Language Detection for YouTube Comments

Rahul T Gonchikar
Information Technology
NITK Surathkal
Surathkal, India, 575025
rahulgonchikar.201it245@nitk.edu.in

Arjun Gowda A G
Information Technology
NITK Surathkal
Surathkal, India, 575025
arjunag.201it210@nitk.edu.in

Hanuma Vamsi Narasingula
Information Technology
NITK Surathkal
Surathkal, India, 575025
hanumavamsinarasingula.201it222@nitk.edu.in

Pranav R S
Information Technology
NITK Surathkal
Surathkal, India, 575025
pranavrs.201it143@nitk.edu.in

***Abstract*—Language detection is an important task in natural language processing that is becoming increasingly important with the rise of global communication and social media platforms. YouTube, being one of the most popular social media platforms, contains vast amounts of user-generated content in various languages. In this research paper, we built a web crawling-based language detection algorithm for determining the language of YouTube comments. The suggested approach can recognize transliterated languages like Hinglish(Hindi written using the English alphabet), Kanglish(Kannada + English) in addition to the comment's original language. To accurately determine the language of the comment, the system uses a combination of machine learning algorithms and natural language processing techniques. The system's language detection accuracy was quite good when tested on a sizable sample of YouTube comments. Potential applications of this research include sentiment analysis and social media monitoring, both of which call for multilingual data analysis.**

***Index Terms*—Web crawling, Transliteration, Classification, Machine Learning**

I. Introduction

Language detection is an important task in natural language processing that involves automatically identifying the language of a given text. Language detection has numerous applications, including text classification, information retrieval, and machine translation. In recent years, social media platforms have become an important source of user-generated content in various languages, which makes accurate language detection critical for social media monitoring and sentiment analysis.

Hinglish, Kanglish, and Telglish are examples of transliterated languages, where words and phrases are written using the English alphabet, but the language being spoken is not English. Hinglish is a combination of Hindi and English, Kanglish is a combination of Kannada and English, and Telglish is a combination of Telugu and English.These languages are commonly used in social media platforms, such as YouTube, where users express their opinions and thoughts in their native language, which may not be in English. In this research, we focus on detecting these languages to better understand user behavior and sentiment on social media platforms.

To train our language detection system, we collected data from various sources. We used the Google Research Dataset for Hinglish, which contains a large collection of Hinglish text. For Kanglish and Telglish, we created our dataset using the Microsoft Translate API and indic_transliteration libraries. The dataset was labeled manually to ensure the accuracy of the language detection system..

Our proposed language detection system uses a combination of feature engineering, supervised learning etc. We used supervised learning algorithms, such as Naive Bayes, logistic regression, and support vector machines, to train the language detection model techniques to accurately detect the language of the text. The system achieved high accuracy in detecting Hinglish, Kanglish, and Telglish, as well as other languages commonly used in social media platforms.

II. Dataset

For the language detection of transliterated language and code-mixed language, we have generated our own dataset containing two columns, first column for text and second column is the language of the text. Dataset has English, Hindi, Kannada, Telugu, Tamil and Malayalam sentences,all written in English, and are correctly labelled as the language it belongs to.

English and Hinglish(Code-mix of Hindi and English) sentences are labelled as English and Hindi respectively,these sentences are taken from a well-known Hinglish-TOP-Dataset dataset by Google Research Datasets.

Code-mixing of English and South Indian languages such as Kannada, Telugu, Tamil and Malayalam are generated by translating English sentences of the above mentioned dataset to the required language and transliterating it back to English.This produces code-mixed language such as Kanglish(Kannada+English) which is labelled as Kannada.

Translate is a translation tool written in Python with support of multiple translation providers.This tool was used to translate English sentences to required South Indian language.The

generated South Indian language was transliterated to English using indic_transliteration which is a transliteration tool written in Python.

The dataset contains text written in English script(Latin).There are 2993 sentences for each language label of English, Hindi, Kannada, Telugu, Tamil and Malayalam. The dimension of the dataset is (2993*6) X 2. Each of the 6 languages have 2993 sentences with a language label.

III. LITERATURE SURVEY

[1] A Systematic Review on Language Identification of Code-Mixed Text: Techniques, Data Availability, Challenges, and Framework Development, Ahmad Fathan Hidayatullah, Atika Qazi, Daphne Teck Ching Lai,AND Rosyzie Anna Apong.

The paper discusses the challenges and techniques associated with identifying languages in code-mixed text. The authors provide an overview of the prevalence of code-mixing in multilingual societies and the difficulties in standardizing language identification due to the presence of multiple languages and dialects in the same sentence. The paper reviews various techniques used for language identification, including rule-based approaches, statistical methods, and machine learning techniques. Additionally, publicly available datasets for research purposes are analyzed, and the authors provide insights into their challenges, such as their small size and lack of diversity. The authors propose a new framework for language identification in code-mixed text that combines rule-based and machine learning techniques.

In conclusion, the paper provides a comprehensive overview of the existing literature on language identification in code-mixed text. The authors highlight the importance of addressing the challenges of language identification in code-mixed text and propose a new framework that can improve identification accuracy. This paper provides valuable insights for researchers and practitioners working in the field of natural language processing, and the proposed framework can have practical applications in various fields, such as social media monitoring and customer service.

[2] ”Sentiment Analysis in Dravidian Code-Mixed YouTube Comments and Posts”, Sanjeepan Sivapiran, Charangan Vasantharajan and Uthayasanker Thayasivam Department of Computer Science and Engineering, University of Moratuwa

The paper presents a comprehensive analysis of sentiment analysis techniques on code-mixed text. The authors, Sanjeepan Sivapiran, Charangan Vasantharajan, and Uthayasanker Thayasivam, belong to the Department of Computer Science and Engineering at the University of Moratuwa. The paper addresses the challenges associated with analyzing the sentiment of code-mixed text, where two or more languages are mixed in a single text, and provides insights into the effectiveness of different approaches in the context of Dravidian languages.

The authors conduct an extensive literature review of existing research on sentiment analysis in code-mixed text, including approaches such as machine learning, rule-based, and hybrid techniques. They also review previous studies on sentiment analysis in Dravidian languages and code-mixed text in other languages. The authors then propose a new approach based on a deep learning architecture for sentiment analysis in Dravidian code-mixed YouTube comments and posts. The proposed approach combines convolutional neural networks and bidirectional long short-term memory networks, which outperforms other existing techniques in terms of accuracy, precision, recall, and F1 score. The results of this study can be useful for researchers and practitioners in developing better sentiment analysis models for code-mixed text in other languages as well.

Overall, this paper provides a valuable contribution to the field of sentiment analysis in code-mixed text by presenting a novel approach to sentiment analysis in Dravidian code-mixed YouTube comments and posts. The authors provide an extensive literature review of existing research and compare their approach's performance to other existing techniques, demonstrating its superiority in accuracy and other metrics. This paper can be useful for researchers and practitioners interested in sentiment analysis and code-mixed text, particularly in the context of Dravidian languages.

[3] Cross-lingual Language Model Pretraining by Guillaume Lample, Alexis Conneau

The paper titled ”Cross-lingual Language Model Pretraining” by Guillaume Lample and Alexis Conneau is a seminal work in the field of cross-lingual natural language processing. The authors propose a method for training a single neural network model that can perform well on multiple languages without requiring labeled data for each language. The proposed method leverages unsupervised pretraining and transfer learning techniques to learn a shared representation of different languages.

The paper begins with a comprehensive literature review of previous work in cross-lingual language modeling, including methods based on parallel corpora, bilingual dictionaries, and multilingual embeddings. The authors then introduce a new approach, called Cross-lingual Language Model Pretraining (XLM), that extends the success of monolingual language modeling to a cross-lingual setting. They demonstrate the effectiveness of XLM on several cross-lingual benchmark datasets and compare its performance to other state-of-the-art methods.

In conclusion, this paper provides a valuable contribution to the field of cross-lingual natural language processing by proposing a novel approach for unsupervised pretraining of a neural network model that can perform well on multiple languages. The authors provide a comprehensive literature review of previous work in the field and demonstrate the effectiveness of their proposed approach on several benchmark datasets.

TABLE I
LITERATURE SURVEY

Number	Title	Authors	Methodology	Merits
1	A Systematic Review on Language Identification of Code-Mixed Text: Techniques, Data Availability, Challenges, and Framework Development	Ahmad Fathan Hidayatullah, Atika Qazi, Daphne Teck Ching Lai,AND Rosyzie Anna Apong	The paper presents a systematic review of existing research on language identification of code-mixed text, including techniques, data availability, challenges, and framework development. The authors conduct an extensive literature review of previous work and provide insights into the effectiveness of different approaches.	Provides a comprehensive analysis of language identification techniques for code-mixed text. The authors identify research gaps and propose a framework for future research.
2	Sentiment Analysis in Dravidian Code-Mixed YouTube Comments and Posts	Sanjeepan Sivapiran, Charangan Vasantharajan, and Uthayasanker Thayasivam	The paper proposes a novel deep learning approach for sentiment analysis in Dravidian code-mixed YouTube comments and posts. The authors conduct an extensive literature review of existing research on sentiment analysis in code-mixed text and compare their approach's performance to other existing techniques.	Presents a new approach for sentiment analysis in code-mixed text that outperforms existing techniques in terms of accuracy, precision, recall, and F1 score. The authors provide insights into the effectiveness of different sentiment analysis techniques in the context of Dravidian languages.
3	Cross-lingual Language Model Pretraining	Guillaume Lample and Alexis Conneau	The paper proposes a novel approach for cross-lingual language modeling called Cross-lingual Language Model Pretraining (XLM), which leverages unsupervised pretraining and transfer learning techniques. The authors provide a comprehensive literature review of previous work in cross-lingual language modeling and demonstrate the effectiveness of XLM on several benchmark datasets.	The proposed approach allows for the training of a single neural network model that can perform well on multiple languages without requiring labeled data for each language. The authors demonstrate the effectiveness of XLM on several cross-lingual benchmark datasets and compare its performance to other state-of-the-art methods.

IV. PROBLEM STATEMENT

The problem statement is to develop a model that can accurately detect the language of the crawled dataset of YouTube comments, the model should also be able to recognize the regional languages as well. This requires a machine learning model that can identify the characteristics of the language, as well as differentiate it from other languages. The model should be able to classify the comments based on the language accurately. The development of such a model can be useful for content creators to understand the audience engagement and to make decisions regarding their content strategy.

V. METHODOLOGY

We have built a Language Detection model for any given text. The first phase of the model is Script Detection, where the script of the text is identified. Here all the languages which are written in Non-Latin script are identified. The text classified as Latin can be English or a code mix of English and any other language. We then train multiple models with different classification algorithms for the above dataset to classify the Latin text as English or the language it is code mixed with. Now we obtain the comments of a specified video using the YouTube v3 API. We pass these comments to the above model to identify the language for each of the comments.

A. Script Detection

We detect the script of the text by checking which UTF-8 range the characters in the text lie in. We start by creating an empty dictionary ”store” to store the frequency of each script detected in the input string. It then defines a set ”c” containing the Unicode categories for control characters, format characters, surrogate code points, private use characters, and unassigned characters. These categories are excluded from the script detection process, as they do not contain information about the script of the text.

We then iterate through each character in the input string and checks whether its Unicode category is not in the set ”c”. If the category is not in the set, we extract the character's Unicode name and add it to the ”store” dictionary as a key, with a value of 1 if the key does not exist, or increments the value by 1 if it does. Finally, we check whether the ”store” dictionary is empty. If it is not empty, the function returns the key with the highest value (i.e., the script that occurs most frequently). If the ”store” dictionary is empty, we return an empty string.

B. Language Detection for text written in Latin Script

In the first phase of Language Detection, the languages for all the texts written in their original script is detected. We now have to detect the languages for the texts written in Latin script. Texts in Latin script may be purely English or any language code mixed with English.

For this, we train multiple models which use different algorithms using the above dataset. The dataset contains text in different languages and their labels. We preprocess the data to clean and normalize the text. We split the data into training and testing sets, extracting numerical features from the preprocessed text data using a CountVectorizer. We then train machine learning models on the extracted features. These models will help us to identify the language written in Latin script.

C. Fetching YouTube Comments

First, to fetch the comments from a video, we have to set up the API service name, version, and developer key. The Google API client library is a Python library that allows developers to interact with Google APIs. It provides a simple way to access Google APIs such as YouTube, Google Drive, and Google Maps. The library handles many of the low-level details of making requests and handling responses from Google APIs. Then, we specify the video ID for which comments are to be retrieved. We then use a while loop to retrieve all comments from the video by making requests to the YouTube API using the nextPageToken. The comments are stored in a dictionary with comment IDs as keys and comment text as values. Finally, we convert the dictionary into a pandas DataFrame with two columns: ID and Comment.

We pass these comments to the earlier trained model to detect the language used.

VI. Results

We can build classification models based on different algorithms. Here are the classification models we have tried on our dataset and their respective accuracies:

1) Multinomial NB: 99.05%
2) Logistic Regression: 99.44%
3) Decision Tree: 98.71%
4) Random Forests: 99.44%
5) Support Vector Machines (SVM): 99.27%
6) K-Nearest Neighbors (KNN): 97.43%
7) Gradient Boosting: 96.54%
8) MLP Classifier: 99.61%

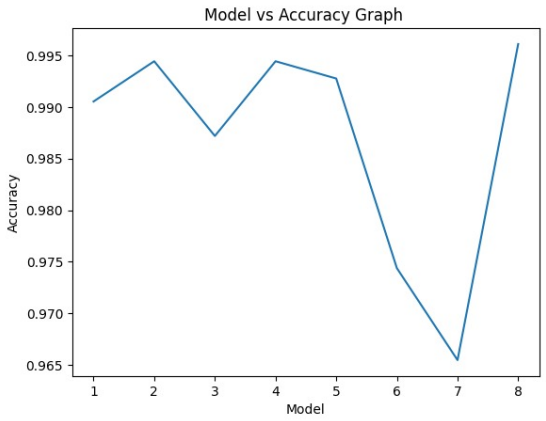

Fig. 1. Model vs Accuracy Graph

We observe that the MLP Classifier model has the highest accuracy and thus we will use it in our implementation.

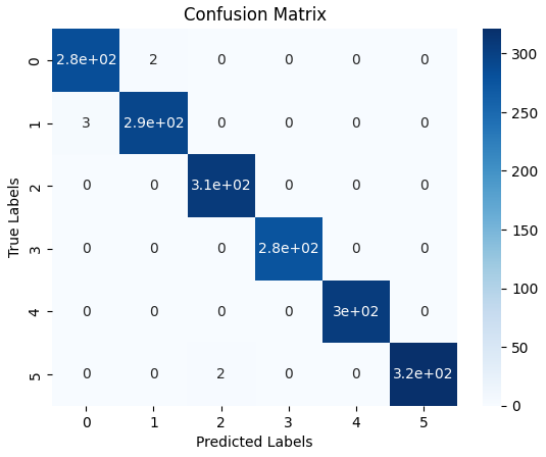

Fig. 2. Confusion Matrix of MLP Classifier

The comments which are fetched by using web are passed through to the and identify the language used, this model even detects the transliterated model.

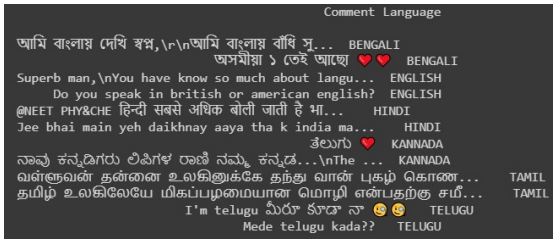

Fig. 3. Language of comments detected

VII. Conclusion

We have developed a language detection system specifically designed for Youtube comments. Our approach has the capability to accurately identify transliterated languages such as Hinglish (Hindi written using the English alphabet), Kanglish (a mixture of Kannada and English), Telgish, etc in addition to detecting the comment's original language. After extensive testing, we have determined that the Multilayer Perceptron (MLP) classification model provided the highest level of accuracy, and therefore, we have chosen to implement it in our approach.

References

[1] Ahmad Fathan Hidayatullah, Atika Qazi, Daphne Teck Ching Lai, and Rosyzie Anna Apong, *"A Systematic Review on Language Identification of Code-Mixed Text: Techniques, Data Availability, Challenges, and Framework Development"*.

[2] Sanjeepan Sivapiran, Charangan Vasantharajan, and Uthayasanker Thayasivam *"Sentiment Analysis in Dravidian Code-Mixed YouTube Comments and Posts"*.

[3] Cross-lingual Language Model Pretraining*"Guillaume Lample and Alexis Conneau"*.

[4] YouTube v3 API *https://developers.google.com/youtube/v3*.

[5] Translate *https://pypi.org/project/translate/*.

[6] Indic-Transliteration *https://pypi.org/project/indic-transliteration/*.