**Phonetic Patterns and Cross-Language Correspondences: A Polyglot Comparative Linguistics Analysis**

1Dhairvi Shah, 2Preeti Kale

[1dhairvi.shah@mitwpu.edu.in](mailto:1dhairvi.shah@mitwpu.edu.in) , [2preeti.kale@mitwpu.edu.in](mailto:2preeti.kale@mitwpu.edu.in)

1, 2 Department of Computer Engineering & Technology, Dr. Vishwanath Karad MIT-World Peace University, Pune, India

***Abstract***

*This research looks into the examination of phonetic similarities across various languages, notably centering on the Dravidian and Devnagri scripts. It is composed of three main components: (1) language categorization; (2) letter-by-letter likeness analysis; and (3) overall language similarity assessment The study achieved great classification accuracies, obtaining 0.9857 for Gujarati-Marathi/Hindi and 0.9649 for Tamil-Kannada, by utilizing the methodologies of Random Forest and Gradient Boosting Through cosine similarity analyses, the study 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 (0.69) and Tamil-Kannada (0.11). These outcomes highlight the strides taken in multilingual optical character recognition (OCR), transliteration, and translation systems. The implications of this study extend to various applications such as speech recognition, language acquisition, and linguistic studies, highlighting the extensive technological and interdisciplinary impact of the research. This study 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, Evolution and Linguistic Features.*

1. **Introduction:**

Linguistic landscapes of India reflects 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.

This study explores the phonetic correspondences among the varied Indian subcontinent's languages, concentrating on the Dravidian and Devanagari scripts. 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 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.

All things considered, this work sets a new standard for multilingual linguistic analysis in addition to providing clarification on the phonetic correspondences and cross-linguistic patterns present in the Indian scripts. By leveraging archaeological discoveries, linguistic experiments, and historical records, we aim to unravel the complex webs that have constructed the linguistic legacy of the Indian subcontinent, thus contributing to our understanding of human civilization’s extraordinary journey across this diverse and historically rich region.

1. **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 as given below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Study** | **Techniques** | **Application** | **Future Work** | **Languages** | **Dataset** |
| **[1]** | **Transformer-based embedding models** | **Recognition of hate speech in Dravidian languages** | **Understanding misclassification** | **Kanada Malyalam, Tamil** | **Dravidian CodeMix: Kannada-English, Malayalam-English, Tamil-English. From offensive language identification tasks.** |
| **[2]** | **Wikipedia text corpora, Unified multilingual model** | **Uniform ASR for South Indian languages** | **Exploration of language coverage and model effectiveness** | **Kannada, Telugu, Sanskrit, Malayalam, Tulu** | **Speech data: Telugu, Kannada, Malayalam, Sanskrit. Includes training, development, testing subsets. Sanskrit has OOD subset.** |
| **[3]** | **SVM, Random Forest, m-BERT** | **Hate speech detection in Indian languages** | **Improvement of classifier performance** | **Hindi, Dravadian** | **Organizers' datasets: HASOC, TRAC, Dravidianlangtech. Fine-grained labels available for detailed hate speech info.** |
| **[4]** | **mT5 transformer, Asian Language Treebank database** | **Multilingual Neural Machine Translation System** | **Tackling translation difficulties** | **Hindi, Bengali, English, Bhojpuri, Sindhi, Magadhi, Marathi, URDU** | **Modified Asian Language Treebank dataset (Hindi corpus).** |
| **[5]** | **LSTM, attention techniques** | **Inter-language translation for Indian languages** | **Further research in machine translation** | **Hindi, Odia, Malayalam, Indian Sign Language** | **Review paper.** |
| **[6]** | **Markov Chain, n-gram models** | **Analysis of Indus Valley civilization's writing system** | **Understanding ancient communication techniques** | **Indus Script** | **Texts from ICIT database in CSV files. Indus Corpus.** |
| **[7]** | **Weiner filter, HOG, ANN** | **Handwritten digit recognition in South Indian languages** | **Integration of ANN and HOG for accuracy** | **Malayalam, Kannada, Telugu, Devanagiri, Hindi** | **Handwritten digit images: Malayalam, Kannada, Telugu, Devanagiri, Hindi. Binary images (130x66 pixels) with various writing styles and numeral sizes.** |
| **[8]** | **NMT, RBMT, SMT** | **Machine translation for Indian languages** | **Refinement of translation techniques** | **Hindi, Punjabi, Bengali, Kannad, Marathi, Telugu** | **Survey paper.** |
| **[9]** | **Language-independent transcribers, RNN, SVM** | **Automatic Spoken Language Identification (LID)** | **Improvement of LID accuracy** | **Language transcribers** | **Studio Record dataset. 8 transcribers for Indian languages.** |
| **[10]** | **Tokenization, semantic triplets, WordNet, TF-IDF** | **Linguistic semantic structure for Indian languages** | **Further refinement of semantic graphs** | **English, European, Indian** | **NA.** |
| **[11]** | **Lexicon-based analysis, machine learning** | **Sentiment analysis in code-mixed text** | **Development of sentiment analysis resources** | **Code-mixed languages** | **Review paper. English-Hindi, English-Bengali, English-Tamil, English-Urdu.** |
| **[12]** | **Statistical and linguistic approaches, topic modeling** | **Text summarization for Indian languages** | **Improvement of summarization effectiveness** | **Hindi, Gujarati, Marathi, Tamil, Malayalam, Odia, Punjabi, Assamese, Tamil** | **Review paper.** |
| **[13]** | **Hybrid CTC/attention, Multilingual Transformer** | **Speech recognition for low-resource Indian languages** | **Enhancement of speech recognition accuracy** | **Gujarati, Tamil, Telugu** | **Microsoft, SpeechOcean.com challenge: Wave files + UTF-8 transcriptions.** |
| **[14]** | **CTC loss, CNN, TDNN** | **Multilingual and code-switching ASR** | **Integration of LM and noise reduction for ASR improvements** | **Gujarati, Hindi, Odia, Marathi, Telugu, Tamil, Bengali** | **Microsoft Research Open Data, Hindi stories, local dialects.** |
| **[15]** | **Hidden Markov Model (HMM)** | **Part-of-Speech tagging for Indian languages** | **Utilization of larger corpus and tagsets** | **Marathi, Hindi, Assamese, Bengali, Telugu, Konkani, Manipuri** | **Katkari dataset.** |
| **[16]** | **Deep neural networks (Recursive Auto Encoder, CNN, LSTM, Bidirectional GRU)** | **Automated creation and analysis of natural language** | **Integration of deep learning into Indian languages** | **Hindi, Malayalam, Punjabi, Assamese** | **ILCI Phase II Malayalam dataset.** |
| **[17]** | **LSTM, Encoder-Decoder model** | **Machine translation between Indian languages** | **Improvement of translation accuracy** | **Marathi to Gujarati** | **Indicnlp.ai4bharat.org/samanantar/.** |
| **[18]** | **Multilingual TTS systems** | **Multilingual voice training for Indian languages** | **Designing Indic speech synthesis systems** | **Indo-Aryan, Dravidian** | **Indic TTS database. Bengali, Gujarati, Hindi, Odia, Rajasthani (Devanagari script), Kannada, Malayalam, Tamil, Telugu.** |

**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.

1. **Methodology**
2. **Classification**
3. **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. The following figure 3.1.a throws the insights on the technique applied.

The flowchart illustrated in figure 3.1.a is a structured approach for establishing and educating a Convolutional Neural Network (CNN) in order to discern among different languages through their visual depictions. Commencing with the organization of data directories, the process entails meticulous preprocessing of the images, establishing specific image dimensions and batch sizes, and applying sophisticated data augmentation techniques to enhance dataset variability. Subsequently, image normalization is conducted to ensure uniformity in input data. Subsequently, the design of the CNN architecture is carried out with great attention to detail, integrating components like Conv2D, MaxPooling, Flatten, and Dense layers. The model undergoes rigorous training and evaluation to assess its performance metrics. Upon achieving satisfactory results, the trained model is saved for subsequent deployment, ensuring a robust framework for language identification via image analysis.



Figure 3.1.a : Flowchart for Image Classification

1. **Textual Classification**

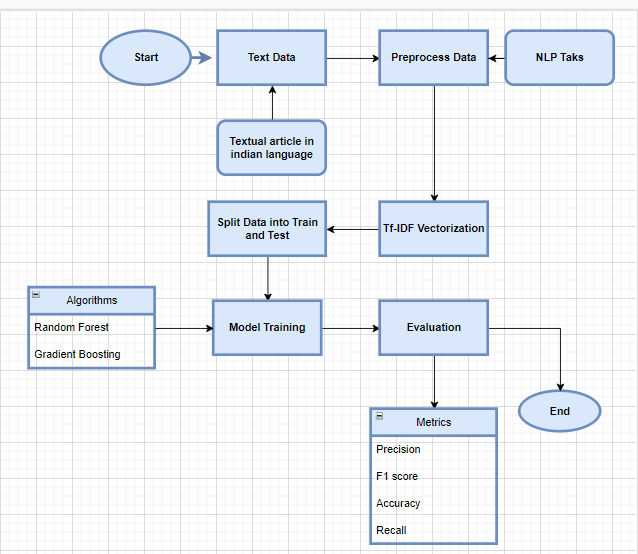
****

Figure: 3.1.b Flowchart for Text data Classification

The figure 3.1.b presents the overall methodology used for text classification.

1. Start: The process begins here.
2. Text Data: The initial input is raw text data, which could be textual articles in Indian languages.
3. 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.
4. 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.
5. 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.
6. Algorithms: Different machine learning algorithms are employed for model training. In this flowchart, two algorithms are highlighted:
   * 1. Random Forest
     2. Gradient Boosting
7. Model Training: The selected algorithms are used to train the model on the training data.
8. Evaluation: The trained model is evaluated using the test data. Various metrics are calculated to assess the model’s performance, including:
   * 1. Precision
     2. F1 Score
     3. Accuracy
     4. Recall
9. 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.

1. **Similarity Trend**
2. **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.

****

Figure 3.2: Flowchart for Overall Similarity Trend

The figure 3.2 given above represents a systematic process for analyzing image data to distinguish between different languages. The procedure is described in the following detailed steps:

1. Start: The process begins.

2. 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.

3. 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.

4. 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.

5. Flatten Images: The resized grayscale images are then flattened into one-dimensional arrays. This transformation allows for easier manipulation and analysis in subsequent steps.

6. 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.

7. 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.

8. 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.

9. 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.

1. **Letter Wise Similarity**

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.



Figure 3.3: Flowchart showing letter wise similarity analysis

The figure 3.3 depicted above 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:

1. Start: The process is initiated.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.

1. **Textual Similarity**

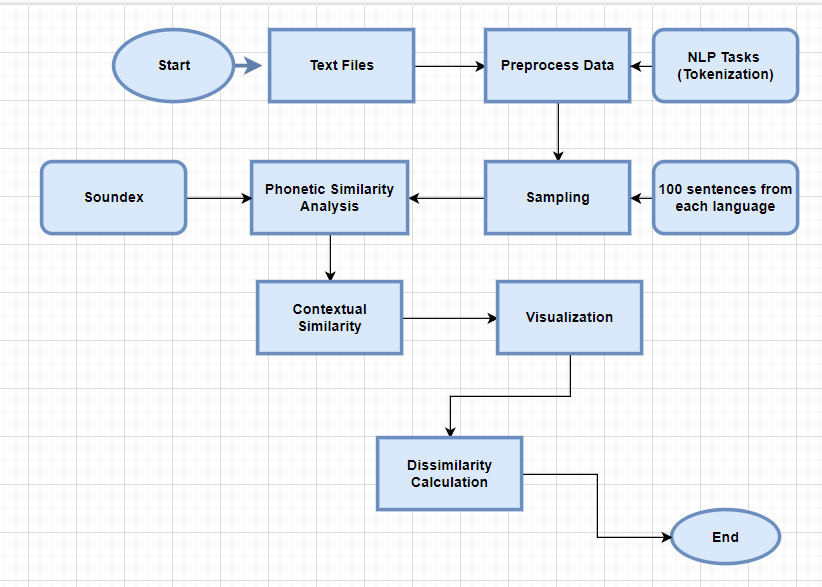


Figure 3.4: Flowchart of textual Similarity

The figure 3.4 presents the overall methodology taken into consideration while finding the similarity among languages of the script.

1. Start: This is the initial point of the process.
2. Text Files: The process begins with the input of text files. The raw data that will be examined is contained in these files.
3. 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.
4. 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.
5. Analysis: The tokenized data is then analyzed. This step involves examining the data to extract meaningful insights.
6. 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.
7. Phonetic Similarity (Soundex): The sampled sentences are analyzed for phonetic similarity using the Soundex algorithm. This step helps in identifying words that sound similar.
8. Contextual Similarity: The sentences are also analyzed for contextual similarity. This involves understanding the meaning and context of the words within the sentences.
9. Visualization: The results of the phonetic and contextual similarity analyses are visualized. This step helps in interpreting the data and identifying patterns.
10. 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.
11. 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.

1. **Results**
2. **Classification**
   1. **Image Classification**

In a recent experiment aimed at classifying images of different scripts, two distinct groups were analyzed: Gujarati and Marathi/Hindi (using the Devnagari script), and Tamil and Kannada (both Dravidian scripts). The classification model demonstrated an impressive performance, yielding an overall accuracy **of 0.9857 for the Gujarati and Marathi/Hindi script** classification task. This high accuracy indicates a robust ability of the model to differentiate between these scripts despite their visual similarities, especially between Marathi and Hindi, which share the same script.

However, the Dravidian language family's **Tamil and Kannada scripts** produced a little lower but still very good total accuracy of **0.9649** in the categorization challenge. The difference in accuracy between the two groups could be attributed to the distinct structural and visual characteristics inherent in the Devanagari and Dravidian scripts, highlighting the varying levels of complexity in script recognition tasks across different language families. These results underscore the effectiveness of the image classification approach in script identification, particularly in recognizing subtle script variations within the same linguistic family.

* 1. **Textual Classification**

The Random Forest algorithm produced findings with an accuracy of 0.9325 for the categorization of texts in Hindi, Gujarati, and Marathi, based on the approach used. All three languages had excellent memory, f1-scores, and precision according to the categorization report; Gujarati scored a flawless 1.00 for precision and a 0.97 for recall. The f1-scores for Hindi and Marathi were 0.90 and 0.91, respectively, showing good performance. On the other hand, the Gradient Boosting method produced an accuracy of 0.8552. The classification report for this approach states that Gujarati had a perfect recall of 1.00 and an accuracy of 0.76, whereas Hindi and Marathi had f1-scores of 0.86 and 0.84, respectively.

For Tamil and Kannada texts, the Random Forest algorithm achieved an impressive accuracy of 0.9906. The classification report showed near-perfect precision and recall for both languages, with Kannada having a precision of 0.98 and a recall of 1.00, and Tamil achieving a precision of 1.00 and a recall of 0.98. The Gradient Boosting algorithm, while slightly less accurate with an accuracy of 0.9396, still performed well. The classification report indicated that Kannada had a precision of 0.89 and a recall of 1.00, while Tamil had a precision of 1.00 and a recall of 0.88.

1. **Similarity Trend**
2. **Overall Similarity**
3. **Gujarati & Hindi/Marathi letters**

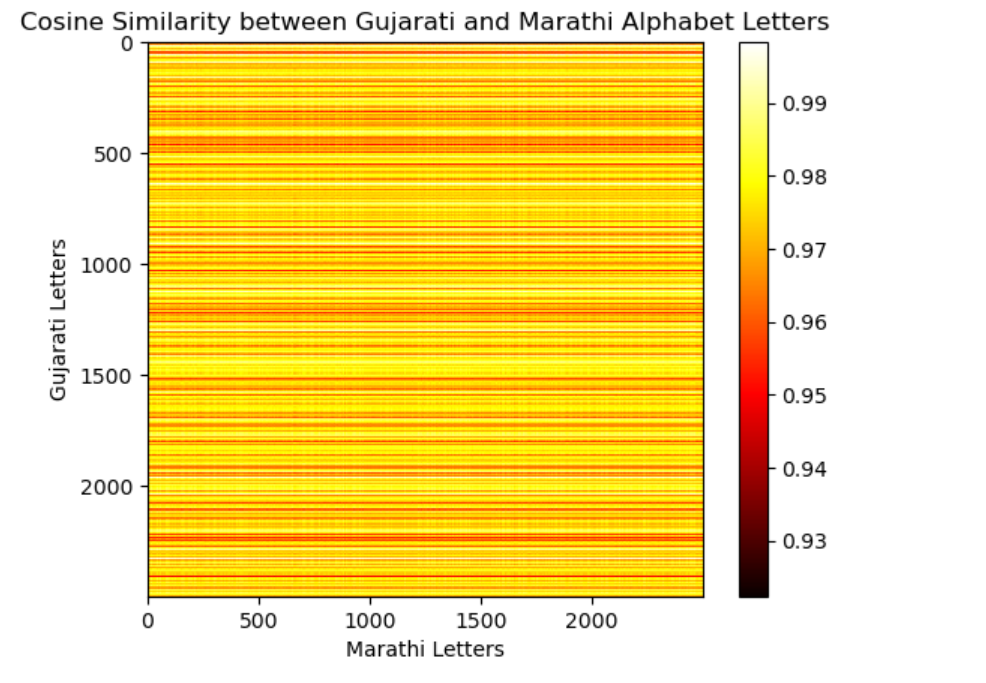
****

Figure 4.1 : Overall Cosine Similarity among Devnagri script

The figure 4.1 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.

1. **Tamil and Kannada letters**

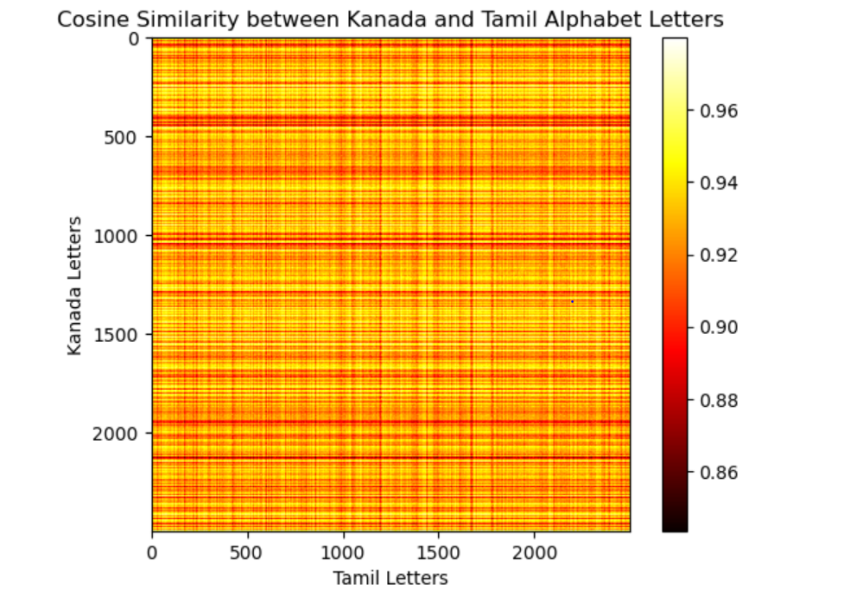
****

Figure 4.2: Overall Cosine Similarity among Dravadian script

The figure 4.2 provides a detailed analysis of the cosine similarity between Kannada and Tamil alphabet letters, represented by a heatmap where similarity scores vary between 0.86 and 0.96. The presence of alternating horizontal and vertical bands, with distinct patterns of yellow (high similarity) and red to darker shades (lower similarity), suggests that specific subsets of Kannada and Tamil letters share notable structural or visual features, while others are more distinct. This varied pattern contrasts with the relatively uniform similarity observed between Gujarati and Marathi scripts, indicating that the relationship between Kannada and Tamil scripts is more complex. The lower overall similarity, especially as highlighted by the more prevalent darker bands, might reflect the inherent differences in the orthographic and phonological systems of these two Dravidian languages. The heatmap underscores the importance of considering these subtle intra-script variations when designing and optimizing models for script recognition.

1. **Letter Wise Similarity**
2. Devnagri Script

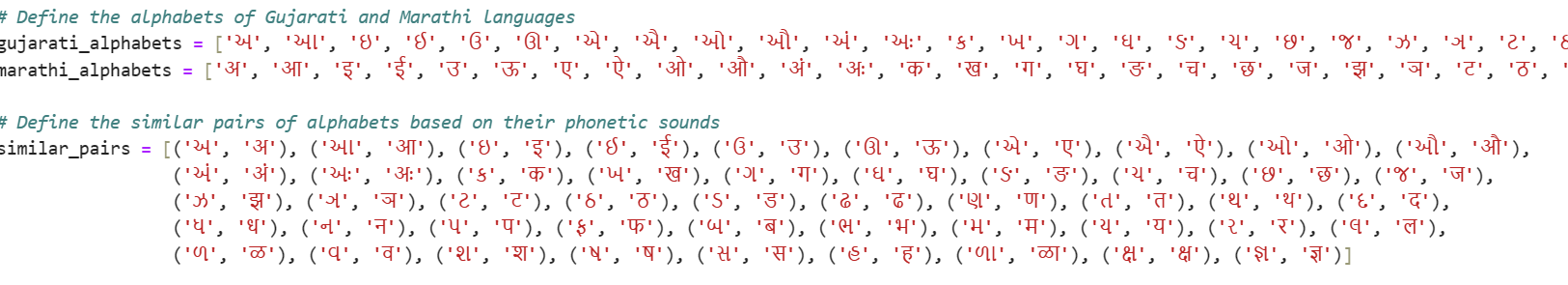


Figure 4.3: Letter wise Combinations

The figure 4.3 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.

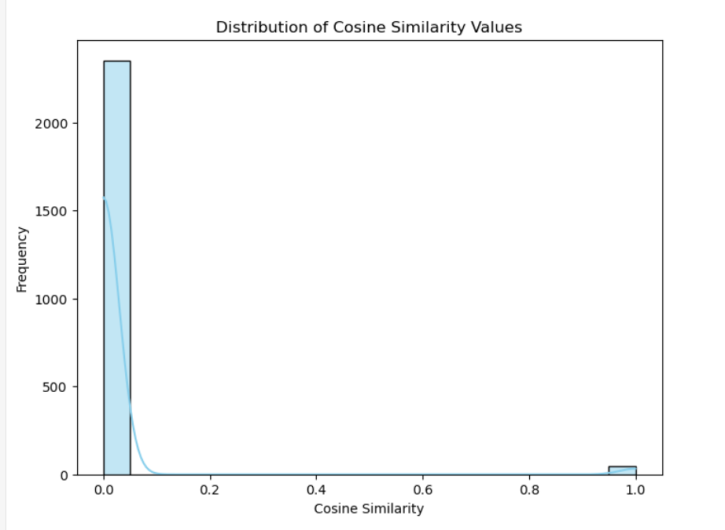
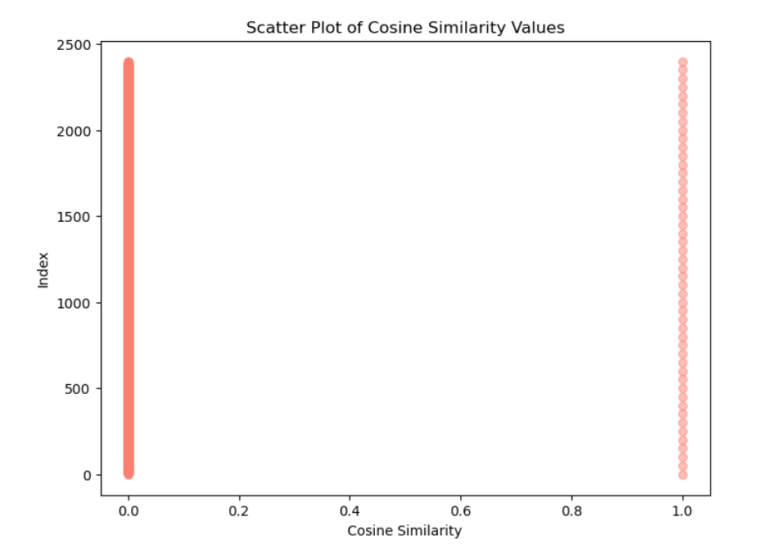


Figure 4.4: Distribution of Cosine Similarity Figure 4.5: Scatter plot Cosine similarity

The figure 4.4 presents a histogram and 4.5 presents a scatter plot displaying the distribution of cosine similarity values between letters of Gujarati, Hindi, and Marathi scripts Since most similarity values cluster around 0.0 and the distribution has a substantial bias towards the lower end, most letter pairings show extremely low similarity. A sharp peak near 0.0 highlights the distinctiveness of the letters across these scripts. There is a minor occurrence of higher similarity values closer to 1.0, which suggests that a few letter pairs share substantial structural similarities, but these instances are relatively rare. This distribution aligns with the calculated low mean similarity of approximately 0.0204, reinforcing the idea that these scripts are generally distinct despite some overlapping features.

1. Dravadian Script

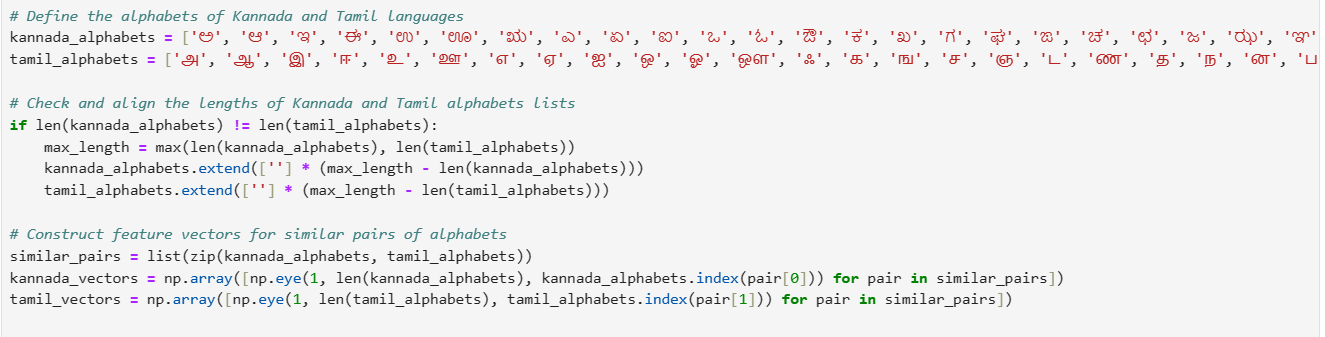


Figure 4.6: Letter Wise Combinations

The figure 4.6 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.

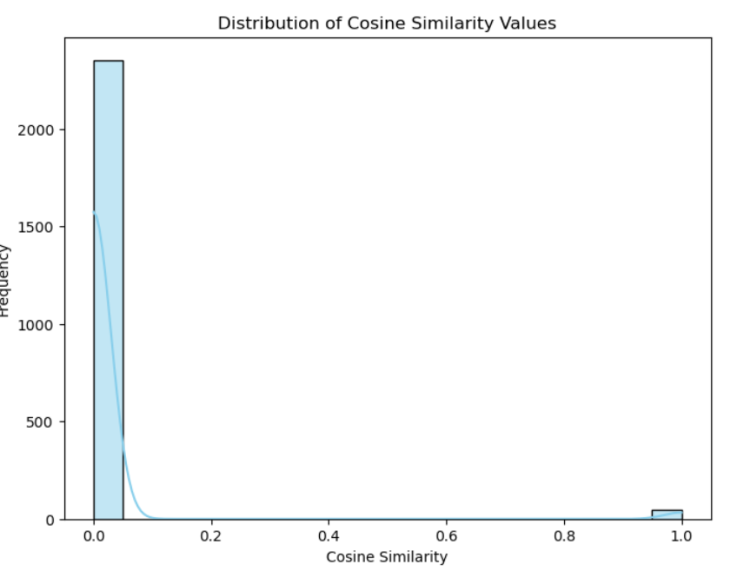
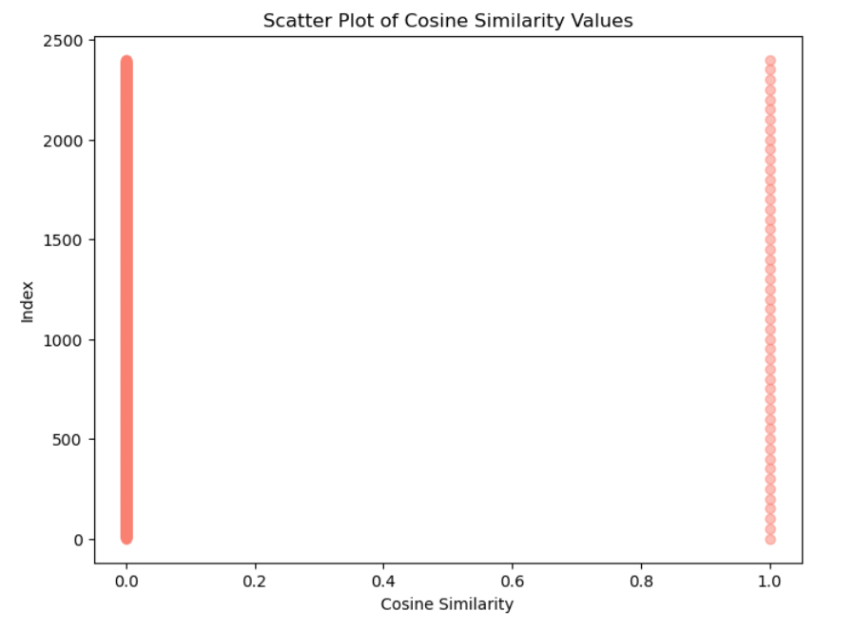


Figure 4.7: Distribution of Cosine Similarity Figure 4.8: Scatter plot

The figure 4.7 presents a histogram while figure 4.8 presents a scatter plot displaying the distribution of cosine similarity values between letters of Tamil and Kannada. The distribution is heavily skewed towards the lower end, with the majority of similarity values clustering around 0.0, indicating that most letter pairs exhibit very low similarity. A sharp peak near 0.0 highlights the distinctiveness of the letters across these scripts. There is a minor occurrence of higher similarity values closer to 1.0, which suggests that a few letter pairs share substantial structural similarities, but these instances are relatively rare. This distribution aligns with the calculated low mean similarity of approximately 0.0206, reinforcing the idea that these scripts are generally distinct despite some overlapping features.

1. **Textual Similarity**
2. Devnagri Script:

The examination of phonetic resemblances and disparities among Hindi, Gujarati, and Marathi languages uncovers some intriguing observations. The mean phonetic resemblance between Gujarati and Marathi stands relatively low at 0.09, and even lower between Gujarati and Hindi at 0.03. Nonetheless, the phonetic resemblance between Marathi and Hindi is notably higher at 0.69, signifying a closer phonetic correlation between these two languages. The sizes of the language documents are reasonably similar, with Gujarati comprising 3,445,837 characters, Marathi 3,361,362 characters, and Hindi 3,384,385 characters. Upon scrutiny of dissimilarities under a specific threshold, it was noted that both Gujarati-Marathi and Gujarati-Hindi pairs had no dissimilarities below the threshold, while the Marathi-Hindi pair had a dissimilarity score of 0.32. This indicates that despite their phonetic similarities, Marathi and Hindi still manifest notable distinctions in particular contexts. All these is as depicted in below given figures 4.9 and 4.10.

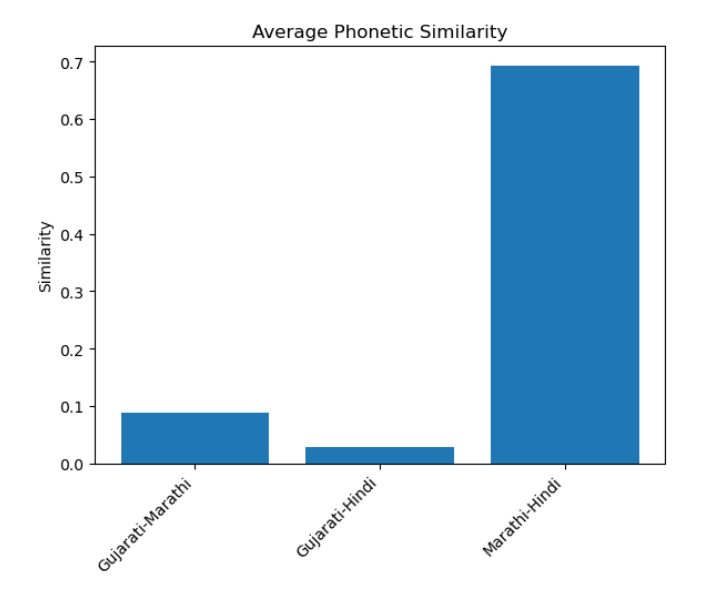
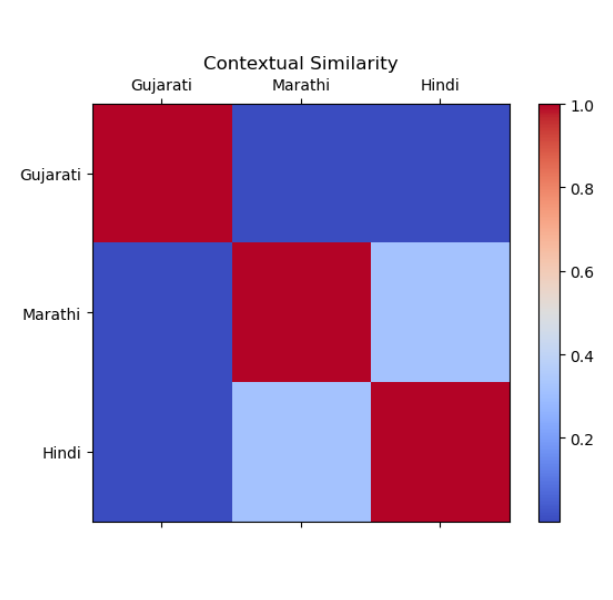


Figure 4.9 Average Phonetical Similarity Figure 4.10: Contextual Similarity

1. Dravadian Script:

The average phonetic similarity between the Tamil and Kannada languages is 0.11, suggesting a moderate degree of phonetic closeness, according to an analysis of phonetic similarities and differences between the two languages. The length of the language articles shows a significant difference, with Tamil articles comprising 3,941,467 characters, while Kannada articles are much longer at 3,699,244 characters. Even though Tamil and Kannada have similar phonetics, their differences fall short of the cutoff point (score of 0.00), indicating that no notable differences exist in the context of the analysis. This highlights the unique yet somewhat related phonetic characteristics of these two languages. The above results are showcased in figures 4.11 and 4.12 given below.

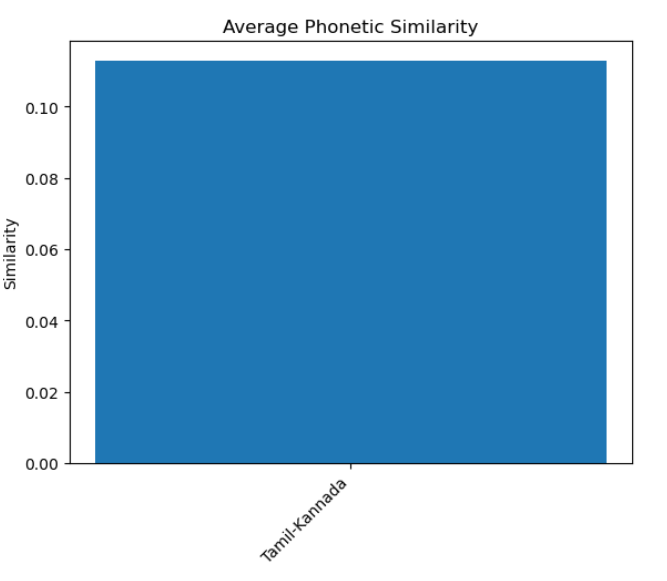
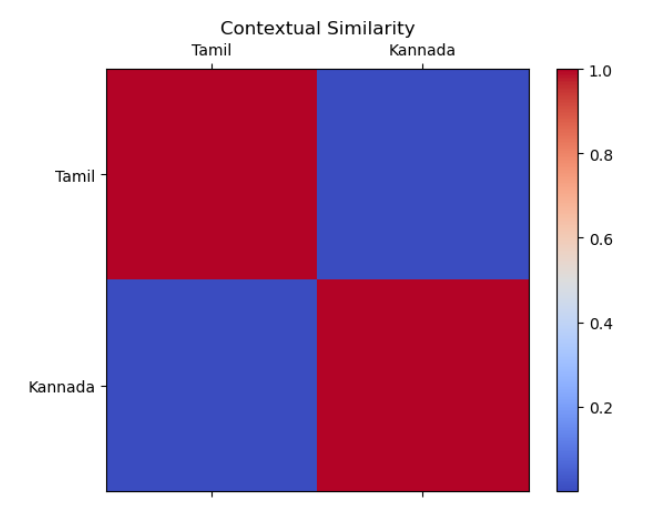


Figure 4.11: Average Phonetic Similarity Figure 4.12: Contextual Similarity

1. **Conclusions**

The aforementioned study has advanced computational linguistics significantly, especially in the fields of multilingual text and picture processing. The work highlights the efficacy of sophisticated predictive models like Random Forest and Gradient Boosting by reaching high classification accuracy in differentiating characters from dialects with similar scripts, including Gujarati and Marathi/Hindi. These developments are critical for enhancing Optical Character Recognition (OCR) systems and improving cross-language communication tools through a deeper understanding of phonetic similarities. The meticulous analysis of phonetic similarity also contributes to advancements in transliteration and translation systems, enabling more accurate and efficient cross-language interactions. The practical applications of these findings extend to the development of sophisticated OCR systems, improved information retrieval across languages, and the enhancement of translation, transliteration, and speech recognition technologies. Moreover, the study's insights offer invaluable contributions to the fields of language learning and cultural and linguistic research, aiding in the preservation and exploration of linguistic heritage.

The importance of the study's findings cannot be overstated, as they represent a leap forward in technological capabilities for handling multilingual data and provide a richer understanding of linguistic relationships. Future studies in machine learning applications for text and picture classification will be measured against this high level of classification accuracy. Beyond technological advancements, the study offers profound linguistic insights that are crucial for both theoretical and applied linguistics. Its interdisciplinary impact bridges computational linguistics and machine learning, showing how collaborative research can address complex issues involving multiple languages. The broad practical applications of this research range from enhancing everyday digital tools to contributing to specialized areas of linguistic education and research. Overall, the study propels the field forward and opens up numerous avenues for future exploration and application, emphasizing its significance in both technological and linguistic domains.

1. **References**