Dharwad

ज्ञानेन विकासः

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Machine Learning Project:

Hate and Fake Detection in Multilingual Social Media Text

Data Science And Artificial Intelligence

Team Members:

A . Saras Chandrika (21bds003)

CH . Srinivas sai (21bds012)

K. Sai Kartheek Reddy (21bds027)

R . Vinay kumar (21bds056)

Date: October 17, 2024

Abstract

The aim of this research is to create a sophisticated machine learning model that can identify hate speech and fake news in social media posts. A new strategy that makes use of a Multi-Tasking Model has been implemented to improve the model's functionality. To identify and classify potentially dangerous content, the system combines techniques from transliteration, translation, embedding extraction, and clustering.

IndicXlit and IndicTrans2, two state-of-the-art transliteration and translation technologies, are used by the project to do this. The method of embedding extraction entails the utilization of multiple advanced language models, such as BERT, MBert, Albert, Electra, and XLM. Notably, an input layer that pulls embeddings from several models is a part of the Multi-Tasking Model implementation process. After that, these embeddings are sent into shared layers, one set of which is used to detect fraudulent content.whereas a different group focuses on identifying hate speech.

In addition, the model makes use of clustering algorithms in order to find cluster centers and construct clusters based on the collected embeddings. The distances between every data point and the cluster centers are calculated and included as extra features to the embeddings. This novel method improves the model's capacity to distinguish between hate speech and fake news, two categories of damaging information. The effectiveness of the proposed method is evaluated using a benchmark dataset, suggesting that it has the ability to identify and remove harmful content from social media networks.

1 Introduction

In today's world, almost everyone has a smartphone and can easily go online. Especially young people are curious about the latest technologies and very active on social media, where they connect with different people and share their thoughts. While this has many positive aspects, there are also some problems.Some people misuse social media, claiming they have the right to free speech, and use it to post private or personal information about others. Additionally, some individuals use social media to spread hate towards others. This negative side of technology presents a challenge.Moreover, some people also misuse social media to spread fake news that leads to hate speech. For instance, during the emergence of Covid-19, many people trolled China, even though the true origin of the virus was uncertain. This example illustrates how false information can lead to hate.Detecting hate speech on the internet and social media has become crucial but difficult. There is a vast amount of online data, requiring significant computer power to sift through it all. What makes it even more important is that many children use the internet and social media, making it essential to protect them from harmful and hateful content.

Our project aims to address this challenge. We understand that as technology advances, so do the risks of its misuse. The sheer volume of online data makes it challenging to identify and stop hate speech effectively. Furthermore, with more young people engaging online, it becomes a top priority to protect them from harmful content. We intend to develop methods and tools to detect and prevent hate speech on social media and the internet. By doing so, we hope to contribute to creating a safer online environment, especially for the younger generation. Our project recognizes the benefits of technology while also acknowledging the need to address its darker side to ensure a positive and secure online experience for everyone.This project aims to address this challenge by implementing a comprehensive approach that involves translation, transliteration, and advanced machine learning techniques to analyze text data. The initial phase of our project involves collecting text data from social media sources, primarily Twitter and Youtube. To overcome language barriers, we utilize IndicTrans2 for translation and IndicXlit for transliteration, ensuring that the text is both comprehensible and maintains its original context. The translated and transliterated text is then preprocessed to ensure optimal quality for subsequent analysis. Following preprocessing, embeddings are extracted for both English and Hindi text using techniques such as BERT, MBert, Albert, Electra, and XLM. These embeddings serve as essential features for the subsequent stages of our model. Our model architecture incorporates a shared layer for simultaneous processing of hate and fake content detection. This layered approach allows for specialized computation in detecting hate speech and fake content, which are then combined to yield comprehensive results. To enhance the model's capabilities, we extend our analysis to a different dataset, applying various transliteration and translation techniques and extracting embeddings using different models. Clusters are formed based on these embeddings, with cluster centers identified for each group. This clustering approach aids in understanding the underlying patterns and relationships within the data. To enrich the feature set, distances from the cluster centers are computed for each record in the dataset. These distances are incorporated into the embeddings as additional features, resulting in an augmented feature set for subsequent analysis. This process is repeated with variations, including normalizing distances using activation functions such as GELU. The ultimate objective is to compare the performance of the model with and without these additional features, providing insights into the effectiveness of incorporating transliteration, translation, and clustering techniques in enhancing the detection of hate speech and fake content in multilingual social media data. This project not only contributes to the advancement of natural language processing techniques but also addresses the growing need for effective content moderation in diverse linguistic landscapes on online platforms.

By following the steps outlined in this process, we achieved noteworthy results. Specifically, in the case of detecting hate speech in Hindi data, obtained through the transliteration of text using the IndicXlit model from AI4Bharat, our F1-Score showed significant improvement. Without the inclusion of distances, the best F1-Score achieved using the Mbert model was 81.97. However, after incorporating normalized distances as features, the result increased substantially to 97.15. This enhancement was observed using Mbert embeddings. For the classification of fake content, the initial F1-Score was 70.16, and it rose to 73.11 after adding normalized distances as features, achieved using the Electra model and huber-loss. In the case of English data, obtained by translating the Hindi text using IndicTrans2, we achieved the highest F1-Score for hate speech as 77.99 with the XLM-roberta model without the inclusion of distances. After adding normalized distances as features with XLM-Roberta, the F1-Score improved further to 78.75. For fake content classification, the initial F1-Score was 72.96, and it increased to 76.21 after including normalized distances as features, using the XLM-Roberta model and huber-loss. These results demonstrate the effectiveness of our approach in both Hindi and English data for detecting hate speech and classifying fake content.

2 Background

As we embark on the creation of a groundbreaking Multi-Tasking Model for detecting both Hate and Fake instances in sentences, it is crucial to delve into existing literature. Since no prior models have specifically addressed our focus, we approach this task by examining insights from three key areas: Multi-Task Learning (MTL), Hate Speech Detection, and Fake News Detection.

Multi-Task Learning (MTL) In the article ”Multitask Learning” and ”An overview of multi-task learning” by Yu Zhang and Qiang Yang, MTL is presented as a robust approach to improve generalization by simultaneously learning related tasks. The essence lies in sharing information among tasks during training, offering flexibility and adaptability across different learning algorithms. The research emphasizes the potential of MTL in diverse domains, demonstrating its effectiveness and applicability.

Hate Speech Detection Our exploration extends to the paper ”Hate me, hate me not: Hate speech detection on Facebook.” This work delves into the challenges posed by harmful content on Social Network Sites (SNSs), emphasizing platforms like Facebook. The study introduces a Hate Speech Classifier for the Italian language, focusing on content classification without resorting to censorship. Contributions include classifying public page content, detecting violent discussions, and introducing a taxonomy for hate categories.

Fake News Detection In the paper ”Learning From Yourself: A Self-Distillation Method For Fake Speech Detection,” a novel self-distillation method for Fake Speech Detection (FSD) is proposed. This method enhances detection performance without increasing model complexity by utilizing a teacher-student model. The deepest network instructs shallower networks, leading to significant improvements in capturing fine-grained information essential for FSD.

Synthesis and Significance In our pursuit of developing a comprehensive Multi-Tasking Model for Hate and Fake detection, we draw on insights from the works of Yu Zhang and Qiang Yang, the examination of hate speech on Facebook, and an innovative self-distillation method for fake speech detection. The literature highlights the potential of MTL, the challenges of hate speech on social platforms, and inventive methods for enhancing fake news detection. Our project aims to fill a critical gap by leveraging these insights to create a model that addresses the unique challenges posed by both hate and fake instances in text-based content.

3 Model Architecture

3.1 Pre-Processing Steps

For our machine learning project, we utilized Code-Mixed Hindi data sourced primarily from Twitter. In the initial phase, our focus was on pre-processing the text data to enhance its quality and suitability for analysis. Pre-processing involves several crucial steps that refine raw data, making it more amenable for machine learning algorithms.In the context of our project, the first step in pre-processing was the removal of symbols and punctuation from the Code-Mixed Hindi text. This step is essential to streamline the data and eliminate unnecessary characters that could potentially introduce noise or hinder the performance of machine learning models. Additionally, we addressed the presence of emojis by converting them into equivalent text representations. This conversion aids in maintaining the semantic meaning of the emojis while ensuring a consistent and standardized format for further analysis.

Effective pre-processing lays the foundation for the success of machine learning models, as it enables them to better understand and extract meaningful patterns from the data. By systematically refining the Code-Mixed Hindi text data through these pre-processing steps, we aimed to enhance the overall quality and relevance of the information for subsequent stages of our project.

3.2 Transliteration and Translation

In this phase, we grappled with the challenge of working with Code-Mixed Hindi text, commonly referred to as Hinglish. Since there were no specific models trained for Code-Mixed Hindi, and many existing models excel in processing monolingual text, we had to find a solution to make our data more accessible for machine learning algorithms.

To overcome this hurdle, we employed the Ai4Bharat Model called IndicXlit for transliterating the Code-Mixed Hindi text into standard Hindi. Transliteration involves converting text from one script to another, and in our case, it transformed the Hinglish text into a more homogeneous Hindi format. This step was crucial for ensuring that our data aligns with available models and can be effectively utilized in subsequent analysis.

Subsequently, we further expanded the versatility of our data by translating the entirely transliterated Hindi text into English. We utilized the AI4Bharat model known as IndicTrans2 for this task. Since English is widely used globally and many pre-trained models are optimized for English data, translating our text allowed us to tap into a broader range of models and resources. This strategic move aimed to maximize the effectiveness of our machine learning project by leveraging the strengths of models trained on English data.

3.3 Extraction of Embeddings

As mentioned earlier, we performed transliteration and translation to make our data more accessible for analysis. Now, to effectively handle this multilingual data, we employed various models for extracting embeddings. Embeddings capture the contextual information within the text, providing a rich representation for machine learning models.

For the English text, we utilized models such as BERT, XLM-Roberta, ALBERT, BART, and Electra to extract embeddings. These models are renowned for their ability to understand and represent the intricate contextual nuances present in the English language. Similarly, for the Hindi data, we employed MBERT, XLM-Roberta, ALBERT, BART, and Electra models. These models are specifically tailored to comprehend the complexities of the Hindi language, ensuring that our embeddings capture the relevant information for meaningful analysis.

3.3.1 About the Models that we used

BERT (Bidirectional Encoder Representations from Transformers)

- Trained on a massive amount of data, including BookCorpus and English Wikipedia, with a focus on bidirectional understanding.
- Employs a transformer architecture, considering both left and right context in all layers, allowing it to capture intricate text relationships.
- Known for its contextual understanding, making it effective for tasks like question answering, sentiment analysis, and text classification.
- Extracts embeddings by capturing the nuances of word relationships and contextual information within the given text.
- Employs attention mechanisms to attend to all parts of the input sequence simultaneously, enabling a thorough understanding of context.

XLM-Roberta (Cross-lingual Language Model - Roberta)

- Developed to understand and represent languages across different language families, making it highly suitable for multilingual applications.
- Trained on vast multilingual corpora, enabling it to comprehend and generate embeddings for a wide range of languages.
- Utilizes a robust transformer-based architecture for effective language representation, capturing cross-lingual nuances.
- Extracts embeddings that encapsulate cross-lingual information and maintain consistency across different languages.
- Demonstrates proficiency in handling diverse linguistic contexts and exhibiting strong performance on cross-lingual tasks.

ALBERT (A Lite BERT)

- A scaled-down version of BERT designed for improved computational efficiency while retaining effectiveness.
- Achieves efficiency by significantly reducing the number of parameters in the model architecture.
- Trained on large datasets, incorporating a self-supervised learning approach to understand contextual relationships.
- Extracts embeddings by capturing intricate details in the text, making it suitable for various natural language processing tasks.
- Maintains high performance while being more resource-efficient compared to larger models.

BART (BART: Denoising Sequence-to-Sequence Pre-training)

- A sequence-to-sequence model specializing in tasks like summarization and language generation.
- Pre-trained on a denoising objective, learning to reconstruct a corrupted input, enhancing its understanding of context.
- Extracts embeddings that encapsulate the essential information for summarization and other sequence-to-sequence tasks.
- Demonstrates proficiency in generating coherent and contextually rich language output.
- Ideal for applications requiring a comprehensive understanding of textual context and meaningful sequence generation.

Electra

- Known for its efficiency in training and outperforms BERT on various downstream tasks.
- Utilizes a more effective pre-training approach, replacing certain words in the input with plausible alternatives.
- Extracts embeddings by focusing on identifying replaced tokens, improving contextual understanding.
- Trained on large corpora, enabling it to capture intricate relationships within the text efficiently.
- Shows effectiveness in various natural language processing tasks, showcasing its versatility and efficiency.

3.4 Input Layer

In the Input Layer, the model processes input embeddings of text data, obtained after the translation, transliteration, and preprocessing stages. The embeddings serve as the foundation for subsequent layers in the model. We pass the embeddings through different models to extract contextualized representations for the text. Specifically, we utilize distinct models for English and Hindi text.

3.5 Shared Layers

Two shared layers are employed for processing the input embeddings separately for Fake Detection and Hate Detection. The model architecture is designed to share certain layers for efficient feature extraction and representation learning across both detection tasks.

3.5.1 Fake Detection Branch

The Fake Detection branch of the model is responsible for discerning the presence of fake content in the processed text data. This branch is characterized by the following layers:

- **Shared Layers:** The initial layers of the Fake Detection branch are shared with the Hate Detection branch. These layers collectively contribute to the joint processing of input embeddings, allowing the model to capture relevant features for both tasks simultaneously.
- **Dense Layer (128 units, ReLU activation):** Following the shared layers, the Fake Detection branch incorporates a dense layer with 128 units and a Rectified Linear Unit (ReLU) activation function. This layer plays a crucial role in feature extraction and non-linear transformations.
- **Output Dense Layer (1 unit):** The final layer of the Fake Detection branch is a dense layer with a single unit, providing the predicted value for fake detection. The linear activation function in this layer produces continuous output values, allowing for the interpretation of the model's confidence in the prediction.

3.5.2 Hate Detection Branch

The Hate Detection branch mirrors the architecture of the Fake Detection branch, utilizing shared layers for initial processing and incorporating task-specific layers for hate detection:

- **Shared Layers:** Similar to the Fake Detection branch, the Hate Detection branch starts with shared layers that jointly process the input embeddings. These layers facilitate the extraction of relevant features for both Fake and Hate Detection tasks.
- **Dense Layer (128 units, ReLU activation):** Following the shared layers, the Hate Detection branch integrates a dense layer with 128 units and a Rectified Linear Unit (ReLU) activation function. This layer is instrumental in capturing task-specific features related to hate speech in the processed text data.
- **Output Dense Layer (1 unit):** The final layer of the Hate Detection branch is a dense layer with a single unit, serving as the output for hate detection. The linear activation function in this layer produces continuous output values, enabling the model to provide a quantitative prediction regarding the presence of hate speech in the input data.

This dual-branch architecture allows the model to effectively handle the distinct tasks of Fake Detection and Hate Detection, leveraging shared layers for common feature extraction and dedicated branches for task-specific learning.

3.6 Model Output

The model produces two outputs — one for fake detection and another for hate detection — each providing a binary prediction. The architecture follows a multi-task approach with shared layers for joint processing of input embeddings, enabling the model to simultaneously perform Fake Detection and Hate Detection.

3.6.1 Fake Detection Output

The Fake Detection branch of the model is responsible for predicting the presence of fake content in the processed text data. The output for Fake Detection is derived from the corresponding dense layer with a linear activation function. The predicted values are continuous, and a threshold of 0.5 is applied to obtain binary predictions.

3.6.2 Hate Detection Output

In a similar vein, the Hate Detection branch looks for hate speech in the text that has been processed. Through a dedicated dense layer with a linear activation function, the output for hate detection is produced. Similar to Fake Detection, the model generates continuous predictions, and a 0.5 threshold is applied to produce a binary classification.

The model goes through several epochs of training, honing its capacity to predict outcomes accurately for both tasks. Using the various loss functions and Adam optimizers, the model parameters are optimized during the training process. At every epoch on the test set, the model's performance is assessed, and metrics like accuracy, precision, recall, and F1 score are calculated for both hate and fake detection.

3.7 Loss Function and Optimization

The model utilizes the different loss functions and the Adam optimizer for training. The choice of different loss functions is to test the models performace on different loss functions and to find which is suitable for multi-task learning scenarios, providing a balanced measure for the model's performance across both Fake Detection and Hate Detection tasks. The Adam optimizer is employed for efficient parameter optimization during training.

3.8 Training Process

The training process involves multiple epochs, with the model iterating through batches of the training dataset using the custom `MultiTaskDataset` class in the code. This custom dataset class ensures that the input embeddings, along with corresponding labels for Fake Detection and Hate Detection, are appropriately processed in batches to facilitate efficient model learning.

3.9 Evaluation

After the training phase, the model undergoes evaluation on a separate test dataset. Various metrics, including accuracy, precision, recall, and F1-score, are computed for both Fake Detection and Hate Detection tasks. This comprehensive evaluation provides a thorough assessment of the model's ability to make accurate predictions across multiple tasks.

3.9.1 Performance Tracking

Throughout the training epochs, the script actively monitors the highest F1-scores for both Hate and Fake Detection. This tracking system is essential for determining the model's optimal performance, providing insights into the moments at which the model performs at its best for every detection task.

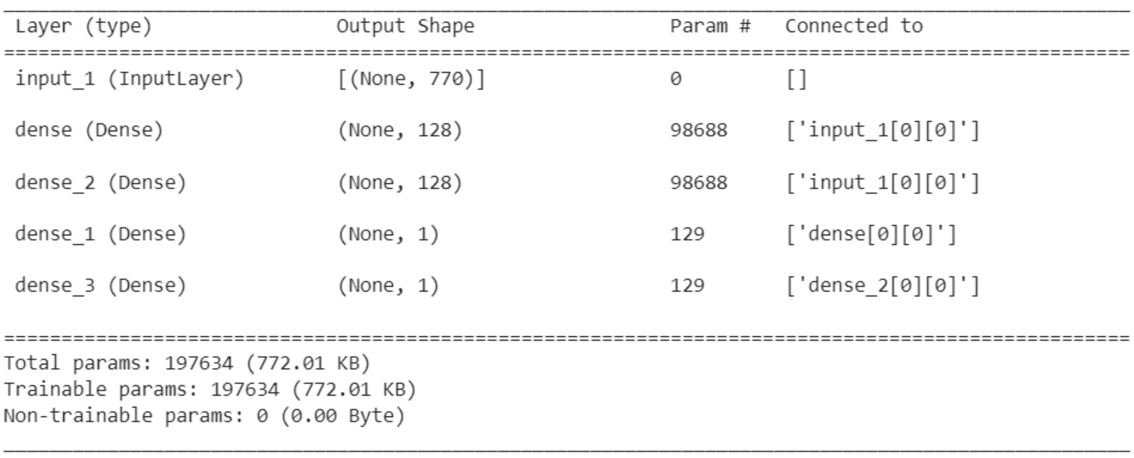
Based on these monitored metrics, the model's overall efficacy is evaluated, offering a thorough understanding of its capacity to distinguish between authentic and fraudulent content and to locate instances of hate speech within the processed text data.

3.9.2 Visualization

The script uses matplotlib for the metrics visualization in order to provide a visual depiction of the training progress. Plots that display the accuracy and F1-score over epochs give a dynamic picture of how the model performs as it is being trained. Understanding the convergence and stability of the model across the designated tasks is made easier with the help of these visualizations.

3.10 Additional Features

Additional features from clustering analysis are incorporated into the code to improve the model's performance. To enhance the input embeddings and create a more comprehensive representation of the input data, distances derived from clustering analysis are incorporated as extra features. Interestingly, various embedding models are used for clustering and then feature extraction, such as BERT, mBERT, Albert, Electra, and XLM. By using a variety of contextual embeddings, this diversified approach guarantees that the model gains an advantage over other models, improving its capacity to identify subtle patterns in the input text data.



```
__________________________________________________________________________________________________
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 input_1 (InputLayer)        [(None, 770)]                0         []

 dense (Dense)               (None, 128)                  98688     ['input_1[0][0]']

 dense_2 (Dense)             (None, 128)                  98688     ['input_1[0][0]']

 dense_1 (Dense)             (None, 1)                    129       ['dense[0][0]']

 dense_3 (Dense)             (None, 1)                    129       ['dense_2[0][0]']

==================================================================================================
Total params: 197634 (772.01 KB)
Trainable params: 197634 (772.01 KB)
Non-trainable params: 0 (0.00 Byte)
__________________________________________________________________________________________________
```

4 Conclusion

In summation, this project signifies a quantum leap in the analysis of multilingual social media content. The intricate process of translation, transliteration, and embedding extraction, coupled with a novel model architecture, has resulted in commendable success in Hate and Fake content identification. The robust F1-scores stand as a testament to the model's accuracy, while the exploration of clustering and distance-based feature engineering unveils a deeper layer of understanding within the data. As natural language processing continues its evolutionary journey, this project serves as a beacon, illuminating the path forward and highlighting the adaptability and efficacy of innovative methodologies in deciphering the complexities of digital communication.

5 Future Scope

In our current model, we have successfully developed a Multi-Tasking Model with shared layers. In one layer, the embeddings, along with Hate Labels, undergo processing, while in another layer, the embeddings, along with Fake Labels, undergo a similar process. The ultimate result is a combined output that reflects the integrated processing of both Hate and Fake Labels.

However, we envision expanding this project to incorporate additional dimensions. Specifically, we aim to introduce two new aspects: Target (categorized into Individual, Group/Organization, and Religion) and Severity (classified as Low, Medium, and High). This expansion will necessitate the design of an updated architecture to accommodate these new elements.

In the enhanced architecture, each layer will be dedicated to processing specific information related to Hate and Fake Labels, along with the newly introduced dimensions of Target and Severity. This comprehensive approach will allow our model to not only discern between Hate and Fake instances but also categorize targets into various groups and assess the severity of the identified instances.

By incorporating these additional dimensions, we anticipate our Multi-Tasking Model will achieve a more nuanced understanding of the input data. This heightened level of sophistication will enable the model to provide more detailed and contextually relevant outputs, ultimately enhancing its effectiveness in detecting and categorizing instances of Hate and Fake content.

So, our current Multi-Tasking Model, with its shared layers processing Hate and Fake Labels, lays the foundation for an expanded architecture. This expansion will involve the integration of Target and Severity dimensions, allowing for a more nuanced and comprehensive analysis of input data. The resulting model is poised to offer improved detection capabilities and a more sophisticated understanding of the nature and impact of Hate and Fake instances in textual content.