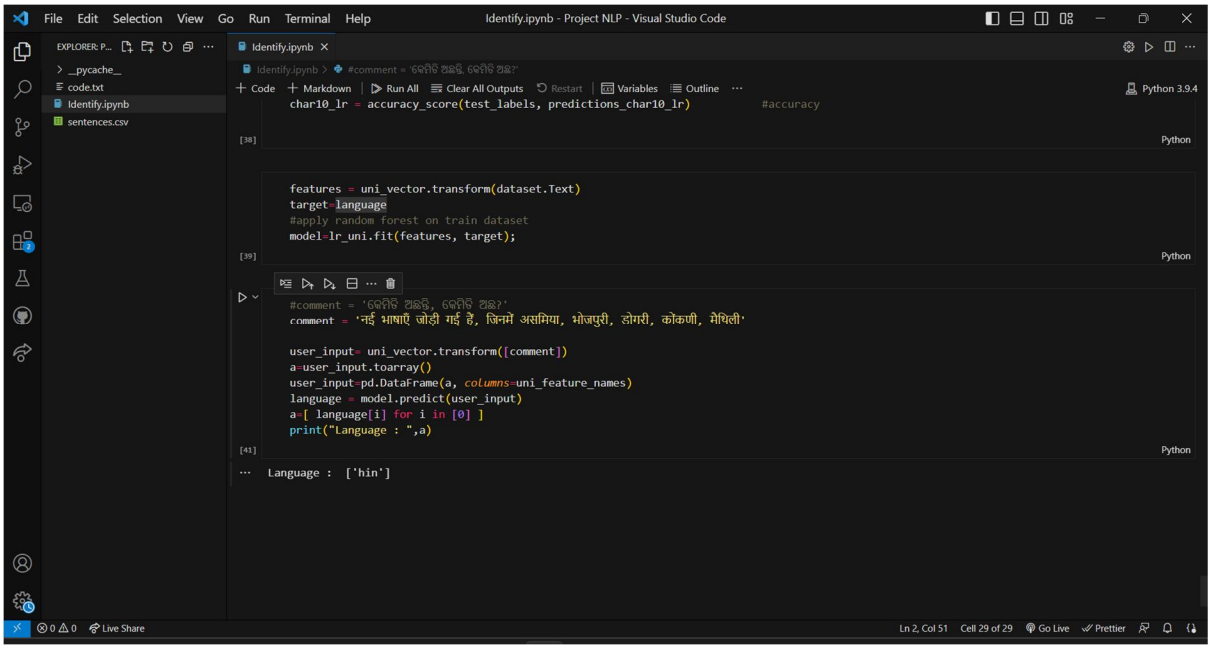
10. Model Evaluation: After training the models, the accuracy of the models is compared using evaluation metrics to determine their performance on the dataset.

11. Prediction: Finally, the user is allowed to enter input text, and the trained models are used to predict the language of the input text. The language code corresponding to the predicted language is displayed as the output.

Overall, the methodology involves data import, data processing, text pre-processing, data visualization, data normalization, dataset division, feature extraction, model training, model evaluation, and prediction to achieve the research objective of language prediction using NLP models.

6.1.2 Output


language (column 0)
counts
17500
15000
12500
10000
7500
5000
2500
0
eng
ben
mar
urd
hin
mal
tam
ori
pan
kan
guj
san


File Edit Selection View Go Run Terminal Help
Identify.ipynb - Project NLP - Visual Studio Code
EXPLORER: P...
__pycache__
code.txt
Identify.ipynb
sentences.csv
Identify.ipynb
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.9.4
char10_lr = accuracy_score(test_labels, predictions_char10_lr) #accuracy
[38]
Python
features = uni_vector.transform(dataset.Text)
target=language
#apply random forest on train dataset
model=lr_uni.fit(features, target);
[39]
Python
#comment = 'ଭେମିତି ଅଛନ୍ତି, ଭେମିତି ଅଛ?'
comment = 'नई भाषाएँ जोड़ी गई हैं, जिनमें असमिया, भोजपुरी, डोगरी, कोंकणी, मैथिली'
user_input= uni_vector.transform([comment])
a=user_input.toarray()
user_input=pd.DataFrame(a, columns=uni_feature_names)
language = model.predict(user_input)
a=[language[i] for i in [0]]
print("Language : ",a)
[41]
Python
Language : ['hin']
Live Share
Ln 2, Col 51
Cell 29 of 29
Go Live
Prettier

7. Conclusion

In conclusion, our research project addressed several challenges in automated language identification when dealing with data from thousands of languages. Despite the difficulties posed by a large number of languages and small sample sizes, we encountered additional issues such as ambiguous language names, incomplete language tables, and incorrect language names and codes. Nevertheless, we successfully processed a dataset containing text from 12 different languages by applying various pre-processing techniques, plotting the data based on language, and training our data using different n-gram models. We also employed three different machine learning algorithms, namely Random Forest Classifier, Linear SVC, and Logistic Regression, to predict the language of input data. However, our model's accuracy is not yet perfect and requires further training with a larger dataset to improve its language detection capabilities. Despite these challenges, our research project provides valuable insights into automated language identification and serves as a foundation for future studies in this field.

8. References

[1] Rachel Mary Milne, Richard A. O'Keefe and Andrew Trotman. A Study in Language Identification. *ADCS '12, December 05-06 2012, Dunedin, New Zealand.*

[2] William. B. Cavnar and John. M. Trenkle. N-gram-based text categorization. *Proceedings of SDAIR-94, 3d Annual Symposium on Document Analysis and Information Retrieval, pages 161-175, 1994.*

[3] Anil Kumar Singh. Study of Some Distance sures for Language and Encoding Identification. *Proceedings of the Workshop on Linguistic Distances, pages 63-72, Sydney, July 2006.*

[4] A. Xafopoulos, C. Kotropoulos, G. Almpanidis and L. Pitas. Language identification in web documents using discrete HMMs. *The Journal Pattern of Recognition Society, Pattern Recognition 37 (2004) 583-594.*

[5] Ali Selamat, Ng Choon Ching and Yoshiki Mikami. Arabic Script Web Documents Language Identification Using Decision Tree-ARTMAP Model. *Proceedings of IEEE Computer Society, 2007 International Conference on Convergence Information Technology.*

[6] Timothy Baldwin and Marco Lui. Language Identification: The Long and the Short of the Matter. Human Language Technologies: *The 2010 Annual Conference of the North American Chapter of the ACL, pages 229-237, Los Angeles, California, June 2010.*

[7] Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink and Theresa Wilson. Language Identification for Creating Language-Specific Twitter Collections. *Proceedings of the 2012 Workshop on Language in Social Media (LSM 2012), pages 65-74, Montreal, Canada, June 7, 2012.*

[8] Bruno Martins and Mário J. Silva, Language Identification in Web Pages. *2005 ACM Symposium on Applied Computing.*

[9] Erik Tromp and Mykola Pechenizkiy. Graph-Based N-gram Language Identification on Short Texts. *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands, 2011.*

[10] S. Ganapathy, K. Han, S. Thomas, M. Omar, M. V. Segbroeck, and S. S. Narayanan, Robust Language Identification Using Convolutional Neural Network Features," *in Proc. of Interspeech, 2014.*

[11] R. Zazo, A. L.-Diez, J. G.-Dominguez, D. T. Toledano, and J. G.-Rodriguez, Language Identification in Short Utterances Using Long Short-Term Memory (LSTM)
Recurrent Neural Networks," PLos ONE, vol. 11(1): e0146917., 2016.

[12] N. Dehak, P. A.T.-Carrasquillo, D. Reynolds, and R. Dehak, Language Recognition via I-vectors and Dimensionality Reduction," *in Proc. of Interspeech, pp. 857–860, 2011.*

[13] N. Cristianini and J. S.-Taylor, Support Vector Machines," *Cambridge University Press, Cambridge, 2000.*

[14] Ranasinghe, T., & Zampieri, M. (2021). An evaluation of multilingual offensive language identification methods for the languages of india. *Information*, *12*(8), 306.

[15] Singh, K., Sen, I., & Kumaraguru, P. (2018, July). Language identification and named entity recognition in hinglish code-mixed tweets. In Proceedings of ACL 2018, Student Research Workshop (pp. 52-58).