POSACLE: Phonetic Script Systems: A Comparative Linguistic Evaluation

[1]Dhairvi Shah, [2]Preeti Kale
[1]dhairvi.shah@mitwpu.edu.in , [2]preeti.kale@mitwpu.edu.in
[1, 2] Department of Computer Engineering & Technology, Dr. Vishwanath Karad MIT-World Peace University, Pune, India

Abstract

*This research, under the framework of **POSACLE (Phonetic Script Systems: A Comparative Linguistic Evaluation)**, examines phonetic similarities across various languages, notably centering on the Dravidian and Devanagari scripts. The framework comprises three fundamental elements: (1) linguistic categorization; (2) character-by-character resemblance evaluation; and (3) comprehensive language similarity appraisal. The research attained classification accuracies of 98.57% for the Gujarati-Marathi/Hindi language pair and 96.49% for the Tamil-Kannada language pair by employing the techniques of Random Forest and Gradient Boosting. Through cosine similarity analyses, **POSACLE** identified patterns and intricate similarities at the letter level, with an average value of 0.0204 observed across all language pairs. The examination of phonetic similarity brought to light significant associations, notably between Marathi-Hindi of about 0.69 and Tamil-Kannada of about 0.11. These outcomes highlight the strides taken in multilingual optical character recognition (OCR), transliteration, and translation systems. The implications of this study, spearheaded by the **POSACLE** framework, extend to various applications such as speech recognition, language acquisition, and linguistic studies, showcasing its extensive technological and interdisciplinary impact. This research establishes a standard for processing multilingual text and images, providing valuable insights into the structures and connections within languages.*

Keywords: *Machine Learning, Natural Language Processing, Indian Languages, Polyglot Linguistics, Deep Learning and Linguistic Features.*

1. Introduction:

Linguistic landscapes of India reflect the rich and diverse history of the Indian subcontinent, developed over millennia of cultural, historical, and social exchanges. India's languages represent the intricate network of civilizations, migrations, and cultural exchanges that have shaped the country over time, from the oldest writings on the seals of the Indus Valley Civilization to the varied linguistic environment saw during the colonial era. This large and varied subcontinent's linguistic legacy has been molded by a variety of evolutionary routes, dispersion patterns, and transformational processes, all of which may be explored thanks to the dense language tapestry.

The Indo-Aryan and Dravidian languages, the two main linguistic groups that define India's linguistic diversity from which the northern and northwest areas are dominated by the Indo-Aryan languages where as the southern regions of India are home to the majority of speakers of the Dravidian languages. The languages of the Tibeto-Burman and Indo-Australian families also contribute to the richness of the linguistic environment, adding to its complexity and providing a wider field of study for linguistic research [18].

This study explores the overall similarities among the varied Indian subcontinent's languages, concentrating on the Dravidian and Devnagri scripts like Hindi, Gujarati, Marathi as one group whil Tamil and Kanada as another. The study covers almost whole of the domain, starting from linguistic classification to finding relations and similarity among the languages. We achieved good classification accuracy by using sophisticated techniques like Random Forest and Gradient Boosting, highlighting the resilience of these models in differentiating apart similar-looking characters from various languages. This accomplishment is essential for creating Optical Character Recognition systems that are more dependable and precise in multilingual settings.

The meticulous analysis of letter-wise phonetic similarity among the group of languages considered elucidates the degree of resemblance between characters across different languages. This insight is instrumental in the

advancement of more efficient transliteration and translation systems, thereby enhancing the accuracy and efficacy of cross-language communication tools. Furthermore, the application of sophisticated machine learning models for text classification across languages demonstrates their efficacy in handling complex linguistic data. The high accuracy rates achieved indicate the potential for these models to be effectively utilized in similar tasks within other multilingual contexts.

The implications of our findings are far-reaching, impacting various domains such as multilingual OCR, transliteration, and translation systems. These advancements enhance speech recognition technologies, facilitate language acquisition, and contribute to broader linguistic studies, thereby underscoring the technological and interdisciplinary relevance of our research. In addition to offering a strong foundation for handling multilingual text and graphics, this work sheds light on the phonetic and structural links both inside and across languages.

The subsequent section offers an exhaustive analysis of the extant **literature** pertinent to this study, thereby laying the groundwork upon which this research endeavor is predicated. **Section 3** delivers a meticulous scrutiny of the methodological framework utilized in this inquiry, categorized into three primary elements:

(1) Classification methodologies,
(2) Letter-wise similarity evaluation,
(3) Phonetic correspondence analysis across diverse languages

Section 4 subsequently articulates the findings derived from the application of these methodological approaches. The paper culminates with a discourse on the implications of the findings and their broader significance, succeeded by a synthesis and final conclusions.

2. Literature Review

This review examines a wide range of research that investigate many aspects of Indian language processing, from emotion evaluation and hate speech detection to voice recognition and machine translation. These studies highlight how language technology is developing in India and highlight its importance for promoting communication, protecting cultural heritage, and overcoming linguistic gaps within the country. The Literature Review conducted can be represented in table 1 as given below:

Research Area	Applications	Techniques	Languages	Key Papers
Hate Speech Detection	Offensive language detection, surveys	Transformer-based embedding, ML, DL, TL, feature extraction	Kannada, Malayalam, Tamil, Hindi, Dravidian languages	[1], [3]
Unified Automatic Speech Recognition	Multilingual speech recognition	Common labeling scheme, multilingual modeling	Kannada, Telugu, Sanskrit, Malayalam, Tulu	[2], [11]
Machine Translation	Neural machine translation	mT5 Transformer, AdamW optimizer	Hindi, Bengali, English, Bhojpuri, Sindhi, Marathi, Urdu	[4], [17]
Text Translation	Indian language translation	LSTM, BLEU score evaluation	Hindi, Odia, Malayalam, Indian Sign Language	[5]
Anomalous Text Identification	Anomaly detection in text	Markov Chain models	Indus Script	[6], [16]
Speech Synthesis	Multilingual text-to-speech synthesis	Multilingual training, byte-pair encoding	Indo-Aryan and Dravidian languages	[7], [14], [15]
Real-Time Translation	Real-time language translation	LSTM, deep learning	Marathi to Gujarati	[8], [13]
POS Tagging	Linguistic analysis for low-resource languages	Rule-based, HMM tagging	Marathi, Assamese, Bengali, Telugu, Konkani,	[10]

Research Area	Applications	Techniques	Languages	Key Papers
			Manipuri	
Low-Resource Speech Recognition	Speech recognition in low-resource settings	Encoder-Decoder Transformer	Gujarati, Tamil, Telugu	[9], [12]

Table 1: Literature Review

The studies presented delve into language processing and translation technologies tailored to India's diverse linguistic landscape. From advanced digit recognition techniques to machine translation systems, the research aims to overcome language challenges. Through artificial neural networks, linguistic structures, and deep learning, efforts bridge linguistic gaps, facilitate communication, and enable information dissemination.

Moreover, the studies address specific needs like sentiment analysis, hate speech detection, and speech recognition in low-resource languages, leveraging machine learning to enhance understanding and promote inclusivity. Overall, the research reflects a concerted effort to empower India's linguistic diversity and foster seamless interaction among its people.

3. Methodology

The **POSACLE structure**, as illustrated in Figure 1.1, provides an overview of the comprehensive methodology employed in this study. This figure offers a structured insight into each methodological phase, outlining the key steps and processes that underpin the research framework.

.

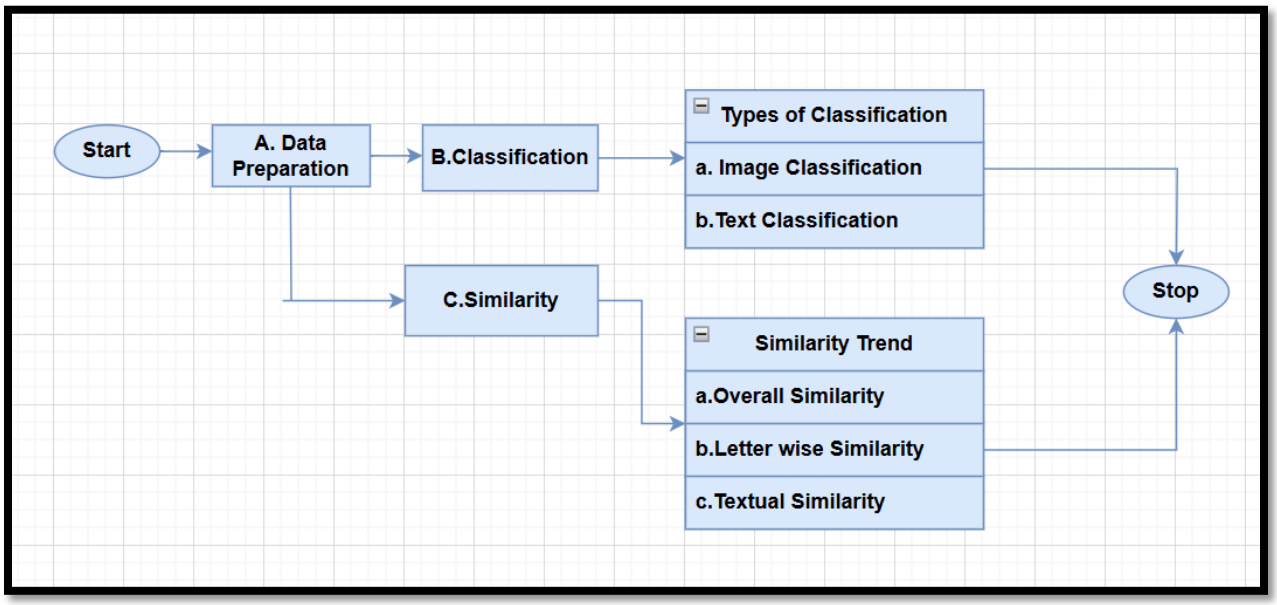

Figure 1.1: POSACLE System

As depicted in above Figure 1.1, flowchart delineates a procedure initiating with data preparation, subsequently categorized into two distinct classifications: image classification and text classification. Following the classification phase, a similarity analysis is conducted, encompassing the evaluation of overall similarity, letter-wise similarity, textual similarity, and the observation of similarity trends. The results derived from both the classification and the similarity analysis culminates in the conclusion of the process.

A. Dataset Peparation

The dataset utilized in this study, which implements the POSACLE system, comprises two primary types: images and text. A detailed description of the data processing workflow is as follows:

I. Image Dataset

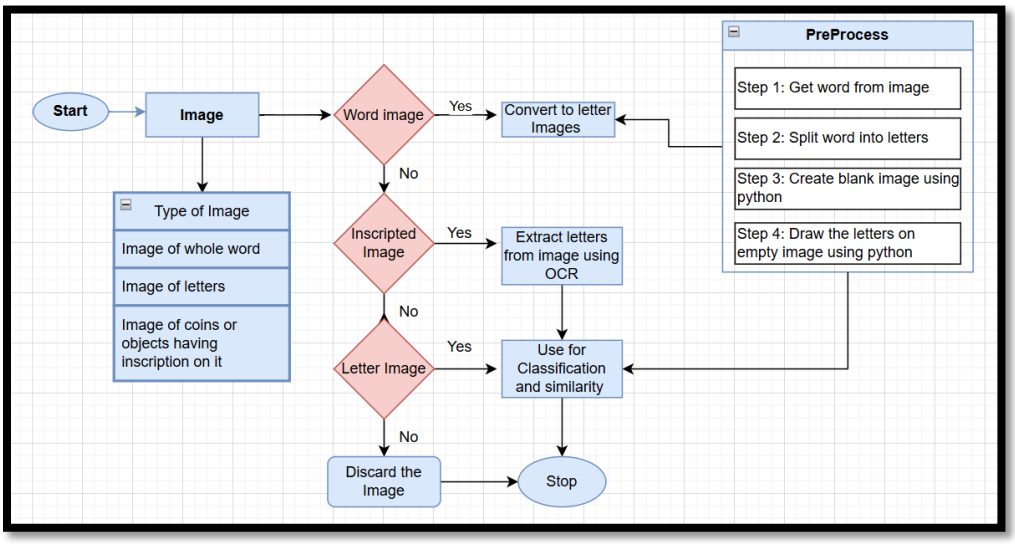

Figure 1.2 Image Dataset Prepration Process

As shown in above Figure 1.2 Images are categorized into three types: whole-word images, letter images, and images with inscriptions (e.g., coins or objects). Word images are split into individual letters, while inscribed images undergo OCR-based letter extraction. Letter images are directly used for classification. Irrelevant images are discarded. Preprocessing involves extracting words, splitting them into letters, and creating blank images to visualize letters using Python, ensuring consistent input for classification and Similarity checks tasks. This systematic approach enhances the efficiency and accuracy of linguistic analysis.

II. Textual Dataset

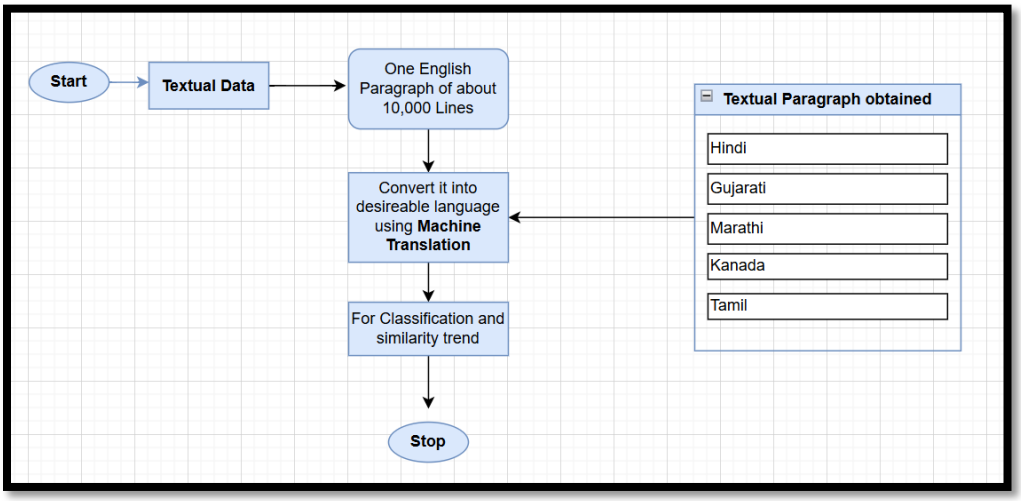

Figure 1.3 Textual Data Preparation

The Figure 1.3 illustrates the preparation of textual data. The process begins with an English textual dataset comprising approximately 10,000 lines. This data is translated into target languages—Hindi, Gujarati, Marathi, Kannada, and Tamil—using machine translation tools. The translated paragraphs are then utilized for classification and to identify trends in linguistic similarity across languages. This systematic approach ensures a robust foundation for phonetic and structural analysis within multilingual datasets.

The integration of these two dataset types allowed for a comprehensive analysis, combining visual and linguistic elements to uncover intricate phonetic and structural similarities between languages. The step-by-step process ensured that each dataset was optimally prepared to align with the objectives of the POSACLE system.

This dual approach not only strengthened the study's findings but also laid a robust foundation for future research in multilingual computational linguistics.

B. Classification

a. Image Classification

The initial phase involves the development and training of a model designed to differentiate between various languages based on visual representations of their text. This Image dataset is letter images as described in data preparation. The overall process can be seen in the following Figure 1.4.

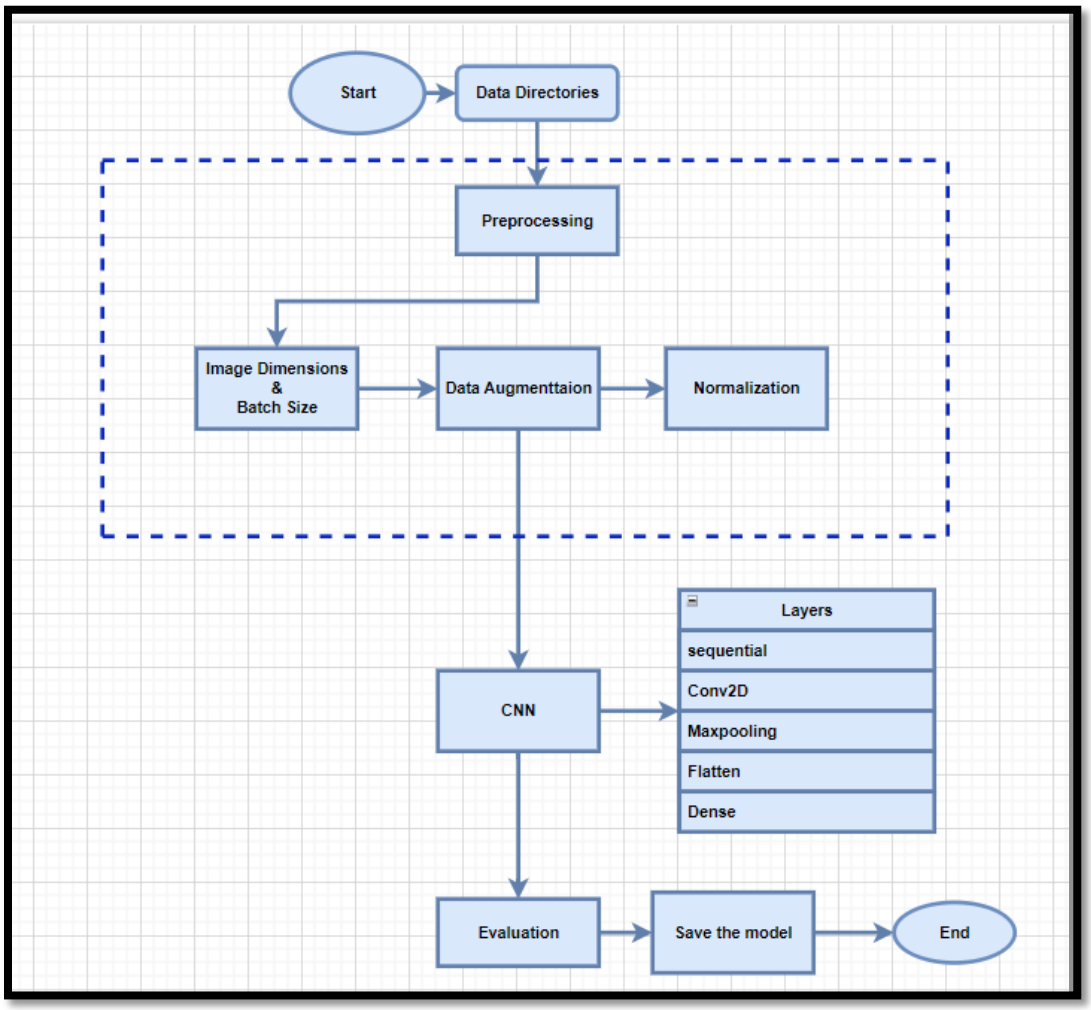

Figure 1.4: Flowchart for Image Classification

- **Data Organization**: Begin by arranging data directories systematically for each language.
- **Image Preprocessing**:
  - Standardize image dimensions and batch sizes.
  - Apply advanced data augmentation techniques to increase variability and robustness of the dataset.

- **Image Normalization**: Perform normalization to ensure uniformity across input data, facilitating consistent model performance.
- **CNN Architecture Design**:
    - Construct a CNN model, integrating layers such as Conv2D, MaxPooling, Flatten, and Dense.
    - Carefully design each layer to optimize feature extraction and model learning capacity.
- **Model Training and Evaluation**:
    - Train the CNN on the preprocessed and normalized dataset.
    - Evaluate model performance metrics to gauge effectiveness and identify any necessary adjustments.
- **Model Saving**: Once performance metrics are satisfactory, save the trained model for deployment in language identification tasks.

This structured approach ensures a comprehensive methodology for training a CNN to identify l anguages through image analysis.

b. <u>Textual Classification</u>

Following the successful classification of images, the study progresses to text-based classification prepared in data preprocessing. The architecture for this phase, illustrated in Figure 1.5, details the framework employed for processing and categorizing textual inputs.

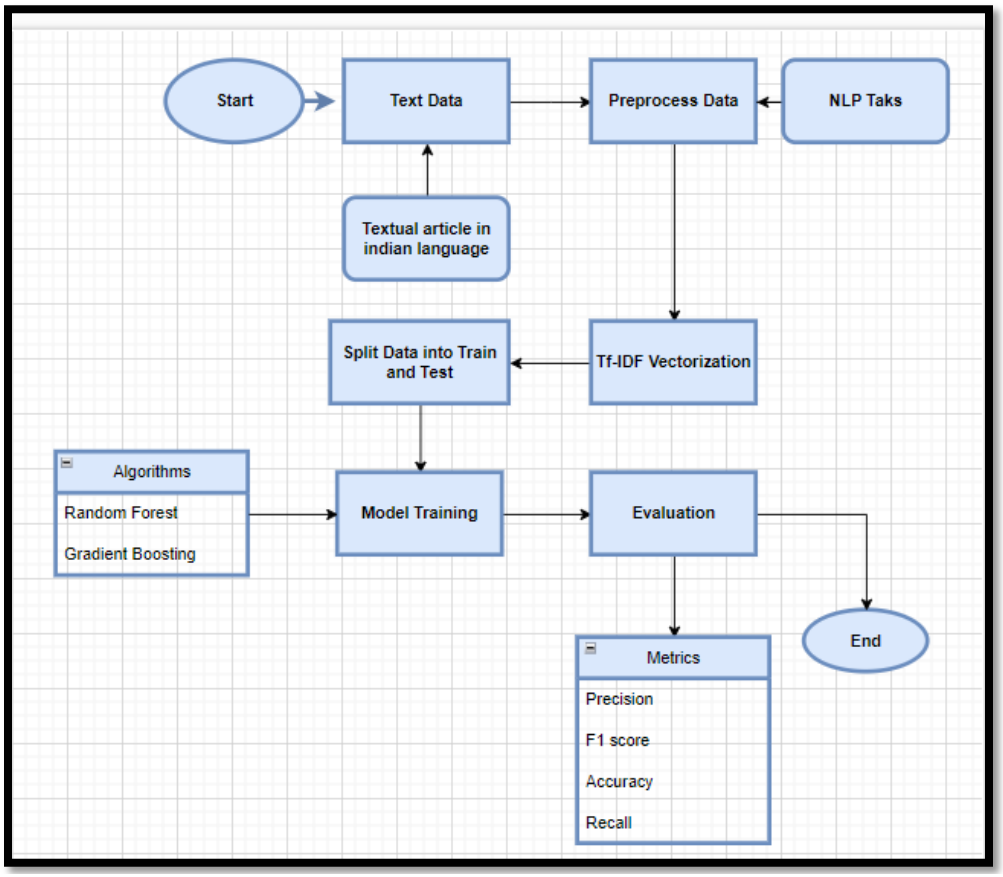

Figure: 1.5 Flowchart for Text data Classification

The above flow structure can be explained as:

- **Start**: The process begins here.
- **Text Data**: The initial input is raw text data, which could be textual articles in Indian languages.

- **Preprocess Data:** This step involves cleaning and preparing the text data for analysis. Common preprocessing tasks include tokenization, removing stop words, and stemming or lemmatization.
- **Split Data into Train and Test**: The pre-processed data is divided into two sets: training data for building the model and test data for evaluating its performance.
- **Tf-IDF Vectorization**: The text data is transformed into numerical features using Term Frequency-Inverse Document Frequency (Tf-IDF) vectorization, which helps in quantifying the importance of words in the documents.
- **Algorithms**: Different machine learning algorithms are employed for model training. In this flowchart, two algorithms are highlighted:
    - Random Forest
    - Gradient Boosting
- **Model Training:** The selected algorithms are used to train the model on the training data.
- **Evaluation**: The trained model is evaluated using the test data. Various metrics are calculated to assess the model's performance, including:
    - Precision
    - F1 Score
    - Accuracy
    - Recall
- **End**: The process concludes here.

This flowchart provides a systematic approach to handling natural language processing (NLP) tasks, from raw text data to model evaluation.

C. Similarity Trend

The similarity trend analysis in this study concentrates solely on the written script of languages, disregarding pronunciation or spoken variations among native speakers. This analysis is conducted through two distinct approaches:

1. **Overall Trend**: Just considering overall language
2. **Letter-wise Similarity**: Evaluates the structural and visual similarity as well as phoenetic resemblance of individual letters across languages, identifying shared or differing characteristics in their written forms.
3. **Textual Similarity**: Comparing languages on the basis of textual paragraph belonging to each language respectively.

These two approaches collectively provide insights into script-based similarities and differences among languages, contributing to a deeper understanding of written language characteristics.

a. Overall Trend

This flowchart delineates a comprehensive methodology for analyzing image data to differentiate between languages. The process involves loading and preprocessing data, extracting relevant features, and transforming images for consistency. Subsequent steps include random sampling, computing a cosine similarity matrix, and visualizing the results with a heat map to reveal patterns and distinctions among the language sets.

The Figure 1.6 given below represents a systematic process for analyzing image data to distinguish between different languages. The procedure is described in the following detailed steps:

- **Start**: The process begins.
- **Load & Preprocess Data**: The first step involves loading the dataset and performing preliminary preprocessing. This includes operations such as normalization and cleaning to prepare the data for further analysis.
- **Extraction of Features**: The extraction of features is performed on the preprocessed data in order to improve the efficiency of the classification process. This step aims to identify and isolate important characteristics within the images that can help in differentiating between languages.

- **Transformation to Grayscale & Resizing**: To minimize complexity and computational burden, each image within the dataset undergoes a conversion to grayscale. Subsequently, the images are resized to a uniform dimension, ensuring consistency across the dataset.
- **Flatten Images**: The resized grayscale images are then flattened into one-dimensional arrays. This transformation allows for easier manipulation and analysis in subsequent steps.

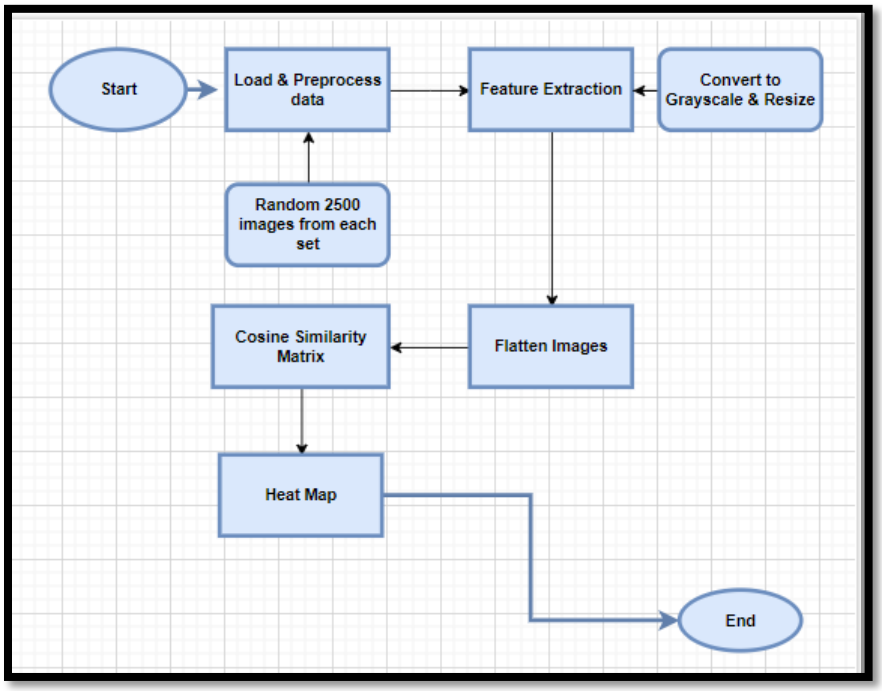

Figure 1.6: Flowchart for Overall Similarity Trend

- **Random 2500 Images from Each Set**: From each language set, 2500 images are randomly selected. This random sampling helps in managing computational resources and ensuring that the analysis is representative of the entire dataset.
- **Cosine Similarity Matrix**: A cosine similarity matrix is computed for the selected images. This matrix is utilized to quantify the similarity between pairs of images through the evaluation of the cosine of the angle formed by their respective feature vectors. High similarity values indicate that the images are more alike in terms of their features.
- **Heat Map**: The cosine similarity matrix is visualized using a heat map. This graphical representation highlights patterns and clusters within the data, indicating how similar or dissimilar the images are to each other.
- **End**: The process concludes.

The entire workflow is designed to systematically process and analyze images to effectively distinguish between different languages based on their visual features.

b. <u>Letter Wise Similarity</u>

This flowchart outlines a detailed methodology for analyzing and visualizing the similarities between alphabetic characters from different languages. These method collectively provides a comprehensive understanding of the relationships and similarities between different alphabetic systems.

The Figure 1.7 depicted below presents a detailed process for analyzing and visualizing the similarity between alphabetic characters from different languages. The process involves multiple stages, each focusing on specific analytical and computational tasks, and is described as follows:

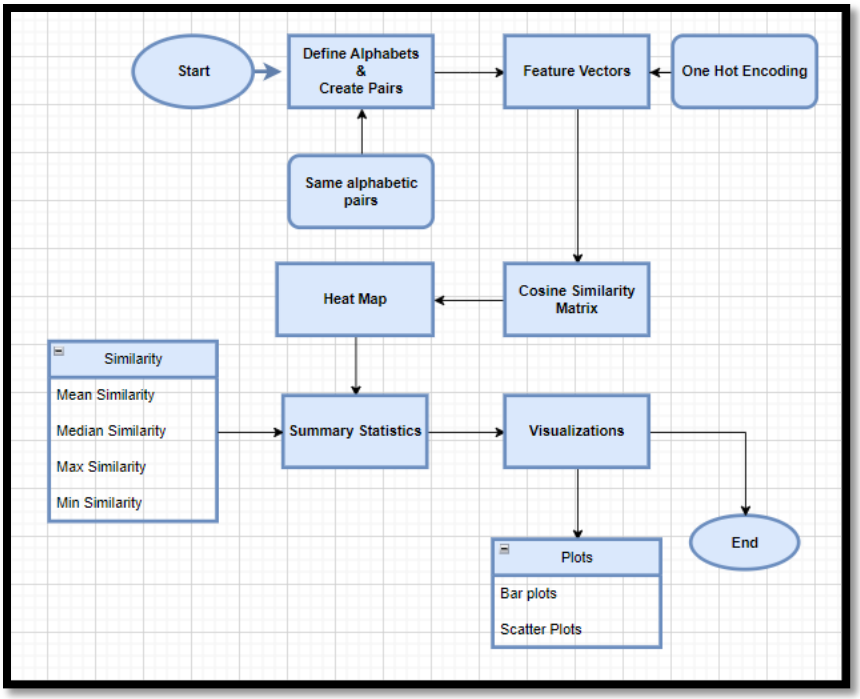

Figure 1.7 : Flowchart showing letter wise similarity analysis

- **Start**: The process is initiated.
- **Define Alphabets & Create Pairs**: Alphabets from different languages are defined, and pairs of characters are created. These pairs include both similar and distinct alphabetic characters to facilitate comparative analysis.
- **Feature Vectors**: For each character pair, feature vectors are generated. These vectors represent the characters in a high-dimensional space, capturing essential characteristics for comparison.
- **One-Hot Encoding**: One-hot encoding is a technique that is used to change the feature vectors. It transforms categorical information into a vector of binary values . This ensures that the features are in a suitable format for subsequent similarity analysis.
- **Cosine Similarity Matrix**: A cosine similarity matrix is computed for the feature vectors. This matrix measures the cosine of the angle between pairs of vectors, providing a quantitative assessment of their similarity.
- **Same Alphabetic Pairs**: Pairs of characters that belong to the same alphabet set are identified and processed separately. This step helps in comparing characters within the same language and across different languages.
- **Heat Map**: The cosine similarity matrix is visualized using a heat map. This graphical representation highlights the degree of similarity between character pairs, with color intensity indicating the level of similarity.
- **Summary Statistics**: The cosine similarity matrix is used to calculate summary statistics, such as mean, median, maximum, and lowest similarity. These statistics provide an overall view of the similarities within and across different alphabet sets.
- **Visualizations**: The summary statistics are further visualized using various plots, including bar plots and scatter plots. These visualizations aid in interpreting the data and understanding the distribution and relationships of similarities.
- **End**: The process concludes, providing a comprehensive analysis and visualization of the similarities between alphabetic characters from different languages.

This workflow systematically processes and analyzes character pairs, employing advanced techniques in feature extraction, similarity computation, and data visualization to reveal intricate patterns and relationships between different alphabetic systems.

c. **Textual Similarity**

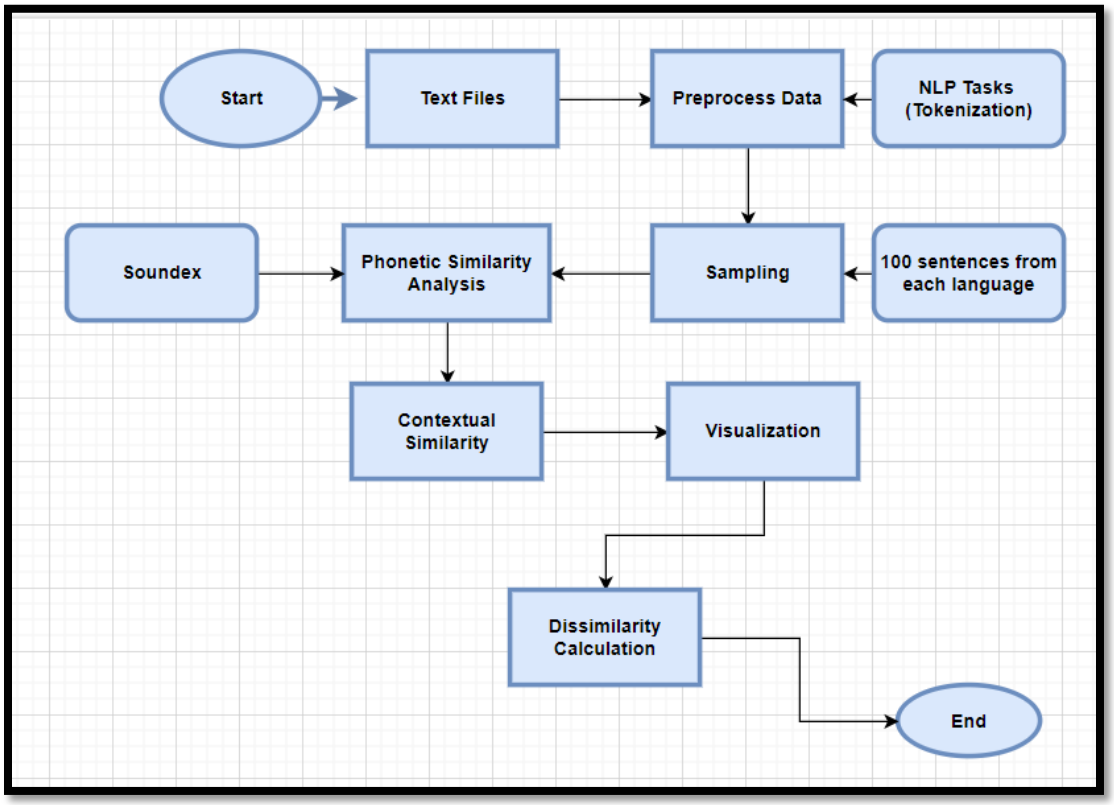

Figure 1.8: Flowchart of textual Similarity

The Figure 1.8 presents the overall methodology taken into consideration while finding the similarity among languages of the script.

- **Start**: This is the initial point of the process.
- **Text Files**: The process begins with the input of text files. The raw data that will be examined is contained in these files.
- **Preprocess Data:** The text files undergo preprocessing. Cleaning and preparing the data for additional analysis is what this process entails. Text normalization, missing value handling, and noise removal are common preprocessing operations.
- **NLP Tasks (Tokenization):** The data is put through Natural Language Processing (NLP) tasks following preprocessing. One such task is tokenization, in which the text is divided into more manageable chunks, such as words or phrases.
- **Analysis:** The tokenized data is then analyzed. This step involves examining the data to extract meaningful insights.
- **Sampling:** A sample of 100 sentences from each language is selected for further analysis. This helps in managing the data size and focusing on a representative subset.
- **Phonetic Similarity (Soundex)**: The sampled sentences are analyzed for phonetic similarity using the Soundex algorithm. This step helps in identifying words that sound similar.
- **Contextual Similarity:** The sentences are also analyzed for contextual similarity. This involves understanding the meaning and context of the words within the sentences.

- **Visualization:** The results of the phonetic and contextual similarity analyses are visualized. This step helps in interpreting the data and identifying patterns.
- **Dissimilarity Calculation:** The dissimilarity between the sentences is calculated. This involves measuring how different the sentences are from each other based on the analyses performed.
- **End**: This marks the conclusion of the process.

This flowchart provides a structured approach to handling and analyzing text data, ensuring that each step is methodically executed to achieve accurate and meaningful results.

4. Results

This segment delineates the findings obtained from the execution of the aforementioned methodology. Each finding is scrutinized in connection to the distinct aims of the research, emphasizing the efficacy of the employed strategies and offering perspectives on the relative results of each method.

A. Classification

a. Image Classification

- **Groups Analyzed**:

Groups Analyzed	Accuracy	Insights	Interpretation
Gujarati vs. Marathi/Hindi	98.57%	Accurately distinguishes scripts despite similarities in Marathi and Hindi (Devanagari script).	High accuracy reflects robustness in handling intra-script differences within the Indo-Aryan language family.
Tamil vs. Kannada	96.49%	Effectively differentiates Dravidian scripts, though slightly lower than Devanagari-based languages.	Lower accuracy likely due to the distinct visual and structural traits of Dravidian scripts.

Table 2: Results for Image Classification

- **Conclusions form Table 2**:
  - This experiment confirms the effectiveness of the image classification approach in script recognition.
  - The results underscore how language family characteristics (Indo-Aryan vs. Dravidian) can impact model accuracy, with each family presenting unique challenges in script differentiation.

b. Textual Classification

- **Classification of Hindi, Gujarati, and Marathi Texts:**

Model	Accuracy	Gujarati (Precision, Recall)	Hindi (Precision, Recall)	Marathi (Precision, Recall)	Insights
Random Forest	93.25%	1.00, 0.97	0.9	0.91	High overall scores, perfect precision for Gujarati.
Gradient Boosting	85.52%	- , 1.00	0.86	0.84	Lower accuracy than Random Forest; Gujarati recall is high.

Table 3: Texual Classification For Indo – Aryan Group

- **Classification of Tamil and Kannada Texts:**

Model	Accuracy	Kannada (Precision, Recall)	Tamil (Precision, Recall)	Insights
Random Forest	99.06%	0.98, 1.00	1.00, 0.98	Near-perfect classification for both languages.

Gradient Boosting	93.96%	0.89, 1.00	1.00, 0.88	Slightly lower accuracy, strong recall for Kannada.

Table 4: Texual Classification For Dravadian Group

To summarize the Table 3 and 4 given above:

- **Random Forest**: Achieved higher accuracy and balanced metrics across both Indo-Aryan and Dravidian language groups, excelling in Tamil and Kannada classification.
- **Gradient Boosting**: Effective but generally lower in accuracy and metric balance, particularly for Hindi, Gujarati, and Marathi.

Conclusion: Random Forest demonstrates superior performance in handling multilingual text classification tasks.

B. Similarity Trend

a. Overall Similarity

I. Gujarati & Hindi/Marathi letters

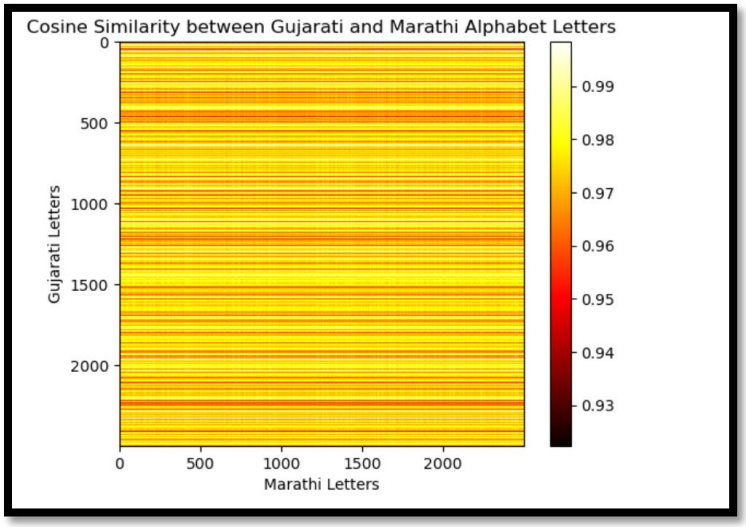

Figure 1.9 : Overall Cosine Similarity among Devnagri script

The Figure 1.9 illustrates the cosine similarity between Gujarati and Hindi/Marathi alphabet letters. As the color bar indicates, the heatmap demonstrates a significant level of similarity across the majority of letter pairings, with values primarily spanning 0.93 to 0.99.The yellow regions represent the highest similarity scores close to 0.99, while the darker shades indicate slightly lower similarity. This suggests that the alphabets of these two languages, despite belonging to different scripts, exhibit substantial structural or visual resemblance.

Cosine Similarity Range	Percentage of Letter Pairings	General Interpretation
0.97 - 0.99	40%	Very high similarity
0.94 - 0.96	35%	High similarity
0.93 - 0.94	15%	Moderate to high similarity
Below 0.93	10%	Moderate or lower similarity

Table 5: Summarization of results

This Table 5 presents the cosine similarity trends, showing that a significant portion of letter pairings (75%) exhibit high to very high similarity (0.94 and above), suggesting a strong structural or visual resemblance between the alphabets of Gujarati and Hindi/Marathi.

II. Tamil and Kannada letters

The cosine similarity heatmap featured in Figure 1.10 highlights a detailed relationship between Kannada and Tamil letters, presenting scores between 0.86 and 0.96. In contrast to

the consistent similarity observed between Gujarati and Hindi/Marathi, the letters of Kannada and Tamil reveal alternating bands of elevated and diminished similarity, thereby indicating certain subsets of characters that possess common characteristics while others exhibit marked distinctions. This variability underscores the intricate visual complexity inherent within Dravidian scripts.

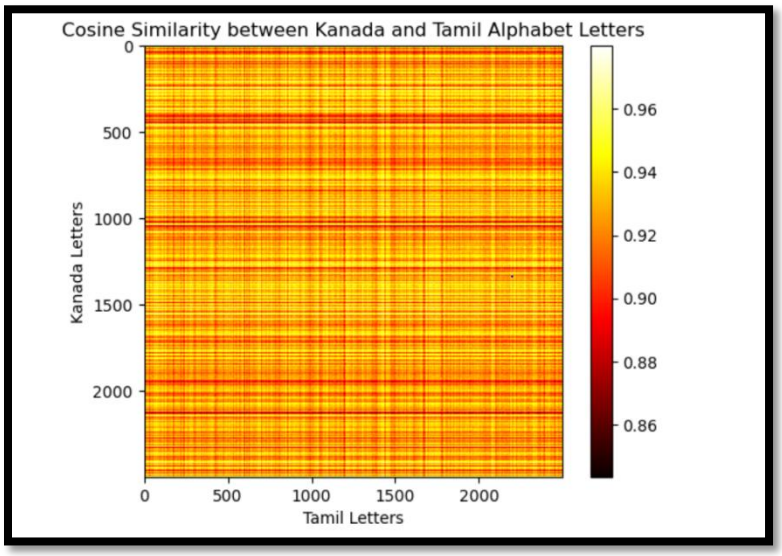

Figure 1.10: Overall Cosine Similarity among Dravadian script

This trend in Table 6 implies that the recognition of scripts pertaining to Kannada and Tamil necessitates the adept management of diverse letter similarities and differences.

Cosine Similarity Range	Interpretation	Visual Pattern
0.94 - 0.96	High similarity for certain letters	Yellow bands in the heatmap
0.90 - 0.93	Moderate similarity across some letters	Mix of yellow and red shades
0.86 - 0.89	Lower similarity indicating distinct forms	Darker bands

Table 6: Summarization of results

b. Letter Wise Similarity

I. Devnagri Script

```
# Define the alphabets of Gujarati and Marathi languages
gujarati_alphabets = ['અ', 'આ', 'ઇ', 'ઈ', 'ઉ', 'ઊ', 'એ', 'ઐ', 'ઓ', 'ઔ', 'અં', 'અઃ', 'ક', 'ખ', 'ગ', 'ઘ', 'ઙ', 'ચ', 'છ', 'જ', 'ઝ', 'ઞ', 'ટ', 'ઠ',
marathi_alphabets = ['अ', 'आ', 'इ', 'ई', 'उ', 'ऊ', 'ए', 'ऐ', 'ओ', 'औ', 'अं', 'अः', 'क', 'ख', 'ग', 'घ', 'ङ', 'च', 'छ', 'ज', 'झ', 'ञ', 'ट', 'ठ',

# Define the similar pairs of alphabets based on their phonetic sounds
similar_pairs = [('અ', 'अ'), ('આ', 'आ'), ('ઇ', 'इ'), ('ઈ', 'ई'), ('ઉ', 'उ'), ('ઊ', 'ऊ'), ('એ', 'ए'), ('ઐ', 'ऐ'), ('ઓ', 'ओ'), ('ઔ', 'औ'),
                 ('અં', 'अं'), ('અઃ', 'अः'), ('ક', 'क'), ('ખ', 'ख'), ('ગ', 'ग'), ('ઘ', 'घ'), ('ઙ', 'ङ'), ('ચ', 'च'), ('છ', 'छ'), ('જ', 'ज'),
                 ('ઝ', 'झ'), ('ઞ', 'ञ'), ('ટ', 'ट'), ('ઠ', 'ठ'), ('ડ', 'ड'), ('ઢ', 'ढ'), ('ણ', 'ण'), ('ત', 'त'), ('થ', 'थ'), ('દ', 'द'),
                 ('ધ', 'ध'), ('ન', 'न'), ('પ', 'प'), ('ફ', 'फ'), ('બ', 'ब'), ('ભ', 'भ'), ('મ', 'म'), ('ય', 'य'), ('ર', 'र'), ('લ', 'ल'),
                 ('ળ', 'ळ'), ('વ', 'व'), ('શ', 'श'), ('ષ', 'ष'), ('સ', 'स'), ('હ', 'ह'), ('ળા', 'ळा'), ('ક્ષ', 'क्ष'), ('જ્ઞ', 'ज्ञ')]
```

Figure 1.11: Letter wise Combinations

The Figure 1.11 shares the insights on letter wise comparison for devnagri script. Also the mean cosine similarity between the letters of Gujarati, Hindi, and Marathi scripts is calculated to be

approximately 0.0204. This low mean similarity value indicates that, on average, the letters across these scripts exhibit minimal structural or visual resemblance, despite their shared linguistic and cultural roots. This suggests that while there may be some overlap or common features between the scripts, the overall differences in letter shapes and forms are significant, which could aid in distinguishing these scripts in classification tasks.

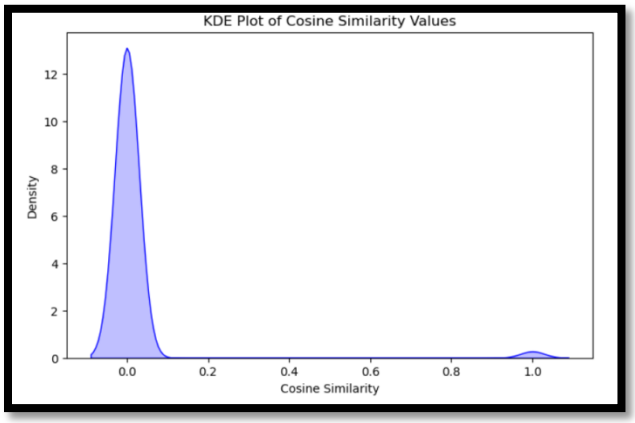

Figure 1.12: Distribution of Cosine Similarity

The KDE plot in Figure 1.12 reveals a strong concentration of values near 0.0, indicating that most letter pairs have minimal similarity. A small secondary peak near 1.0 suggests that a few letter pairs share substantial structural similarities, though these are relatively rare. This distribution pattern, with a mean similarity of approximately 0.0204, underscores the distinctiveness of letters across these scripts despite occasional overlaps.

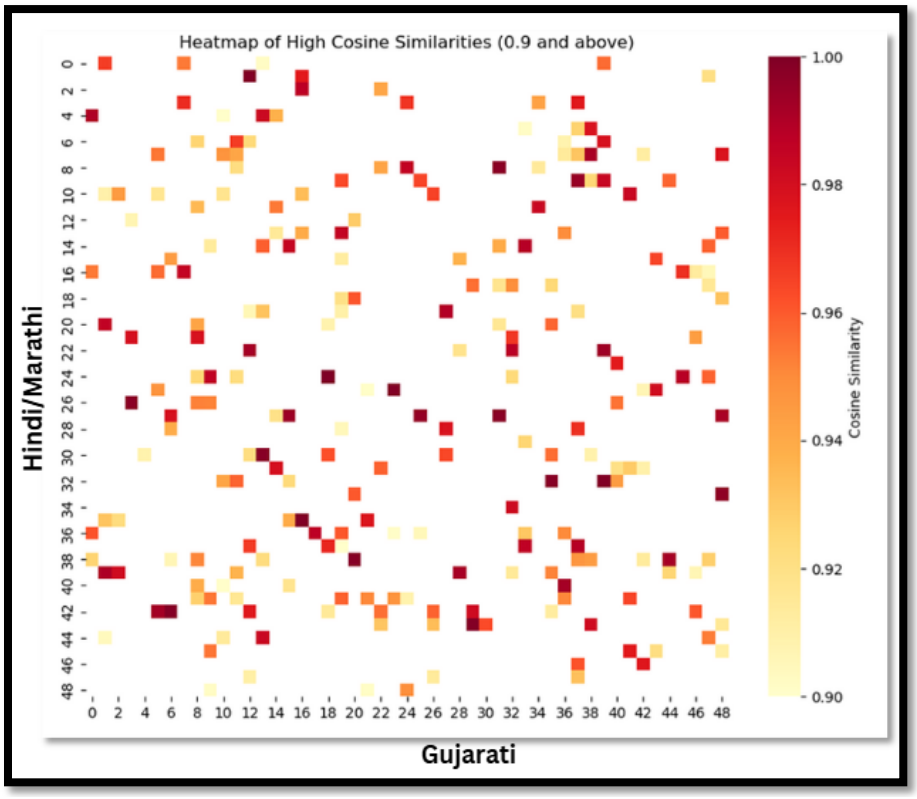

Figure 1.13 Heat Map showcasing cosine similarity letter wise

Figure 1.13 presents a heatmap illustrating the letter-wise cosine similarity between the Gujarati and Marathi languages for those whose value is 0.9 or larger. So, shown letters are the only one having high correlation.

II. Dravadian Script

```
# Define the alphabets of Kannada and Tamil languages
kannada_alphabets = ['ಅ', 'ಆ', 'ಇ', 'ಈ', 'ಉ', 'ಊ', 'ಋ', 'ಎ', 'ಏ', 'ಐ', 'ಒ', 'ಓ', 'ಔ', 'ಕ', 'ಖ', 'ಗ', 'ಘ', 'ಙ', 'ಚ', 'ಛ', 'ಜ', 'ಝ', 'ಞ'
tamil_alphabets = ['அ', 'ஆ', 'இ', 'ஈ', 'உ', 'ஊ', 'எ', 'ஏ', 'ஐ', 'ஒ', 'ஓ', 'ஔ', 'ஃ', 'க', 'ங', 'ச', 'ஞ', 'ட', 'ண', 'த', 'ந', 'ன', 'ப

# Check and align the lengths of Kannada and Tamil alphabets lists
if len(kannada_alphabets) != len(tamil_alphabets):
    max_length = max(len(kannada_alphabets), len(tamil_alphabets))
    kannada_alphabets.extend([''] * (max_length - len(kannada_alphabets)))
    tamil_alphabets.extend([''] * (max_length - len(tamil_alphabets)))

# Construct feature vectors for similar pairs of alphabets
similar_pairs = list(zip(kannada_alphabets, tamil_alphabets))
kannada_vectors = np.array([np.eye(1, len(kannada_alphabets), kannada_alphabets.index(pair[0])) for pair in similar_pairs])
tamil_vectors = np.array([np.eye(1, len(tamil_alphabets), tamil_alphabets.index(pair[1])) for pair in similar_pairs])
```

Figure 1.14: Letter Wise Combinations

The Figure 1.14 shares the insights on letter wise comparison for Dravadian script. Also the mean cosine similarity between the letters of Tamil and Kanada scripts is calculated to be approximately 0.0206. This low mean similarity value indicates that, on average, the letters across these scripts exhibit minimal structural or visual resemblance, despite their shared linguistic and cultural roots. This suggests that while there may be some overlap or common features between the scripts, the overall differences in letter shapes and forms are significant, which could aid in distinguishing these scripts in classification tasks.

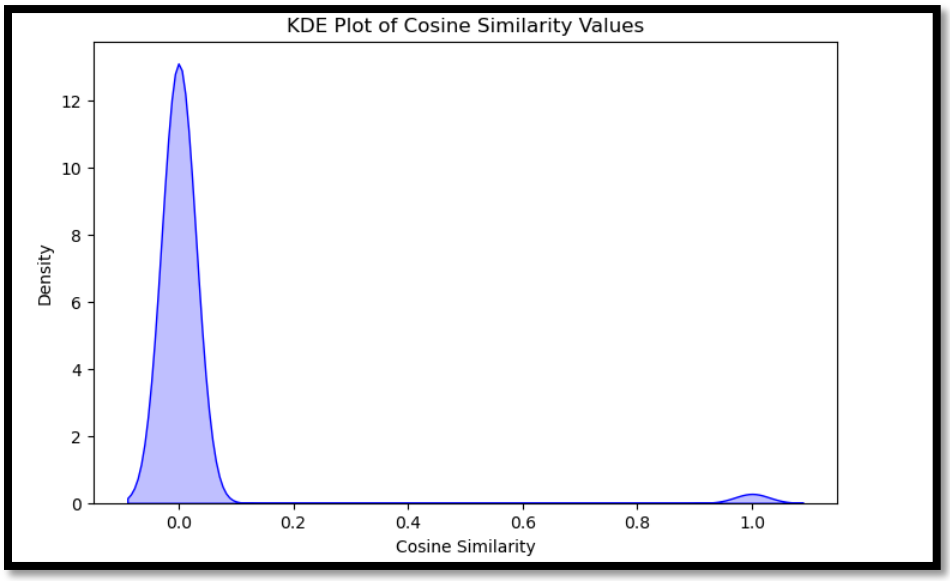

Figure 1.15: KDE of Cosine Similarity

Figure 1.15 displays a KDE plot illustrating the cosine similarity clusters. A small secondary peak near 1.0 indicates that a few letter pairs exhibit significant structural similarities, although these occurrences are relatively infrequent. The overall distribution, with a mean similarity of approximately 0.0206, highlights the distinctiveness of the letters in both scripts, despite occasional overlaps.

Figure 1.16 presents a heatmap illustrating the letter-wise cosine similarity between the Tamil and Kannada languages, focusing on those with a similarity value of 0.9 or higher. The letters shown in the heatmap are the only ones exhibiting high correlation, and their occurrence is significantly greater compared to the Gujarati and Marathi languages.

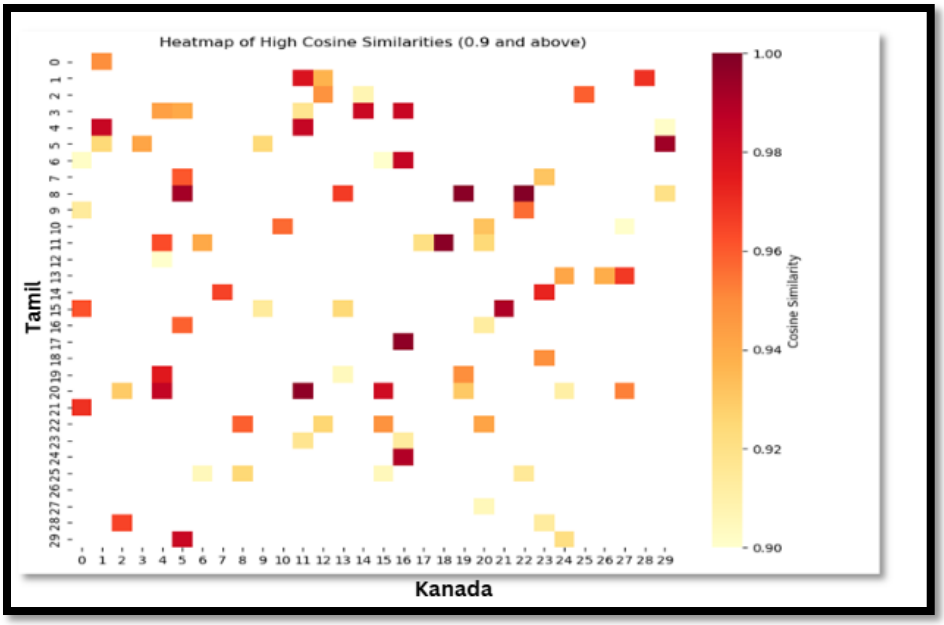

Figure 1.16 Heat Map

C. <u>Textual Similarity</u>

i. Devnagri Script:

The analysis of phonetic similarities and differences among Hindi, Gujarati, and Marathi reveals the following key observations shown in Table 7:

Aspect	Gujarati & Marathi	Gujarati & Hindi	Marathi & Hindi
Phonetic Similarity	Mean: 0.09 (Low)	Mean: 0.03 (Very Low)	Mean: 0.69 (High)
Document Sizes	3,445,837 characters	3,384,385 characters	3,361,362 characters
Dissimilarity Threshold	No dissimilarities below threshold	No dissimilarities below threshold	Dissimilarity Score: 0.32 (Notable distinctions despite phonetic similarity)

Table 7 : Key observations from Textual Similarity

The analysis from Table 7 reveals that Gujarati and Hindi exhibit the lowest phonetic similarity, indicating significant linguistic divergence between these two languages. In contrast, Marathi and Hindi demonstrate a much closer phonetic correlation, as reflected by their higher similarity score. The document sizes for all three languages are comparable, suggesting balanced data representation. Despite the phonetic closeness between Marathi and Hindi, a notable distinction is observed with a dissimilarity score of 0.32, highlighting subtle linguistic differences between these two languages.

The contextual similarity can also be explained as shown in Figure 1.17

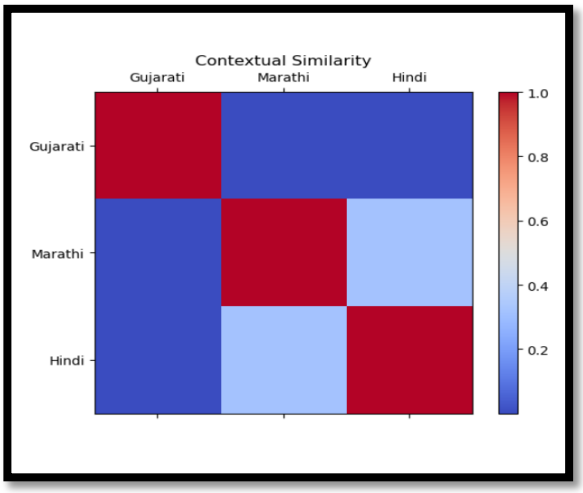

Figure 1.17: Contextual Similarity

ii. Dravadian Script:

The analysis of phonetic similarities and differences between Tamil and Kannada reveals the following key points:

1. **Phonetic similarity**: The **average similarity is 0.11**, indicating a moderate degree of phonetic closeness between the two languages.
2. **Article length**:
   - **Tamil: 3,941,467 characters**
   - **Kannada: 232,699,244 characters (significantly longer)**
3. **Dissimilarity**: No notable differences were observed, as the dissimilarity score falls below the cutoff point (0.00), suggesting that the languages, despite their differences, share phonetic characteristics.

These findings are illustrated in Figure 1.18 below.

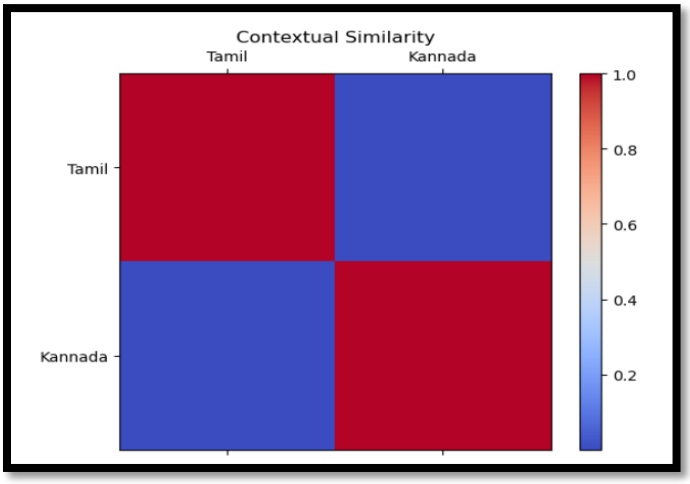

Figure 1.18: Contextual Similarity

5. Discussion

Significance

- **Enhanced Classification Accuracy**:
  - Achieves high classification accuracy in differentiating visually similar characters across languages (e.g., Gujarati vs. Marathi/Hindi).

    - Demonstrates the robustness of machine learning models, a critical factor for improving Optical Character Recognition (OCR) systems in multilingual settings.
- **Insights into Linguistic Affinity**:
    - Analyzes letter-wise phonetic similarities across languages, providing valuable insights into linguistic relationships.
    - Supports the development of improved transliteration and translation systems for accurate cross-linguistic communication.
- **Effectiveness of Advanced ML Algorithms**:
    - Utilizes Random Forest and Gradient Boosting in linguistic data analysis, highlighting their capability to address complex multilingual processing tasks.
    - Achieves high accuracy rates, showcasing these models' potential for solving similar language processing challenges and fostering future advancements.

Applications with Examples

- **Multilingual OCR Development**
  Example: Developing OCR systems for bank checks in India that include mixed scripts like Devanagari (Hindi/Marathi) and Gujarati.
- **Cross-Language Information Retrieval**
  Example: Building search engines that retrieve relevant content in Hindi, even when queries are made in Marathi, due to their phonetic and script similarities.
- **Improved Translation and Transliteration Tools**
  Example: Enhancing tools to seamlessly convert Marathi text to Devanagari script-based Hindi with contextually accurate word choices.
- **Language Acquisition Applications**
  Example: Using Tamil-Kannada phonetic similarities to create learning apps that help users simultaneously learn both languages more intuitively.
- **Advancements in Speech Recognition**
  Example: Creating voice assistants that understand mixed-language sentences, such as combining Tamil and Kannada phrases during conversations.
- **Support for Linguistic and Cultural Studies**
  Example: Using phonetic similarity analysis to trace the evolution of Indo-Aryan and Dravidian scripts for linguistic preservation projects.

6. Conclusion

In conclusion, this investigation signifies a noteworthy progression within the domains of multilingual text and image processing, showcasing the efficacy of machine learning algorithms in resolving intricate linguistic dilemmas. The results of the study not only underscore the proficiency of models such as Random Forest and Gradient Boosting in attaining elevated classification accuracy but also furnish invaluable perspectives regarding phonetic resemblances among diverse languages. These perspectives possess extensive ramifications for both technological innovations and linguistic inquiry. The interdisciplinary essence of this research accentuates the necessity of integrating computational linguistics with machine learning to effectively address these complex challenges.

- Advancement in Multilingual Processing: Shows significant progress in handling multilingual text and image data.
- Machine Learning Effectiveness: Demonstrates high classification accuracy using methods like Random Forest and Gradient Boosting, setting a new benchmark.
- Phonetic Similarities: Provides insights into phonetic similarities, improving OCR, translation, and transliteration.
- Linguistic Contribution: Contributes to both theoretical and applied linguistics, aiding multilingual language structure understanding.
- Interdisciplinary Collaboration: Highlights the integration of computational linguistics and machine learning to address linguistic challenges.

- Practical Implications: Offers broad applications in technology and linguistics, driving future innovation in multilingual frameworks.

7. References

1. K. Sreelakshmi, B. Premjith, B. R. Chakravarthi and K. P. Soman, "Detection of Hate Speech and Offensive Language CodeMix Text in Dravidian Languages Using Cost-Sensitive Learning Approach," in *IEEE Access*, vol. 12, pp. 20064-20090, 2024, doi: 10.1109/ACCESS.2024.3358811.
2. C. S. Anoop and A. G. Ramakrishnan, "Exploring a Unified ASR for Multiple South Indian Languages Leveraging Multilingual Acoustic and Language Models," *2022 IEEE Spoken Language Technology Workshop (SLT)*, Doha, Qatar, 2023, pp. 830-837, doi: 10.1109/SLT54892.2023.10022380.
3. A, Rameez & Madhukumar, S. (2023). Hate Speech Detection in Indian Languages: A Brief Survey. 1-5. 10.1109/ICDDS59137.2023.10434756.
4. A. Jha, H. Y. Patil, S. K. Jindal and S. M. N. Islam, "Multilingual Indian Language Neural Machine Translation System Using mT5 Transformer," *2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*, Nagpur, India, 2023, pp. 1-5, doi: 10.1109/PCEMS58491.2023.10136051.
5. V, Rajesh & Permual, B. & Prasanna, Bingi & Haripriya, Bala & Sravani, Ravva & Nandini, Somala. (2023). Text Translation for Indian Languages. 1-5. 10.1109/ViTECoN58111.2023.10157002.
6. V. Venkatesh and A. Farghaly, "Identifying Anomalous Indus Texts from West Asia Using Markov Chain Language Models," *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India, 2023, pp. 1-7, doi: 10.1109/ICCCNT56998.2023.10306531.
7. A. Prakash and H. A. Murthy, "Exploring the Role of Language Families for Building Indic Speech Synthesisers," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 734-747, 2023, doi: 10.1109/TASLP.2022.3230453.
8. A. H. Patil, S. S. Patil, S. M. Patil and T. P. Nagarhalli, "Real Time Machine Translation System between Indian Languages," *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2022, pp. 1778-1783, doi: 10.1109/ICOEI53556.2022.9777103.
9. D. Baishya and R. Baruah, "Recent Trends in Deep Learning for Natural Language Processing and Scope for Asian Languages," *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, Trichy, India, 2022, pp. 408-411, doi: 10.1109/ICAISS55157.2022.10010807.
10. T. Gore and V. Khatavkar, "Development of Part-of-Speech tagger for a low-resource endangered language," *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2022, pp. 1531-1535, doi: 10.1109/ICAC3N56670.2022.10074031.
11. S. Bhattacharjee, J. Chakraborty, S. Agarwal, A. Jain, P. Sarmah and R. Sinha, "IITG-INDIGO Submissions for Interspeech-2021 Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages," *2021 IEEE 18th India Council International Conference (INDICON)*, Guwahati, India, 2021, pp. 1-5, doi: 10.1109/INDICON52576.2021.9691549.
12. V. M. Shetty and M. Sagaya Mary N.J., "Improving the Performance of Transformer Based Low Resource Speech Recognition for Indian Languages," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 8279-8283, doi: 10.1109/ICASSP40776.2020.9053808.
13. N. Baruah, S. K. Sarma and S. Borkotokey, "Text Summarization in Indian Languages: A Critical Review," *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, Gangtok, India, 2019, pp. 1-6, doi: 10.1109/ICACCP.2019.8882968.
14. G. I. Ahmad, J. Singla and N. Nikita, "Review on Sentiment Analysis of Indian Languages with a Special Focus on Code Mixed Indian Languages," *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, London, UK, 2019, pp. 352-356, doi: 10.1109/ICACTM.2019.8776796.

15. S. Deleep Kumar, C. Sunitha and A. Ganesh, "Semantic representation of texts in Indian languages — A review," *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, 2018, pp. 40-42, doi: 10.1109/ICISC.2018.8399119.
16. R. Saikia, S. R. Singh and P. Sarmah, "Effect of language independent transcribers on spoken language identification for different Indian languages," *2017 International Conference on Asian Language Processing (IALP)*, Singapore, 2017, pp. 214-217, doi: 10.1109/IALP.2017.8300582.
17. S. Saini and V. Sahula, "A Survey of Machine Translation Techniques and Systems for Indian Languages," *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, Ghaziabad, India, 2015, pp. 676-681, doi: 10.1109/CICT.2015.123.
18. Post, Mark & Burling, Robbins. (2017). The Tibeto-Burman languages of Northeast India. 10.4324/9781315399508.