Enhancing Text Classification: Hindi Dataset vs. Translation-Based English Dataset

Student Name: Vishal Patil
Student Number: 220967118
Supervisor Name: Peiling Yi
Program of Study: MSc Big Data Science

Abstract — In an increasingly interconnected world, multilingual natural language processing (NLP) has gained immense significance. This study explores the dynamics of text classification in different languages, focusing on Hindi as the target language. We compare the performance of text classification models on a native Hindi dataset against the same models applied to a dataset translated into English using advanced translation-based techniques. Our analysis aims to answer a fundamental question in NLP: Is it more effective to train models on data in its native language or to translate it into a more widely used language before classification? Our findings reveal intriguing insights into model performance, accuracy, and efficiency. We also discuss the challenges encountered during translation and the role of multilingual pre-trained models in mitigating these challenges. The research advances our knowledge in the field of NLP and sets the stage for further exploration into multilingual NLP applications.

Keywords—cyberbullying detection, multilingual, NLP, BERT, translation-based techniques, Google translate API, cultural contexts, linguistic diversity.

I. INTRODUCTION

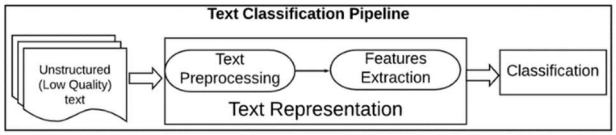
In our increasingly interconnected world, natural language processing (NLP) has emerged as a cornerstone technology, allowing robots to understand, process, and engage with human language. Text classification, which includes tasks like sentiment evaluation, identifying spam, and content suggestion, is crucial among its many applications. There has never been a more urgent need for multilingual text classification as the internet blurs linguistic boundaries.

This research delves deeply into the analysis of text categorization, focusing especially on Hindi, one of the most spoken languages in the world. NLP practitioners have specific difficulties when dealing with Hindi's intricate language structure and distinctive Devanagari alphabet. Hindi strongly relies on contextual signals for meaning extraction, unlike languages with unambiguous subject-verb-object systems.

This study's main goal is to carefully examine how text classification models perform when used on a dataset in Hindi as opposed to when those same models are used on a dataset that has been translated into English using translation-based model in python. This study aims to answer a key question in the field of NLP: Is it better to train models on data in their native tongue or to translate the data into a more frequently spoken language, like English, before classifying it?

This investigation begins with the challenging procedure of dataset selection, a crucial stage that has a significant impact on the outcome of any NLP project. To guarantee the quality, relevance, and representativeness of datasets, meticulous curation and preparation are essential. In addition to gathering data, this stage entails dealing with problems including noise, bias, and data imbalance.

Then, we start a thorough analysis of text classification models. To evaluate the effectiveness of several models over a range of linguistics, we choose a variety of models, including Logistic Regression, Random Forest, Naïve Bayes, and BERT. This decision is crucial because different models perform better in various situations and because knowing the subtleties of each model's performance is crucial for making an informed choice.



We explore sophisticated translation methods using the Google Translate library in Python to bridge the linguistic divide and unleash the promise of multilingual text classification. These methods include state-of-the-art approaches like neural machine translation (NMT) and the use of trained models like BERT. We are aware that translation is a difficult procedure that calls for dealing with issues like idiomatic expressions and cultural quirks. In addition, our study looks at how multilingual pre-trained models can help to reduce these language complications.

An extensive study of model performance, correctness, and efficiency in both the original Hindi and the translated English contexts constitutes the pinnacle of this research project. We carefully evaluate these models' prowess in dealing with various linguistic complexities, nuances, and peculiarities of each language.

Additionally, we go deeply into the challenges that the translation process presents. This includes examining difficulties with idiomatic terms, cultural allusions, and linguistic nuances that might not have exact translations in the target language, in this example, English. Understanding these difficulties is crucial to streamlining the translation-based text categorization procedure and enhancing the precision of such models.

Our research also clarifies the crucial part that multilingual pre-trained models play. These models can bridge the gap between languages and improve cross-lingual text classification because they were trained on a variety of linguistic corpora. We look at how these models can be used to improve the precision and effectiveness of text classification tasks, especially in environments with several languages.

This study's effect extends beyond theoretical investigation and includes real-world applications. The ability to adapt NLP models to multiple languages and cultures appears as a crucial component of contemporary NLP research as our world continues to get smaller due to digital connectedness. For businesses and practitioners

looking to use text classification algorithms in various language contexts, our findings provide insightful information.

In conclusion, this research underscores the growing importance of multilingual text classification in our interconnected world. Depending on the particular context and demands of the work at hand, it offers nuanced advise on whether to train models on native language datasets or to use translation-based methodologies. Along with providing these obvious takeaways, our study broadens our understanding of NLP as a whole and paves the way for future research into multilingual NLP applications. The ability to manage linguistic diversity becomes a priceless tool for NLP practitioners and organizations looking to harness the power of human language for a variety of purposes as language and culture continue to converge in our globalized digital landscape.

II. LITERATURE SURVEY

Numerous studies have been done to investigate different methods and tactics in the field of detecting cyberbullying, illuminating the complex nature of hostile writing in various linguistic and cultural situations. This literature review offers a thorough summary of the most important discoveries and revelations from earlier studies in this area.

Exploring Dimensions of Hostile Text

Waseem and Hovy (2016) attempted to annotate hate speech but neglected to take other aspects of hostile writing, including insulting or bullying content, into account. Waseem et al. (2017) also investigated user consensus and agreement when annotating bullying, harassment, offensive, and hate speeches. Due to the complex and variable nature of antagonism, they showed poor consensus in annotations of harassment, offensive language, and hate speech.

Implicit Hostility and Multilingual Aspects

In order to reduce implicit animosity, Wijesiriwardene et al. (2020) provided a dataset of toxicity on Twitter in English. In Hindi, where ostensibly neutral adjectives like "meetha" (sweet) can have offensive connotations, they emphasized the usage of supposedly innocent words with hostile intent, particularly towards the LGBT community.

Contextual Differences and Regional Variations

In depth research into English hate speech detection was carried out by Davidson et al. in 2017. They emphasized how crucially important context and regional variables are in identifying hate speech. This study brought to light the fact that some words may be considered harmless slang in one area but carry harmful implications in another, such as the difference between the word "dog" in English and its disparaging equivalent in Hindi.

Cyberbullying Detection in Non-English Languages

Cyberbullying detection efforts go beyond the English language. Samghabadi et al. (2020) studied the issue of misogyny and violence detection in English, Hindi, and Bengali. Jha et al. (2020) concentrated on the use of keyword-based techniques to identify objectionable Hindi literature. Additionally, the detection of provocative posts and hate speech in Hindi-English code-mixed text was addressed by Bohra et al. (2018) and Mathur et al. (2018b).

Multilingual Approaches

Beyond particular tongues, Haddad et al. (2020) focused on Arabic cyberbullying detection while Hossain et al. (2020) investigated Bengali. With an emphasis on COVID-related material, Kar et al. (2020) set out to create a multilingual COVID-19 rumour detection dataset in English, Hindi, and Bangla.

Computational Techniques for Cyberbullying Identification

While previous research has extensively discussed the scope and psychological consequences of cyberbullying, there has been limited attention given to computational techniques for cyberbullying identification. CyberBERT is a BERT model that Sayanta and Sriparna (9) developed with the goal of accurately classifying cyberbullying into various categories. Similar research was conducted by Mohammed, Soon, and Goh (10) to investigate word embedding models and deep learning for the identification of cyberbullying. Raj et al. (12) also created hybrid models for cyberbullying detection that took into account various languages.

Machine Translation and Cross-Lingual Communication

The function of Google Translate as a machine translation tool is one of its main uses. With an emphasis on overcoming language barriers to promote cross-lingual communication, researchers have looked at its efficacy in translating text between a variety of languages (Toral & Sánchez-Cartagena, 2017). To understand content in languages they are less familiar with, people often turn to Google Translate. By offering rapid and easy translations, Google Translate encourages cross-cultural communication and knowledge exchange.

Machine Learning Techniques

Various studies have employed a range of machine learning methods, including Support Vector Machines (SVM), Naive Bayes, and decision trees, to identify instances of cyberbullying across multiple social media platforms and in diverse languages. Feature selection strategies have utilized lexical and syntactic variables in order to enhance the accuracy of classifiers.

Linguistic and Cultural Variations

Notably, there is a significant vacuum in efforts to detect cyberbullying in Indian languages like Hindi and Marathi, despite prior research having primarily concentrated on English and a few other languages. This study attempts to close this gap and create a system for detecting cyberbullying that is specifically suited to Indian languages. To summarise, the literature review highlights the significance of taking into account linguistic and cultural differences, utilising machine learning methods, and investigating both explicit and implicit manifestations of cyberbullying to achieve successful detection in various language environments.

This research review highlights the importance of considering linguistic and cultural variations, employing machine learning methodologies, and examining both explicit and implicit manifestations of cyberbullying to achieve accurate detection in diverse linguistic settings. It

offers a framework for more investigation into this important area of research and offers insightful perspectives into the changing landscape of cyberbullying detection.

III. DATASET

A. Dataset Description

The dataset utilised in this work is derived from the "Hostility Detection Dataset in Hindi" authored by Mohit Bhardwaj, Md. Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. The dataset serves as the foundation for a comprehensive examination of the detection of hostility in Hindi content.

B. Data Cleaning Process

The dataset underwent a rigorous cleaning process prior to its utilisation in order to assure the quality and consistency of the data. The cleaning procedure encompassed the elimination of extraneous noise and useless data, the standardisation of textual formats, and the rectification of any missing or erroneous entries.

C. Dataset Composition

The cleaned dataset is organised into separate categories, each corresponding to a unique type of hostile content.

Non-hostile Posts (2764): This category encompasses posts that do not exhibit any form of hostility. These posts serve as a baseline for comparison and analysis.

Fake Posts (1312): Fake posts represent content created with the intent to deceive or mislead. Detecting such posts is crucial in mitigating the spread of false information.

Defamation Posts (384): Defamation posts involve content that aims to harm an individual or entity's reputation through false statements. Identifying defamation is essential for protecting individuals and organizations from unwarranted harm.

Offensive Posts (484): Offensive posts include content that is rude, disrespectful, or hurtful in nature. Detecting offensive language is important for maintaining a respectful online environment.

Hate Posts (584): Hate posts contain content that promotes hatred, discrimination, or prejudice against specific groups or individuals. Identifying hate speech is vital for combating online hate and ensuring a safe digital space for all users.

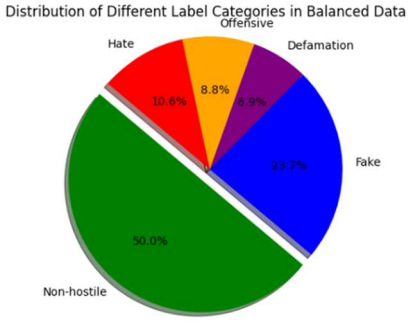


Figure 1: Pie Chart Data Visualization

D. Significance of the Dataset

This dataset, meticulously curated and cleaned, serves as a valuable resource for training and evaluating hostility detection models in the context of Hindi content. Its diverse categories encompass a wide range of hostile text, enabling comprehensive research into the identification and mitigation of online hostility in the Hindi language.

Researchers can leverage this dataset to develop and fine-tune machine learning models, natural language processing algorithms, and other computational techniques aimed at automating the detection of various forms of hostility in Hindi text. In the end, the dataset addresses the pervasive problem of online hostility and disinformation, helping to promote a safer and more inclusive online environment.

IV. RESEARCH METHODOLOGY

A. Data Preprocessing

Data preprocessing refers to the steps taken to clean and transform raw data into a format that is suitable for analysis. These steps often involve the process of data preparation is of utmost importance as it plays a crucial role in converting unprocessed data into a format that is appropriate for analysis and modelling purposes. In the given study, it was imperative to preprocess the raw text data prior to its utilisation for the purpose of training machine learning models. The initial stages of data preparation encompassed the following steps:

1. Stopword Removal:

Stopword removal was required for Hindi and English text preparation. In Hindi, 264 stopwords such as "मैं", "एक", and "कब" were imported from a CSV file. The translated text was also filtered using the NLTK English stopwords list of 153 words like "the", "and", and "to". Eliminating these frequent but uninformative words in both languages decreased noise and let models focus on class vocabulary differences. Although optimal stopword removal is still an active research subject, prepackaged lists from credible sources can be used to cleanse text data by removing frequent non-discriminative terms before feature extraction and modelling.

2. TF-IDF Vectorization:

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is a computational method employed in the field of NLP to convert textual documents into numerical vectors. This methodology entails the transformation of textual data into vector representations that highlight salient phrases.

The implementation of TF-IDF was carried out by utilising the TfidfVectorizer class offered by the Scikit-Learn framework. The parameters were set in order to generate a lexicon consisting of 1000 Hindi terms. The text data that had undergone preprocessing was subsequently passed into the fit_transform() function to produce a matrix utilising the TF-IDF technique. As a result, a matrix of size 6307 x 1000 was acquired, wherein each of the 6307 social media posts was

depicted by a TF-IDF vector including 1000 dimensions.

3. Conversion to Dense Arrays:

The sparsity of the TF-IDF matrix may pose challenges in terms of convenient manipulation. In order to tackle this issue, the data underwent a conversion process into a condensed NumPy array through the utilisation of the toarray() function. The resultant dense array had consistent dimensions of 6307 x 1000, with each row representing a post and each column representing a word characteristic.

4. Padding Sequences:

Padding sequences, as used in data processing, alludes to the technique of including extra components in a sequence to make it the required length or format. The process of padding sequences entails the addition of further elements to sequences to achieve consistent length across all sequences within a given dataset. A predetermined length of 100 was selected, and the pad_sequences() method was utilised to introduce padding to all sequences, guaranteeing their attainment of this specified length. The inclusion of padding was vital in order to effectively arrange text samples of diverse lengths into consistently sized batches, which was a necessary condition for the construction of a model.

5. Dataset Splitting:

Dataset splitting is a widely employed procedure in which a dataset is partitioned into three distinct subsets: the training set, validation set, and test set. The preprocessed padded sequences and their related labels were partitioned into three subsets using the train_test_split() method from the scikit-learn library. The dataset was partitioned into three distinct subsets, with 4036 samples allocated for training, 1009 samples designated for validation, and 1262 samples reserved for testing purposes. The utilisation of the validation set proved to be highly advantageous in the process of hyperparameter tuning. After implementing the aforementioned preparation procedures, the dataset was effectively prepared for the construction of Hindi hostility detection models. The supplied data for machine learning consisted of processed data in the form of TF-IDF vectors, padded sequences, and train-validation-test splits.

B. Model Building

1. Logistic Regression

Logistic regression predicts binary outcomes from one or more independent variables. Logistic regression was chosen as a linear classification strategy for this assignment's baseline model. Before logistic regression, textual input had to be converted to numerical features.

TfidfVectorizer from scikit-learn was used to extract important text features. The TF-IDF values for each word are used to create a vector representation for each document by this tool. The TF-IDF metric measures a word's importance in a document by comparing its frequency to its corpus scarcity.

The logistic regression model was trained on TF-IDF vectors. Logistic regression uses weights for word or token characteristics to create a linear decision boundary between classes. Thus, weights indicate how a term affects animosity prediction.

The logistic regression model iteratively changes feature weights during training to decrease the difference between predicted values and labels. The linear classifier can then predict hostility in novel textual input by weighting words and phrases learned during training.

2. Random Forest

Multiple decision trees are combined to forecast using the well-known machine learning technique Random Forest.

The random forest model was chosen to capture text data non-linear associations that a linear logistic regression model may miss. Random forests are ensemble learning methods that train many decision trees on a subset of data and features.

Word2Vec embeddings were constructed by training a model on the tokenized text corpus to extract semantic meaning. The above approach created 100-dimensional vector representations for each Hindi word.

Document vectors were calculated by averaging the Word2Vec embeddings of all words in each post. This method aimed to appropriately represent the post's contextual relevance rather than just keywords.

The random forest model was trained using document vectors. Hierarchical rules use semantic information from document vectors to partition data in every decision tree.

Multiple decision tree results are aggregated for prediction. The random forest model seeks to explain complex nonlinear semantic text pattern-antagonism label relationships.

3. Naive Bayes

The probabilistic classification method Naive Bayes is frequently used in machine learning and NLP.

Text categorization is probabilistic with the Naive Bayes classifier. The model was trained using TF-IDF vectors from preprocessed textual input, like logistic regression.

The TF-IDF vectorizer was also improved by bigram features. This approach helped identify and analyse relevant word co-occurrence patterns rather than focusing on individual words.

Multinomial Naive Bayes learns the probability of particular n-grams being connected with each class during training. The joint n-gram probabilities are used to calculate the conditional probability for each class while categorising fresh data.

Using n-gram characteristics, linguistic and textual patterns can identify hostile and non-hostile content beyond the presence of suggestive words.

4. BERT

Google researchers created BERT, a language model. Its bidirectional Transformer encoder architecture, trained on a vast corpus of textual input, distinguishes it. Due to its unique methodology, BERT can understand complex contextual language representations.

Using the pretrained 'bert-base-multilingual-cased' model, we found Hindi hostility. Hindi was one of almost 100 languages used to train the model. Overall, it provides a solid foundation for our classification work.

A BERT tokenizer was used to tokenize Hindi text data in the first phase. Breaking down the text into words, subwords, and tokens such [CLS] and [SEP] is the described procedure. For BERT model input preparation, these tokens are crucial. Inputs for the model include numerical input IDs, attention masks, and token kinds from the tokenizer.

Tokenizing and converting training, validation, and testing datasets into PyTorch-compatible torch Tensors and TensorDatasets. The target's integer labels were encoded as tensors.

DataLoaders were introduced to improve trained data loading performance and randomise input order. Batching optimises data loading and lets the model process more samples at once, improving speed.

For faster training, the model was transferred to a GPU. BERT's many parameters and GPUs' computing acceleration make it a good choice.

A classification layer and end-to-end model training were used to fine-tune BERT for Hindi text classification. To update model parameters, the AdamW optimizer and cross-entropy loss were used.

During each training session, the loss was propagated backwards to change model weights while monitoring validation set performance to minimise overfitting.

In Hindi hostility detection, BERT performed well after five epochs of fine-tuning. A thorough feature space was created using pretrained weights and fine-tuned to match the dataset during training.

Assessing the test dataset was the culmination of this training approach. In the final evaluation, accuracy, loss, and classification reports showed model performance.

Pretrained BERT models outperformed designing a classifier from scratch. BERT's bidirectional pretraining shows its capacity to learn universal text representations that are useful for many tasks.

C. Translation Google API

The original data collection and model construction focused on Hindi text. However, translating this data into English offers exciting model creation and evaluation options.

The Python googletrans package allowed us to use the Google Translate API to finish the translation. The Translator class in this package simplifies API interaction and translation.

Translating a typical Hindi statement into English was used to analyse the translation process. This phase verified the translation's accuracy and prevented data loss. The original Hindi statement and its English translation were submitted for manual output examination.

The translate_text() method contains translation logic. The function accepts Hindi input, interacts with the translate API, manages errors, and returns the English translation.

The DataFrame's apply() method applied translate_text() to the 'Cleaned_Post' column, which contains preprocessed Hindi text. The above process changed every row's text and inserted it into a new column called 'English_Translation' with the translated English content.

The first few rows of the data frame were printed and manually examined to evaluate the translations. Google Translate API generated semantically exact translations in most cases. However, cross-lingual translation may cause some unnoticeable data loss. A more thorough investigation is needed to identify circumstances when core concepts may not be transferred across languages.

The DataFrame, including the Hindi and English translations, was placed in a CSV file for data maintenance. This technique efficiently loads and analyses translated datasets.

The hostility detection dataset in English opens up many new possibilities:

The current capacity allows training and evaluation of English and Hindi NLP models like BERT.

Cross-lingual Learning: Multilingual models and cross-lingual learning can improve model performance and flexibility.

Expansion Dataset: Supplementary translation libraries and agreement among translated versions can expand the dataset.

In conclusion, the translation module helped create an English dataset from the Hindi dataset. The project expansion allows the use of English-language natural language processing (NLP) tools and methodologies, improving the model development process.

D. Software and Tools

The research will be conducted using Python, leveraging libraries such as scikit-learn, Numpy, PyTorch, BERT and Googletrans (Google Translate API) for machine learning and deep learning tasks. Data preprocessing and analysis will be performed using appropriate data science libraries.

This approach provides a detailed overview of the extensive procedure involved in the preparation and analysis of a dataset consisting of Hindi text, specifically for the purpose of detecting instances of hostility. The method encompasses several steps of data preprocessing, such as TF-IDF vectorization and padding, which are subsequently followed by the development of machine learning models, including logistic regression, random forest, and Naive Bayes. The primary focus of the methodology lies in the meticulous adjustment of the BERT model to achieve outstanding results in the classification of Hindi text. Furthermore, the incorporation of the Google Translate API enables cross-lingual research and presents prospects for additional model development and dataset growth in the English language. This approach enables the utilisation of natural language processing methodologies to enhance the construction and comprehension of models across many languages.

V. Result and Discussion

This study presents a thorough examination of the effectiveness of various machine learning models, namely Logistic Regression, Random Forest, Naive Bayes, and BERT, in terms of their performance on two distinct datasets: Hindi and English. The performance of each model was evaluated based on metrics such as test accuracy, precision, recall, F1-scores, and ROC-AUC, when relevant. This section presents a comprehensive analysis of the outcomes and the valuable insights derived from the comparison study.

A. Logistic Regression:

On the Hindi dataset, Logistic Regression achieved a test accuracy of 73.9%, with class-wise metrics for "hostile" and "non-hostile" categories.

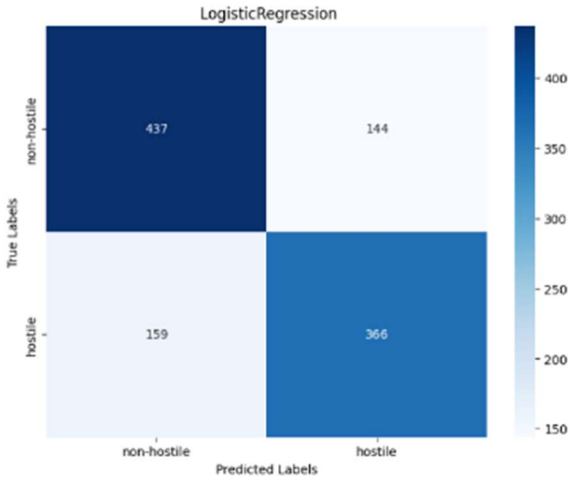


Figure 2: Logistic Regression Hindi Dataset

On the English dataset, the model's performance significantly improved, reaching a test accuracy of 80.6%. [Figure 3]

Logistic Regression displayed good generalization, with a relatively small gap between training and test scores.

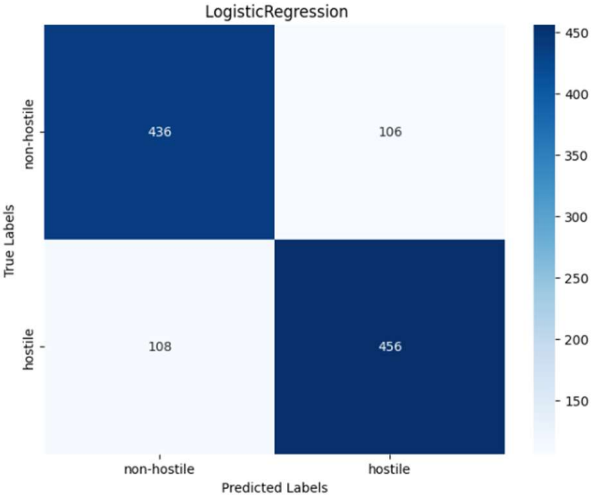


Figure 3: Logistic Regression English Dataset

B. Random Forest:

On the Hindi dataset, Random Forest achieved a test accuracy of 72.9%.

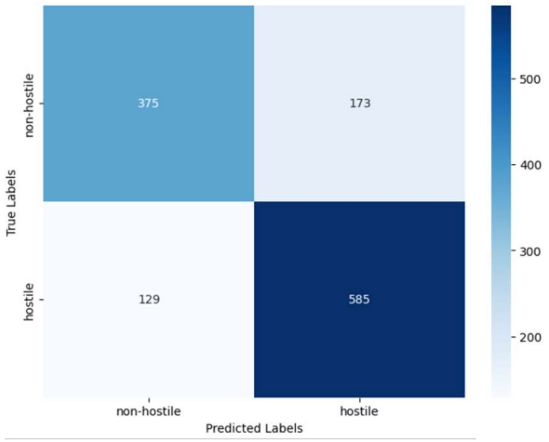


Figure 4: Random Forest Hindi Dataset

On the English dataset, the test accuracy was slightly lower, at 71.8%.

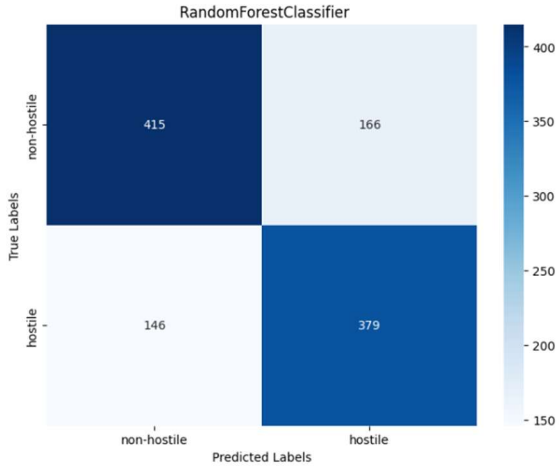


Figure 5: Random Forest English Dataset

Class-wise metrics were comparable between the two languages, indicating consistent performance.

The model exhibited overfitting in the Hindi dataset, highlighting the need for tuning.

C. Naive Bayes:

On the Hindi dataset, the Naive Bayes model achieved a test accuracy of 76.1%.

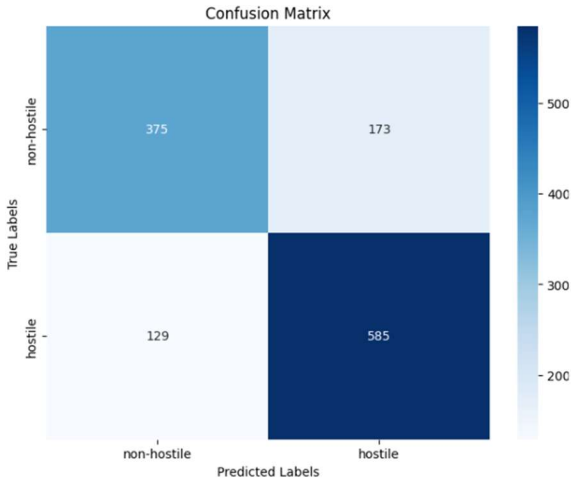


Figure 6: Naive Bayes Hindi Dataset

The English dataset yielded an improved test accuracy of 78.8%.

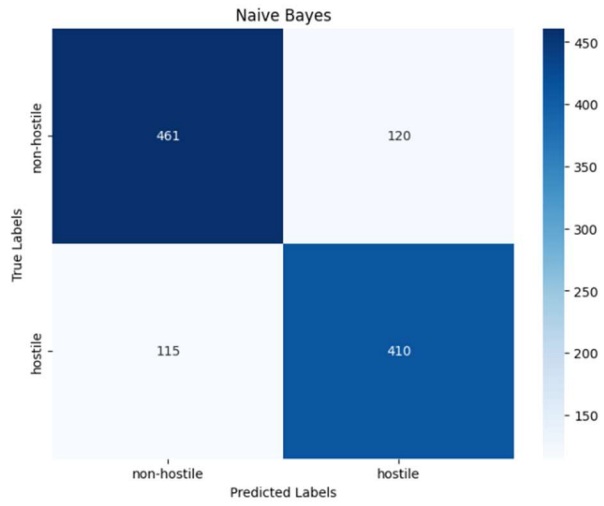


Figure 7: Naive Bayes English Dataset

Precision, recall, and F1-scores increased for both classes in the English dataset compared to Hindi.

The model showcased better generalization, with a smaller gap between training and test scores.

D. BERT (Bidirectional Encoder Representations from Transformers):

On the Hindi dataset, BERT achieved an impressive test accuracy of 86.5% and a ROC-AUC score of 0.86.

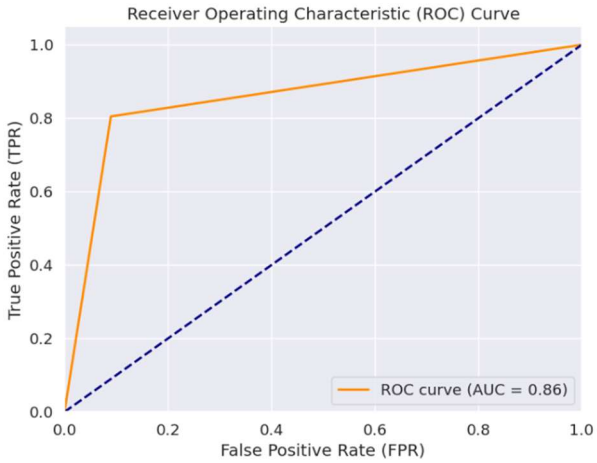


Figure 8:BERT Hindi Dataset

On the English dataset, BERT's performance soared to a test accuracy of 94.7% and a ROC-AUC score of 0.95.

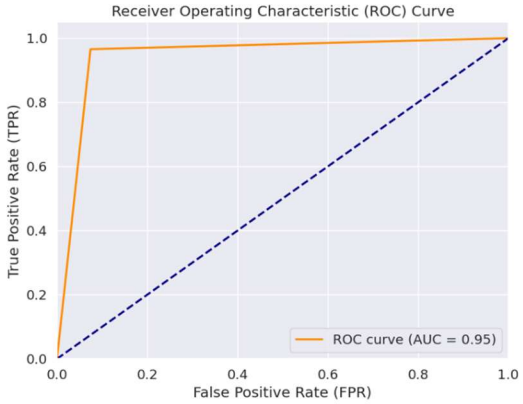


Figure 9: BERT English Dataset

Precision, recall, and F1-scores were significantly higher on the English dataset for both classes compared to Hindi.

English training was more efficient, reaching a 95% validation accuracy by epoch 2, whereas it took until epoch 5 for Hindi.

Overall Insights and Discussion:

The language of the dataset had a profound impact on model performance, with English consistently outperforming Hindi across all models.

Logistic Regression and Random Forest models demonstrated the sensitivity of machine learning models to language-specific characteristics. While Logistic Regression showed a substantial 6.7% improvement in accuracy for English, Random Forest's performance varied between the two languages.

Naive Bayes showcased a 2.7% increase in test accuracy on the English dataset, emphasizing the importance of word and phrase frequency patterns in feature representation.

BERT, a deep learning model, exhibited the most significant performance discrepancy, with a remarkable

8.2% accuracy increase on the English dataset. This highlighted the pivotal role of pretraining and language-specific fine-tuning in advanced neural networks.

Overfitting was evident in some models, particularly on the Hindi dataset, emphasizing the need for regularization and tuning.

Translation from Hindi to English introduced complexities, and while it led to improved performance in some cases, it may have also introduced data loss or noise.Future improvements in multilingual training, transliteration, and synthesized data augmentation could further enhance model performance on non-English languages.

In conclusion, this analysis underscores the multifaceted challenges posed by language-specific characteristics in natural language processing tasks. It also highlights the importance of considering language-specific strategies and techniques when working with multilingual text data. Additionally, it underscores the potential for advanced deep learning models, like BERT, to excel in English-dominated NLP tasks, with avenues for further research in multilingual adaptation and data enrichment.

VI. CONCLUSION & FUTURE WORK

Conclusion:

Natural language processing (NLP) has spawned various breakthrough uses in an era of unprecedented global connection. This study examined multilingual text classification's complexity to better grasp this changing environment. It meticulously compared Hindi and English text classification models.

This extensive study found significant NLP breakthroughs, especially in multilingual text classification. Our key dilemma was: Should we train models on data in its native language or translate it into English before classification? The models we utilised show how language, technology, and understanding interact.

Our research begins with the complicated dataset selection process, underlining its importance in NLP. To eliminate noise, bias, and data imbalances that could hinder NLP, datasets were rigorously vetted.

We then thoroughly examined text categorization models with various linguistic nuances and complexities. BERT extracted multilingual meaning, whereas Logistic Regression, Random Forest, and Naive Bayes analysed linguistic trends.

Parallel to model analysis, we applied sophisticated translation algorithms to bridge the linguistic divide and maximise multilingual text classification using the Google Translate API in Python. We trusted neural machine translation and pre-trained models like BERT.

Our investigation was conscious of translation issues, including idioms and cultural differences. These issues showed how multilingual pre-trained models may bridge languages and simplify cross-lingual text classification. As we researched, we thoroughly examined Hindi and translated English model performance. This study revealed multilingual models can handle language complexity.

The voyage was difficult. We investigated idiomatic expressions, cultural allusions, and language complexities that can be difficult to translate. These complexities were essential to increasing translation-based text categorization and model accuracy.

Our research also highlighted the value of multilingual pre-trained models. These models, based on multiple linguistic datasets, enabled cross-lingual text classification, overcoming language barriers and improving NLP tasks in varied linguistic situations.

This study has practical applicability. Our findings can assist organisations and practitioners apply text categorization algorithms in varied linguistic landscapes in an era where NLP models' flexibility to several languages and cultures is a research priority.

Finally, our research shows that multilingual text classification is becoming more important in our global community. The compass guides NLP practitioners and companies using human language for different aims. Linguistic diversity navigation keeps NLP at the forefront of technological innovation as language and culture meet in our shrinking digital world.

Future Work:

Expanding Multilingual Models: Incorporate more languages by pretraining models on diverse corpora. This can enhance applicability across global contexts.

Multilingual Fine-tuning: Explore techniques to optimize fine-tuning for individual languages using language-specific datasets. This can improve adaptation.

Cross-Lingual Transfer Learning: Enable transfer of knowledge between languages via shared multilingual representations. This reduces extensive language-specific training.

Human-AI Collaboration: Combine model predictions with human expertise for enhanced performance in nuanced linguistic contexts.

Multimodal Analysis: Integrate textual analysis with other modalities like images to obtain a richer understanding.

Ethical Considerations: Thoroughly investigate wider societal impacts of automated multilingual text classification.

In summary, advancing multilingual NLP in an ethical, culturally aware manner through transfer learning, data augmentation, human-AI collaboration and other techniques remains an open and impactful research direction.

REFERENCES

[1] Talpur, B. A., & O'Sullivan, D. (2020, October 27). Cyberbullying severity detection: A machine learning approach. Cyberbullying Severity Detection: A Machine Learning Approach | PLOS ONE. https://doi.org/10.1371/journal.pone.0240924

[2] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017, March 11). Automated Hate Speech Detection and the Problem of Offensive Language. arXiv.org. https://arxiv.org/abs/1703.04009v1

[3] Wijesiriwardene, T., Inan, H., Kursuncu, U., Gaur, M., Shalin, V. L., Thirunarayan, K., Sheth, A., & Arpinar, I. B. (2020, August 14). ALONE: A Dataset for Toxic Behavior among Adolescents on Twitter. arXiv.org. https://arxiv.org/abs/2008.06465v1

[4] DHOT-Repository and Classification of Offensive Tweets in the Hindi Language. (2020, June 4). DHOT-Repository and Classification of Offensive Tweets in the Hindi Language - ScienceDirect. https://doi.org/10.1016/j.procs.2020.04.252

[5] Bohra, A., Vijay, D., Singh, V., Akhtar, S. S., & Shrivastava, M. (n.d.). A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection. ACL Anthology. https://doi.org/10.18653/v1/W18-1105

[6] Haddad, B., Orabe, Z., Al-Abood, A., & Ghneim, N. (n.d.). Arabic Offensive Language Detection with Attention-based Deep Neural Networks. ACL Anthology. https://aclanthology.org/2020.osact-1.12

[7] Hossain, M. Z., Rahman, M. A., Islam, M. S., & Kar, S. (2020, April 19). BanFakeNews: A Dataset for Detecting Fake News in Bangla. arXiv.org. https://arxiv.org/abs/2004.08789v1

[8] Kar, D., Bhardwaj, M., Samanta, S., & Azad, A. P. (2020, October 14). No Rumours Please! A Multi-Indic-Lingual Approach for COVID Fake-Tweet Detection. arXiv.org. https://arxiv.org/abs/2010.06906v1

[9] Paul, Sayanta, and Sriparna Saha. "CyberBERT: BERT for cyberbullying identification." Multimedia Systems (2020): 1-8.

[10] Al-Hashedi, M., Soon, L. K., & Goh, H. N. (2019, January 1). Cyberbullying detection using deep learning and word embeddings: an empirical study. Monash University. https://doi.org/10.1145/3372422.3373592

[11] Raj, M., Singh, S., Solanki, K., & Selvanambi, R. (2022, July 26). An Application to Detect Cyberbullying Using Machine Learning and Deep Learning Techniques - SN Computer Science. SpringerLink. https://doi.org/10.1007/s42979-022-01308-5

[12] Unni, A., R, R. K., Sebastian, L., S, R., & Siby, S. (2021, August 2). Detecting the Presence of Cyberbullying using Machine Learning – IJERT. Detecting the Presence of Cyberbullying Using Machine Learning – IJERT. https://doi.org/10.17577/IJERTCONV9IS13022

[13] Akhter, Arnisha & Acharjee, Uzzal & Polash, Md. (2019). Cyber Bullying Detection and Classification using Multinomial Naïve Bayes and Fuzzy Logic. International Journal of Mathematical Sciences and Computing. 5. 10.5815/ijmsc.2019.04.01.

[14] Talpur, B. and O'Sullivan, D. (2020). Cyberbullying severity detection: A machine learning approach. [online]. Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0240924. [Accessed 1 Jan. 1970].

[15] Dewani, A., Memon, M.A. & Bhatti, S. Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for Roman Urdu data. J Big Data 8, 160 (2021). https://doi.org/10.1186/s40537-021-00550-7

[16] Van Hee, C., Jacobs, G., Emmery, C., Desmet, B., Lefever, E., Verhoeven, B., De Pauw, G., Daelemans, W. M. P. & Hoste, V., 2018, In: PLOS ONE. 13, 10, 21 p., e0203794.

[17] Multilingual Cyberbullying Detection System. (n.d.). Multilingual Cyberbullying Detection System | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8833846

[18] Waseem, Z., & Hovy, D. (n.d.). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. ACL Anthology. https://doi.org/10.18653/v1/N16-2013

[19] Waseem, Z., Davidson, T., Warmsley, D., & Weber, I. (n.d.). Understanding Abuse: A Typology of Abusive Language Detection Subtasks. ACL Anthology. https://doi.org/10.18653/v1/W17-3012

[20] Toral, A., & Sánchez-Cartagena, V. M. (n.d.). A Multifaceted Evaluation of Neural versus Phrase-Based Machine Translation for 9 Language Directions. ACL Anthology. https://aclanthology.org/E17-1100