Counterhate Arguments: Countering the hate speech on social media

Swabhi Papneja
George Mason University
Fairfax, VA, USA
spapneja@gmu.edu

Sweta Pati
George Mason University
Fairfax, VA, USA
spati@gmu.edu

1 Introduction

1.1 Task / Research Question Description

In this paper, we address the critical task of identifying and validating authentic counterhate arguments in response to online hate speech targeting individuals. The central research task here is that, we propose a methodology to effectively identify the authenticity of counterhate arguments and its specificity to the individual of interest from a corpus derived from real-world data.

Our approach centers on the potential to leverage existing online content to find potential counterhate responses that are not only relevant and on-point but also grounded in factual, testimonial, or statistical evidence. It is also distinguished by its focus on authenticity—prioritizing responses that engage directly with the content and context of the hate speech rather than relying on generic or artificially generated replies.

The significance of this task lies in its direct application to improving the mechanisms for combating online hate. By developing a method to systematically extract and assess the relevance and effectiveness of counterhate arguments, this paper aims to contribute to the broader efforts of creating safer and more respectful online communication environments. Our goal is to provide tools that not only counteract the spread of hate but also promote a culture of positive and respectful online interactions.

1.2 Motivation and Limitations of existing work

In this digital age, platforms like Twitter and Facebook are not just tools for connectivity but pivotal in spreading information rapidly, with 72% of U.S. adults engaged on at least one social media site (Auxier and Anderson, 2021). They broadcast critical updates rapidly and even catalyze social movements. However, the same speed of reach also facilitate the rapid spread of harmful content, particularly hate speech. About two-thirds of Americans (64%) believe that social media platforms have a mostly negative effect on progress of the country, with significant concerns centered around hate and harassment highlighted by 16% of the respondents (Auxier, 2020). Social media platforms are grappling with the challenge of hate speech- a significant volume of content shared daily on Twitter estimates to send out 500 million tweets per day, equating the content volume to that of the New York Times over 182 years (Yaraghi, 2019). This massive scale presents challenges in monitoring and managing content effectively. This dual-use of social media drives as the the motivation behind this study, driven by the necessity to mitigate these negative impacts by authentic counterhate identification through contextually and factually appropriate responses.

Previous Efforts and Their Limitations: Previous efforts are limited to counterhate to fight against hateful content toward groups and not individuals. Identifying hateful content in user-generated content has received substantial attention in recent years (Fortuna and Nunes, 2018). Prior research on detection includes several datasets and models for hate detection conducted on several social media platforms such as Reddit (Qian et al., 2019). Previous efforts to counteract hate speech have been grouped into two primary

categories: detection and generation. The detection approach focuses on identifying text that counters hate, such as the methodology used by (Mathew et al., 2020) detecting counterhate in responses to hateful tweets through a simple pattern: I hate "group" . (He et al., 2021) take a 42 keyword-hashtag based approach to identify hate and counterhate in the context of COVID-19 related tweets. Most of the work primarily focuses on detecting hate speech and generating synthetic counterhate responses that also may hallucinate unsupported arguments. These efforts have laid a foundational understanding, but often result in responses that are either too generic or fail to address the specific claims made in hate speeches towards a specific individual. The reliance on generated content has led to the creation of responses that lack authenticity, reducing the likelihood of positive impact. Along with this all previous works are limited to fight against hateful content toward groups whereas we explore counterhate for hateful content toward individuals.

1.3 Proposed Approach

Our paper follows a methodological approach that clearly defines counter-hate arguments as something that counters hateful comments against a specific individual. By focusing on responses that are contextually grounded and directly relevant to the specific instances of hate speech, our study aims to enhance the effectiveness and authenticity of counterhate measures rather than generating synthetic counterhate responses that are generic and refer to a community or group. Let's consider an example from the research paper for a better understanding. If a hateful comment says "bad at acting", then an argument simply containing positive words like "beautiful" and "amazing" will not be directly considered as a counter-hate argument. The argument should address the hateful segment mentioned in the hateful comment or tweet with logic and reason. We know that the hate speech present online is very complex by nature. The model used advanced NLP techniques like like RoBERTa and LongFormer to handle hateful speech and potentially counter hate data. The model will handle the data based on its granularity, dividing the data into two levels - paragraphs and articles. After preparing the data after data preprocessing and tokenization using relevant tokenizers, we have designed several experiment combinations based on different input features to verify the results published in the paper. After training and testing the model on the data given by the authors, we then used different evaluation metrics to verify if the results matched the ones published in the research paper.

This approach not only addresses the shortcomings of prior models by adding depth and relevance to the responses but also contributes significantly to the ongoing efforts against online hate speech. We expect this model to support the creation of better NLP models for monitoring and managing content on vast social platforms.

Addressing Prior Shortcomings: One significant limitation in earlier works is their limited engagement with the content of hate speech beyond surface-level features. Our research directly engages with the content and context of each instance, using sophisticated NLP techniques to analyze and match the most appropriate counter-response.

1.4 Likely challenges and mitigations

As we already know, that language is difficult. There are certain linguistic aspects that we need to take care of while working on text data, especially the ones extracted from social media platforms. We have sarcasm, irony and various other expressions to deal with, that can be difficult to detect for machine learning models. This might lead to a misclassification of hateful tweets and potential counterhate texts containing sarcastic remarks. Another notable challenge is the highly imbalanced data. This imbalance signifies that there are a lot of hateful comments that are being generated in millions of numbers at this very instant. However, when it comes to authentic counter-hate arguments, they will be rare to find in comparison to hateful speech. Apart from this, the model might not perform so well on unseen data including individuals not included in the training data.

If something turns out to be more difficult to handle than expected, we can experiment with the dataset and apply some additional

techniques while preparing the data. We can perform data augmentation to artificially expand the dataset so increase the size of the data and ensure the model's validity. If the model is trained on a diverse validation set, there are higher chances that the model will perform better on different subsets of the unseen data. We can also plan to include data balancing techniques like SMOTE if we are not able to reproduce the published results. We can also plan to use different hyperparameter choices to experiment with the results. Still, if we observe that the model does not perform as expected. we will try to alternative model architectures, enhance data preprocessing and try to work with different feature inputs.

2 Related Work

The landscape of hate speech detection and mitigation is ever-evolving. Basile et al. (2019) put forth a method to differentiate hate speech targets, distinguishing between individuals and groups, with a further subdivision into specific categories like immigrant women. Their approach is instrumental in understanding the granularity of hate targets. Our work is different from theirs as we are focusing on authentic counterhate identification towards a particular individual, and not a group.

Extending beyond detection, Qian et al. (2019) curated a dataset to foster the generation of counter-hate narratives, opening avenues for proactive hate speech mitigation. This is complemented by the work of Chung et al. (2019), who provide a dataset comprising anti-Muslim hate and corresponding counter-responses, albeit the content's synthetic nature raises questions about its real-world applicability. Our work diverges by focusing on the retrieval of authentic counterhate arguments directly from online articles, which has been less explored in existing literature.

We aim to address the gap identified by Chung et al. (2021) regarding the generation of factually accurate counterhate, by leveraging existing credible online discourses to combat hate towards individuals, a relatively uncharted area in contrast to the group-focused counterhate efforts seen in prior studies.

3 Experiments

3.1 Datasets

In this study, we have utilized the dataset specifically curated for the task of identifying authentic counterhate arguments. The dataset comprises of two CSV files:

1. **articles.csv**
   (a) 'id_str': Unique identifier ID
   (b) 'tweet': Hateful tweet content
   (c) 'article': Text articles taken from online articles related to the content of the hateful tweet.
   (d) 'label': 0 or 1 value, indicating whether a paragraph from the article is considered to be a valid counter-hate argument (1 for yes, 0 for no).

2. **paragraph.csv**
   (a) 'id_str': Unique identifier ID
   (b) 'tweet': Hateful tweet content
   (c) 'paragraph': Sentences from articles that can be potential counter-hate arguments
   (d) 'label': 0 or 1 value, indicating whether this paragraph is a valid counter-hate argument (1 for yes, 0 for no).

The model takes into account not only the contents of a hateful tweet but also contextual information from related articles to determine an appropriate counterhate response.

Moving on to the size of our dataset, we have a corpus of 250 hateful tweets targeted towards 50 public figures, amounting to an average of 5 tweets per individual. There is a collection of 2,500 candidate articles which were sourced for potential counterhate arguments. We have a set of annotations on 54,816 paragraphs across these articles indicating whether they represent authentic counter-hate arguments. Since this dataset was public, We had access to this dataset, including the hateful tweets, the candidate articles, and the annotations. The dataset was created using specific filtering criteria to

ensure the hateful tweets target the individuals mentioned and to verify that the segments within these tweets are indeed hateful.

The tweets containing the names of the 50 targeted individuals are retrieved and then passed through a hate classifier, HateXPlain, to automatically identify tweets with hateful content. There are several cases when a hateful tweet mentioning an individual is not necessarily hateful towards them. We are using specific patterns based on part-of-speech tags to define a hate segment (a part of the tweet) that directly targets the individual with hate. We are also performing data preprocessing after selecting the feature inputs including removing URLs, user mentions, and non-ASCII characters, as well as tokenizing and encoding the text data using a BERT tokenizer. Finally, we created the same data splits as mentioned in the published paper for training, development (dev), and testing during the preprocessing phase, ensuring that the train/dev/test sets were consistent for model training and evaluation.

3.2 Implementation

We have implemented this model using the published code as the base code to verify its validity by performing experiments along with different input features, as mentioned in the research paper. This code was publically accessible at this - https://github.com/albanyan/counterhate_paragraph GitHub repository. In this model, we are preparing the data on two granularity levels - Paragraph and Article. We are training our data using the RoBERTa transformer model Liu et al. (2019) for paragraph-level predictions and a LongFormer model Beltagy et al. (2020) for article-level predictions. To evaluate the model, we are using Precision (P), Recall (R), and F1 score (F1) as metrics. Here is the GitHub link of our reimplementation https://github.com/swabhipapneja/Implementing_Counter-hate_Paragraph

We have mainly used four files in our model.

1. **'prepare_data.py'**: After critically understanding the code given for this file, we figured that this file performs the following tasks:
   (a) In this file, tweets and articles are preprocessed to remove noise such as URLs, user mentions, and non-ASCII characters, and transformed into a lower-case format to standardize the text.
   (b) Then the BERT tokenizer is used to encode the texts into a format suitable for the model, which includes creating input IDs and attention masks.
   (c) We also define the input features and which experiment we are conducting by defining the granularity level (article or paragraph) and selecting the features. The 'EXP' variable is a list that determines which features from the dataset are included for model training and evaluation.
   (d) EXP is set depending on the –level argument passed through the command line, which decides the granularity of text data (paragraph or article) to be used in the experiments.
   (e) The processed data is then split into training, validation, and test sets.
2. **'train.py':**
   (a) In this file, we initialize a sequence classification model using the Hugging Face transformers library. Depending on the level of analysis (paragraph or article), a RoBERTa or Longformer model is loaded.
   (b) The training data loader feeds batches of the preprocessed and tokenized data to the model. Each training batch includes input IDs, attention masks, and ground truth labels.
   (c) For each epoch, the model goes through the training set, computing the forward pass, calculating the loss, performing backpropagation, and updating the weights using the AdamW optimizer.
   (d) After each epoch, the model is evaluated on the validation set to track performance metrics such as accuracy, precision, recall, and F1 score.
3. **'test.py':**

(a) This file loads the trained model and the test data loader.

(b) Predictions are made, and then the script calculates the classification metrics precision, recall, and F1 score for the predicted labels against the true labels.

4. **'error_analysis.py':**

(a) This file evaluates a trained classification model's performance on a test dataset, calculating and printing key metrics like precision, recall, and F1-score.

(b) It specifically identifies and saves instances where the model has made incorrect predictions. These results, including detailed logs of incorrect predictions, are saved to CSV and numpy files for in-depth analysis.

We have experimented with a few hyperparameters during this task. Below are the final selected values for the following hyperparameters:

	Epochs	Batch Size	LR	EPS
Values	6	16	1e-5	1e-8

Figure 1: Hyperparameters for Paragraph-level

	Epochs	Batch Size	LR	EPS
Values	6	8	1e-5	1e-8

Figure 2: Hyperparameters for Article-level

3.3 Results

As shown in Table 1 (Our Results) and Table 2 (Published Results), our reimplementation of the project yielded results that closely mirrored the published findings, with minor variances observed across metrics. For instance, in the article-level experiments using "tweet" as the feature, there is a slight increase in precision for the "No" category and a decrease in the "Yes" category, indicating a more conservative prediction of the positive class. Notably, in the "tweet", and "article" experiments, the reimplementation appears to better balance precision and recall, suggesting a more robust model fit. At the paragraph level, the performance remains consistent, demonstrating the reproducibility of the model across different granularities of text. Overall, the replication effort successfully validates the original study's claims, with our replicated model exhibiting similar results verifying the reproducibility of the study.

3.4 Discussion

We met with several challenges while reimplementing this paper, mentioned below in the pointers:

- We encountered notable dependency and version compatibility issues which required careful management: majorly with Spicy and Tornado. Downgrading the tornado version as specified in the requirements.txt file ended up costing us 3 days of Hopper Cluster Access since we downgraded the tornado to a version less than 6.1.0 which is required by Jupyter Server in Hopper. This issue existed for both of the contributors of the paper, and turned out to be a major challenge for getting GPU resources.

- A significant deviation was observed in the evaluation metrics between the code and the paper's description and reporting. The code focused on accuracy, while the paper reported precision, recall, and F1-score, leading to discrepancies between the provided code to that of the original paper's findings. We modified train.py and test.py to modify the code to get the results as given in the paper.

- Furthermore, the code provided did not include the capacity to test the metrics for various feature combinations detailed in the paper. We addressed this by adapting the code to a subset of all the feature combinations mentioned in the paper, enabling us to validate selected experiments and explore these feature interactions.

- The results of our experiments closely align with the experiments detailed in the paper. Although the values are not a perfect match, they don't differ significantly from the results between the implemented code to that reported in the

Table 1: Our results

Level	EXP	No			Yes		
		P	R	F1	P	R	F1
Article	"tweet"	0.95	0.9	0.93	0.7	0.83	0.76
Article	"tweet", "article"	0.93	0.95	0.94	0.79	0.75	0.77
Paragraph	"tweet"	0.99	0.98	0.98	0.61	0.7	0.65
Paragraph	"tweet", "paragraph"	0.99	0.98	0.98	0.61	0.71	0.66

Table 2: Published results

Level	EXP	No			Yes		
		P	R	F1	P	R	F1
Article	"tweet"	0.96	0.87	0.91	0.65	0.85	0.74
Article	"tweet", "article"	0.83	0.95	0.89	0.62	0.3	0.41
Paragraph	"tweet"	0.99	0.98	0.99	0.65	0.76	0.7
Paragraph	"tweet", "paragraph"	0.99	0.97	0.98	0.57	0.76	0.65

original study. The slight discrepancies observed could be attributed to inherent variability in machine learning workflows, including differences in computational environments or the stochastic nature of the algorithms. However, the results were consistent enough to validate the original findings, confirming the robustness of the reported research.

- The codebase provided lacked comprehensive documentation and dedicated READme files making the existing code deficient in readability and understandability. To improve this, we meticulously added descriptive comments to the code and created detailed READme files for each executable Python file, thereby enhancing the clarity of the implementation. These efforts, although time-consuming, were critical in aligning our results closely with those reported in the publication.

- Moreover, the original codebase did not provide bash scripts, which are often essential for streamlining the execution of experiments. Recognizing this, we developed our own set of bash scripts, which will further allow interested researchers to efficiently perform experimental runs and test various feature combinations. The introduction of these scripts ensures that our experiments will be reproducible and consistent, thus serving as a valuable resource for our implementation.

- We implemented and performed error analysis as mentioned in detail in section 3.6.

- Despite these challenges, we took rigorous steps to enhance our understanding of both the paper and the code. We documented a detailed overall READme file along with code READme files, that includes our implementation in detail for interested researchers to replicate the results for our implementation of the paper.

3.5 Resources

The cost of our reproduction in terms of resources: computation, time, people, development effort, and communication with the authors (if applicable).

In reproducing the results of our study, we leveraged computational resources equivalent to 50 GPU hours using Hopper clusters. It took us time to finalize a paper and replicate the results of the paper. We started with finding a few relevant papers, going through them and understanding, the methodologies described in them. This was followed by shortlisting 2 papers, running both and finalizing one because of various dependency issues in one of them. We spend time doing literature reviews too. As a team of two contributors, we spent a total of 100 hours per person in the project. The result replication effort was substantial. We spend considerable amount of time in code modification, documentation,

enhancing code readability, error analysis and report writing. Overall, the reproduction was resource-intensive but feasible within our allocated timeframe.

3.6 Error Analysis

In the article-level experiment where the input feature included "tweet only, we recognized that 44 samples were incorrectly classified. Below mentioned is an analysis of some incorrectly classified articles. Note that on the article level, we label it as 1 if the article includes one sentence that can be considered an authentic counterhate argument.
Hateful Tweet 1 - "Avril lavigne is ugly she have no telant"
Article 1 - "Avrils career is still going, although her big moment was really in the early 2000' s, and her biggest success was the first album, Let Go. She continues to have success and is an iconic figure in the world of punk pop. Avril Lavigne is a Canadian singer, songwriter, and actress."
True Class - 0
Predicted Class - 1
Explanation - This article focuses on Lavigne's career trajectory and does not address the hate speech present in the tweet, yet it might have been classified as counter-hate because it presents positive information about her career.

Hateful Tweet 2 - "Jimmy Carter is a disgusting Jew hater. A pox on him and his birthday"

Article 2 - "Carter was the first president to put human rights at the forefront of his foreign policy. "I know how easy it is to overlook the persecution of others when your own rights and freedoms are not in jeopardy," he later wrote, referring to his upbringing in segregationist Georgia. "To me ···moral principles were the best foundation for the exertion of American power and influence."
True Class - 0
Predicted Class - 1
Explanation - This paragraph highlights Carter's human rights efforts and could be misclassified as counter-hate because probably the model does not understand that it does not address the specific hateful claim about anti-Semitism.

4 Plan for Checkpoint 2

In Checkpoint 2, in order to perform deeper analysis and improvement upon the baseline model, we want to go forward with the Robustness dimension.

For robustness analysis and improvement, we can do two things:

- Evaluating Model Robustness: We plan to implement a robustness evaluation of the model. The evaluation will focus on how the model performs when faced with tweets with noise, spelling errors, typos, grammar mistakes, and ambiguity.

- Multilinguality Exploration: We plan to train our model with multiple languages since the original model is English language-specific. We want to experiment with models like mBERT or XLM-R that have multilingual capabilities. We will substitute the current monolingual BERT model with a multilingual counterpart and retrain with translated versions of the dataset or new multilingual datasets if available based on our research.

We will expand our report to include the results and methodologies from Checkpoint 2, highlighting robustness improvements, and multilingual performance conducted by us. The workload will be evenly divided among the team, ensuring equal contributions to both code development and report writing.

5 Workload Clarification - Checkpoint 1

Both of us collaborated on this project equally. We started by looking into a wide variety of research papers. We researched papers together and discussed their scope, possible implementation steps and potential challenges. We then divided 3 papers each for the in-depth research analysis. After going through several interesting papers and understanding their scope, we started on the implementation of two papers individually. We did face a lot of dependency version errors and we collaborated on fixing these errors. We maintained team collaboration with daily project meetings, discussing progress and blockers. After finalising

our paper, both of us worked on understanding the code and creating documentation files. We then divided the experiments and ran the experiments on different hyperparameters. We then compared our results, rectified our errors and finalised the re-implementation of the code together. We discussed the answers to all the questions asked and then divided the sections of the project report to be documented, equally. Overall, we were able to divide the workload, communicate and collaborate while working on this project effectively.

6 Conclusions - Checkpoint 1

We were able to reproduce the results in the paper as our obtained results are in close agreement with that of the original paper.

To reproduce the results, we were able to follow, the methods and procedures described in the paper to achieve comparable results using the same data and computational processes.

Things that majorly contributed to reproducing the results were:

- Availability of dataset and details around their data generation process.
- Clarity of methodology described in the paper. Since the original paper provided enough detail about the model architecture, training process, and hyperparameters to allow for replication.

Checkpoint 2: Model Analysis & Improvement

We chose the **Multilingualality** dimension to perform a deeper analysis and improve upon the baseline model. Our data set was originally in English language only, on both article and paragraph levels. Now, for checkpoint 2 we have expanded our dataset onto 9 more languages other than English. Accordingly, we have made several modifications to our model to handle the multilingual experiments. The approach for training our multilingual model is described below.

7 Multilingual Model Training Approach

We have described our training approach and modifications we made to our model to make it a multilingual one. We will be discussing this in three sections: **Data Translation, Data Preparation and Model Training.**

7.1 Data Translation

Since the primary objective of this study is to identify the authentic counter-hate arguments for hateful tweets, we started by researching the top 10 languages used on Twitter. According to (sos, 2023) we found out that the most used languages on Twitter are: English, Japanese, Spanish, Portuguese, Arabic, French, Indonesian, Russian, Turkish, and Hindi. So, we started by translating our original English language dataset, both articles and paragraphs, into all these languages. Fig 3 shows the percentage of frequency of use of the top 10 languages used in twitter.

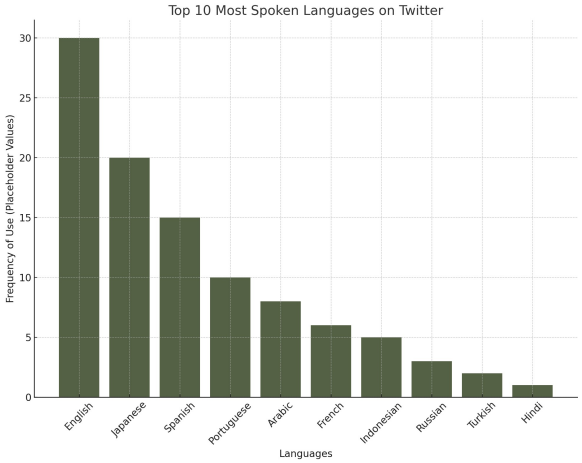


Figure 3: Top 10 Most Spoken Languages in Twitter

7.1.1 Article Level Translation:

- To translate our article level data, we have created a script that leverages the Google Translate API to automatically translate the Article Dataset from English into multiple target languages. It is specifically designed to translate datasets containing tweets and articles, supporting a range of languages.
- We have named this file 'translate_article.py' and it processes our large article datasets by iterating through each record.
- We have used pandas, googletrans, tqdm to support our translation process.

- This file utilizes a progress bar to display the real-time progress of the translation process.
- Finally, it saves the translated dataset into a CSV file, called translated_articles.csv.

7.1.2 Paragraph Level Translation:

- We have created another Python script that automates the translation of paragraph datasets in our implementation into multiple languages using the Google Translate API.
- We designed this file to handle large batches of data, translating text fields from English into specified target languages and saving the translated content in a new file.
- We tried to translate the paragraphs dataset into all the top 10 languages on Twitter as described above. However, since the volume of the paragraph data is much more than the article data, it was taking a massive amount of time just to iterate over less than half the data.
- Due to this computation restriction, we decided to decrease the translations to 2 languages other than English on the paragraph level. Hence we are translating the data to these 2 languages - Japanese and Spanish.
- Still, we were facing the issue of a lot of time being taken for the translation process and thus we created a Python script to divide our paragraph-level data into batches, each with 6000 records. These files are kept under the folder 'spilt_files'.
- We automated the translation process so that each batch will be processed for translation and the file with the translated data will be created under the directory, 'translated_files'.

After our translations are completed, we have 25000 records of the article-level data and 164448 records of the paragraph-level data. Figure 4 shows a pictorial representation of our translated data.

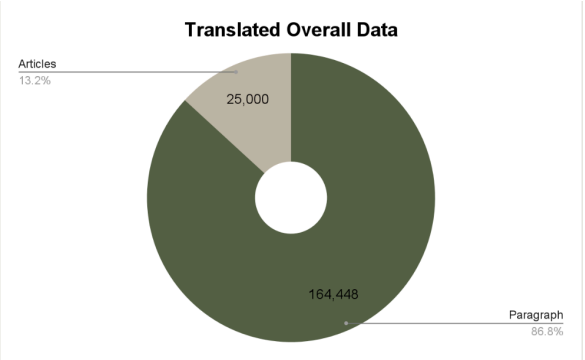


Figure 4: Translated Data Distribution

7.2 Data Preparation

The following description outlines the modifications and changes implemented to the data preparation process to enhance text processing and accommodate multilingual data handling.

We worked on enhancing the data preprocessing to enhance its capabilities to handle multiple scripts. We also made changes to our hyperparameters to adjust our model better to the updated multilingual data. Detailed information on the changes implemented in the data preparation script, `ML_prepare_data.py` can be found below.

1. **Normalization Enhancement:**
   - **Description:** We have introduced Unicode normalization in data preprocessing using `unicodedata.normalize('NFKC', x)`. This helps in representing characters with diacritical marks uniformly.
   - **Impact:** It ensures consistency in processing text across various languages, aiding in the normalization of characters and diacritics.
2. **Updated Text Processing:**
   - **Description:** We enhanced URL removal and simplified the cleaning of mentions, retweets, and hashtags. Regex patterns were updated for robust handling of modern web text.
   - **Impact:** It improves the reliability of text input into the model by removing extraneous web text elements efficiently.
3. **Handling Special Characters:**

- **Description:** We expanded processing to handle HTML entities and special characters more effectively by replacing or removing them.
- **Impact:** This ensures the quality of text data remains high, crucial for accurate model input.

4. **Whitespace Normalization:**
    - **Description:** We have normalized whitespace by condensing multiple spaces into a single space.
    - **Impact:** This has helped us in standardizing text input for tokenization and maintaining correct token boundaries.
5. **Tokenizer Update:**
    - **Description:** We have changed the tokenizer across different data levels to `xlm-roberta-base`, suitable for multilingual text.
    - **Impact:** We chose XLMR because it has been trained in more than 100 languages enhancing its capabilities to handle languages written in different scripts.
6. **Batch Size Adjustment:**
    - **Description:** We have changed the batch sizes for both data levels when using the `xlm-roberta-base` tokenizer. Now, we are using batch size 16 for Article-level data and 24 for paragraph level while fine-tuning our model.
    - **Impact:** This change will ensure that our model adjusts to the requirements of the new tokenizer, ensuring efficient data handling during training.
7. **Saving raw splits to CSV:**
    - **Description:** Alongside, we are saving the raw splits of train, validation and test to CSV files to save our true labels and raw text data.
    - **Impact:** This helps in enhancement of the error analysis process going forward.

These enhancements are designed to make the data preparation script more robust in handling diverse and noisy web text, particularly for multilingual contexts. The use of a powerful, unified tokenizer aligns with the goal of simplifying and standardizing the preprocessing steps.

7.3 Model Training

After creating our multilingual data, doing the required preprocessing steps, changing our tokenizer, and creating validation, training and testing splits of data, the next part is training our model. We have described the changes we implemented in our training script in detail below. Our modifications prepare the model for improved performance on our multilingual dataset.

We have made several modifications to our `ML_train.py` script to incorporate changes adapting the training script for a different pretrained model (XLM-R) and addressing enhanced performance in multilingual settings.

1. **Model Change**: As mentioned previously, we switched from using `roberta-base` and `allenai/longformer-base-4096` to `xlm-roberta-base` for both paragraph and article levels, because this model is capable of handling multiple languages.
2. **Optimizer and Scheduler Configuration**: We have maintained the use of `AdamW` optimizer but ensured it is optimized for `xlm-roberta-base`. The learning rate (`lr=1e-5`) and epsilon (`eps=1e-8`) remain optimized for stability in gradients during training.
3. **Hyperparameter Tuning**:
    - The epoch count (`EPOCHS`) is set to 6, the same as in checkpoint 1.
    - We tried multiple hyperparameters and ended up choosing the ones that gave the best results. We have mentioned these in Figure 5 and 6.
    - We used a learning rate scheduler get_linear_schedule_with_warmup without warmup steps. This ensures that the learning rate decreases linearly from the initial rate set by the optimizer.

4. **Training Time**:

   - We worked on translating our dataset for our multilingual model on both article and paragraph level data for 5 days approximately.
   - We worked on code modifications for `ML_prepare_data.py`, `ML_train.py` and `ML_test.py` for 5 to 6 days, followed by learning the model and doing hyperparameter tuning and fixing the best parameters for 2 to 3 days.
   - We ran multiple experiments for 3 more days.

Figure 5 and Figure 6 show the hyperparameters used in the training phase of our multilingual model.

	Epochs	Batch Size	LR	EPS
Values	6	24	1e-5	1e-8

Figure 5: Hyperparameters for Paragraph-level

	Epochs	Batch Size	LR	EPS
Values	6	16	1e-5	1e-8

Figure 6: Hyperparameters for Article-level

8 Approach for Analyzing a Multilingual Model

To evaluate our multilingual model in different languages, we are using the same evaluation metrics, Precision, Recall and F1 Score. We will be discussing our efforts and modifications for this purpose in two sections below - **Model Testing and Error Analysis Enhancement**.

8.1 Model Testing

We have modified our script for evaluating the model, `test.py` to enhance evaluation functionality. These changes facilitate a more detailed examination of the model's performance and provide additional utilities for integrating test results into further analysis processes. A detailed description of our modifications is listed below.

1. **Integration with Raw Data:**

   - **Loading Raw Data:** Our testing script now loads a CSV file (`test_raw.csv`) that contains the raw test data. This is crucial for comparing the predicted results with the actual data.
   - **Appending Predictions:** After model evaluation, the predictions are appended to the loaded data frame. This allows for a direct comparison between predicted and actual labels within the same data frame.
   - **Saving Enhanced Data:** The data frame, now containing both original data and predictions, is saved back to a CSV file (`test_with_predictions.csv`). This file is useful for further analysis and review.

2. **Output Management:**

   - **.npy File Saving:** We are also saving our model's predictions as a `.npy` file, offering compatibility with workflows that utilize NumPy arrays for further data manipulation and analysis.

This script also prints detailed performance metrics, such as Precision, Recall, F1-Score for each class, and Weighted F1-Score, directly to the console. This allows users to quickly assess model performance without needing additional scripts. These changes aim to simplify the testing phase of model development, making it easier to evaluate model effectiveness and prepare data for further analysis or presentation. The improvements in output management help maintain a consistent and organized workflow during model evaluations.

8.2 Error Analysis Enhancement

Moving on to the error analysis phase, we have enhanced our previous work on error analysis to identify and understand misclassifications made by the model. This section outlines the key modifications made to the `error_analysis.py` script.

1. **Data Handling Enhancements:**

   - **Text Length Calculation:** We have introduced explicit column op-

erations to calculate the length of text for both tweets and articles directly within the DataFrame using Pandas' `.apply()` method.

- **Direct CSV Output:** We have added a functionality to save mispredicted data points directly to a CSV file named `mispredicted_examples.csv`. This facilitates easy access and further analysis of misclassified cases without additional scripting.

2. **Visualization Improvements:**

   - **Histograms for Text Lengths:** We have added histograms to visualize the distribution of tweet and article lengths where the model predictions were incorrect. This visual representation helps in understanding patterns in the data that may cause incorrect predictions.

   - **Enhanced Output Details:** This script outputs descriptive statistics for the lengths of tweets and articles in the error predictions, providing insights into potential issues related to text length impacting model performance. This enables faster hypothesis testing regarding error patterns, such as whether shorter or longer texts tend to be misclassified more often.

Figure 7 shows an analysis of both the tweet length and article length for the incorrectly predicted samples.

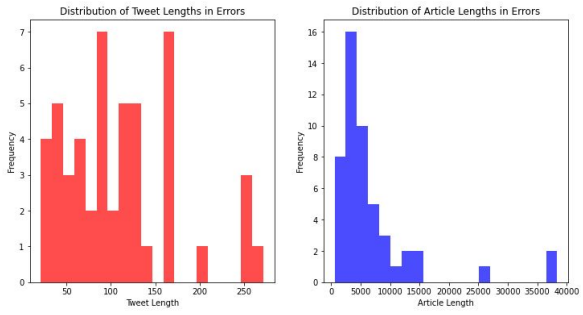


Figure 7: Tweet and Article Length Analysis

We can conclude from this analysis that tweets with lengths between 100-150 are the most prone to errors in our model. The articles that have character lengths from 1000 to 6000 have the maximum number of error predictions.

Let's look at an example to understand errors made by our model in predicting if an article is an authentic counter-hate argument.

Hateful Tweet 1 - "Bill Gates is a ugly bird looking, selfish, awkward, evil bitch"

Article 1 - Gates has been active in the US coronavirus response. The Bill and Melinda Gates Foundation has partnered with the government on vaccine research, committing more than 250 million to help develop coronavirus diagnostics, therapeutics, and vaccines. Gates said he expects we'll get some 'good vaccine tools' in the first half of 2021. Bill Gates criticized the US coronavirus response, calling the current state of the outbreak in the US 'an ugly picture."We had criteria for opening up that said you had to have cases declining and we opened up with cases increasing,' Gates said. 'We somehow got masks as this politicized thing ... and some like, harbinger of freedom, that just covering your mouth was awful.'Now that Trump is endorsing wearing a mask, he said, the US can 'move forward' and try to get mask compliance levels closer to Europe's level and bring down transmission rates.

True Class - 0

Predicted Class - 1

Explanation - The article discusses Bill Gates' involvement in coronavirus response and vaccine development but does not address the derogatory claims in the tweet. However, the positive depiction of Gates' efforts, particularly his critical and proactive stance against the U.S. virus response, might have led the model to misclassify the article as countering the hate speech.

Hateful Tweet 2 - I fucking hate Cristiano Ronaldo. He's so full of shit.

Paragraph 2 - Ronaldo in 2003 and 2017 are two completely different animals. The transformation is not given but through tough work and true commitment to conditioning. Ronaldo had raw talent but his physical gifts were sorely lacking in a league famed for its physicality. Ronaldo could settle for being a traditional winger, just gaining some mass and

keep the raw pace and trickery in tact. However, Ronaldo went on strict training regimes and gained strength and weight without losing any of his raw speed and even increasing it. This meant he gained power alongside pace.

True Class - 1

Predicted Class - 0

Explanation - The article details Ronaldo's dedication to improving his physical abilities and transforming his gameplay, demonstrating his commitment and hard work. It presents a positive view of Ronaldo's character and achievements, potentially serving as a counter to the hateful sentiment expressed in the tweet. However, the model may have failed to recognize this as a counter-argument because it doesn't directly address the negative personal accusations made in the tweet, focusing instead on his professional development.

9 Experimental Results

We performed the same experiments as we did in checkpoint 1 with our modified multilingual model. Table 3 shows the results of our experiments. We experimented with other hyperparameters as well, but we have reported our best results. Our multilingual model substantially outperforms the baseline model across various metrics, demonstrating its enhanced ability to process and understand linguistic nuances. Notably, in the "Yes" category, which measures the model's ability to identify countering arguments, the multilingual model shows significant improvements in recall and F1 scores, such as a rise from 0.85 to 0.93 in recall and 0.74 to 0.95 in F1 for article-level analyses using only tweets. This improvement indicates a stronger capability to accurately identify true counter-hate arguments. This might also be a result of keeping the data balanced across all languages we have chosen to translate to. Additionally, the model maintains high performance at both article and paragraph levels. Overall, the enhanced linguistic capabilities of the multilingual model make it more reliable and effective for practical applications, particularly in identifying both authentic counter-hate and synthetic counter-hate arguments related to hateful content on social media.

9.1 Multilingual Model Analysis based on Languages

Our implemented multilingual model outperforms our base model across various metrics as mentioned in Table 3, demonstrating its ability to process and understand various linguistic nuances. As you can see in the same table we performed multilingual modeling across 4 experiments. Followed by which we did detailed error analysis across all 4 of these experiments.

Figure 8 shows a part of our error analysis on the paragraph level data including Tweets and Paragraphs for the incorrectly predicted samples.

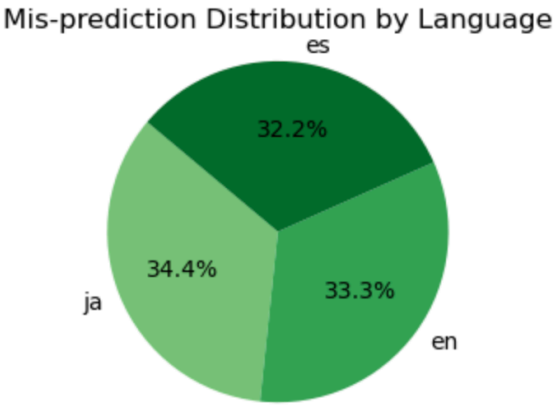


Figure 8: Mis-prediction Distribution by Language on Paragraph Level Data.

Our multilingual model was tested on paragraphs in English, Japanese, and Spanish, showing balanced mis-prediction rates of 33.3% for English, 34.4% for Japanese, and 32.2% for Spanish. The model's overall performance shows that it maintains a level of consistency across languages. However, the slight variation in mis-prediction rates indicates underlying challenges specific to each language, such as the complex context sensitivity of Japanese. Although this consistent performance across language could be potentially due to the creation a dataset, balanced with respect to all the languages ensuring proper representation of each of the languages for the model to learn from.

We conducted similar error analysis for all 4 of the experiments we performed, and these analysis can be found in the ML_Error_Analysis.ipynb for reference.

Table 3: Multilingual Model results

Level	EXP	No			Yes		
		P	R	F1	P	R	F1
Article	”tweet”	1.0	0.98	0.99	0.93	0.98	0.96
Article	”tweet”, ”article”	0.99	0.98	0.99	0.92	0.98	0.95
Paragraph	”tweet”	0.99	0.98	0.99	0.69	0.78	0.73
Paragraph	”tweet”, ”paragraph”	0.99	0.98	0.99	0.69	0.80	0.74

9.2 Performance Analysis with Examples

We have explained some examples of where the model performs well and where it fails on both the article level and paragraph level. As discussed above, we are translating our article-level data into 10 languages, and paragraph-level data in 2 languages. We will be discussing examples from the paragraph-level data to understand the model performance and analyse its mispredicted samples in paragraph data in **Japanese and Spanish.**

9.2.1 Japanese: Correctly Predicted Samples:

- **Tweet:** ディック・チェイニFはF犯だが、ところで彼はもういない、2 兆 3000 億ドルの兄弟はどこにいるのか。
- **Translation:** Dick Cheney is a war criminal, but by the way, he's gone now, where are the 2.3 trillion brothers?
  - **Paragraph 1**: 彼らはイラク人民を解放し、世界のテロリズムを終わらせたので起訴されなかった。人々は彼らにとてもF足しています。
  - **Translation 1:** They were not prosecuted because they liberated the Iraqi people and ended terrorism in the world. People are very satisfied with them.
  - **True Class:** 0
  - **Predicted Class:** 0
  - **Explanation:** The model correctly predicts this as non-counter because the paragraph, rather than contesting the tweet's claim of criminality, shifts focus to positive outcomes, not directly addressing the war crime accusation.
  - **Paragraph 2:** 彼は素晴らしいリFダFです。彼はピッチからもサイドラインからも試合に影響を与えることができる。
  - **Translation 2:** He is a great leader. He can influence the game from both the pitch and the sidelines.
  - **True Class:** 1
  - **Predicted Class:** 1
  - **Explanation:** The model correctly identifies this as a counter because the paragraph presents a contrasting view, framing Cheney as a ”great leader” which opposes the negative connotation of ”war criminal” suggested in the tweet.

9.2.2 Japanese: Incorrectly Predicted Samples:

- **Tweet:** クリスティアFノ・ロナウドは大嫌いだ。彼は本当にたわごとでいっぱいです
- **Translation:** I really hate Cristiano Ronaldo. He is full of crap.
  - **Paragraph 1**: 彼は若くして偉大な選手としてスタFトを切ったものの、成長しているようだ。クリスティアFノは以前は利己的で、確かに少し泣き言も言っていましたが、私はF化を見てきました。私は彼のことは嫌いですが...
  - **Translation 1:** He started as a great player at a young age and seems to be growing. Cristiano was selfish before and certainly complained a bit, but I have seen changes. I don't like him, but I watch his play, and I have seen him grow significantly under Sir Alex Ferguson.
  - **True Class:** 0
  - **Predicted Class:** 1
  - **Explanation:**

    - The paragraph provides a narrative of Ronaldo's professional growth, but it doesn't directly address or refute the hateful sentiment expressed in the tweet, which is about his personal characteristics rather than his professional abilities.
    - The model's misclassification could be due to picking up on the positive tone of the paragraph and mistakenly interpreting it as a counter to the hate expressed in the tweet.
    - The model failed to distinguish between content that shares positive information and content that actively counters a negative claim.
  - **Paragraph 2:** 彼はあなたや私のような人々がなりたいものすべてです。
  - **Translation 2:** He is everything that you and I aspire to be.
  - **True Class:** 1
  - **Predicted Class:** 0
  - **Explanation:**
    - This paragraph provides a direct contradiction to the hateful sentiment expressed in the tweet by highlighting Ronaldo as an exemplary figure, embodying qualities that people generally admire and aspire to achieve. Thus, it should be labelled as a counter-hate argument.
    - This example illustrates the difficulty models can have in interpreting the complex nature of a language where the counter to hate is implied through positive contradictions rather than direct opposition.

9.2.3 Spanish: Correctly Predicted Samples:

- **Tweet:** Odio a Cristiano Ronaldo. Está tan lleno de mierda.
- **Translation:** I hate Cristiano Ronaldo. He is so full of shit.
  - **Paragraph 1**: LE ENCANTA GANAR.
  - **Translation 1:** HE LOVES TO WIN
  - **True Class:** 1
  - **Predicted Class:** 1
  - **Explanation:** The model correctly identifies this as a counter because the paragraph shifts the narrative by emphasizing Ronaldo's passion for winning, which implicitly counters the negative perception expressed in the tweet.
  - **Paragraph 2:** Sin embargo, Ronaldo abrió el camino con un recorte salarial para marzo, abril, mayo y junio del año pasado mientras esperábamos para ver qué haría el brote inicial del coronavirus en el mundo del fútbol.
  - **Translation 2:** However, Ronaldo paved the way with a salary cut for March, April, May, and June of last year as we waited to see what the initial outbreak of the coronavirus would do in the world of football.
  - **True Class:** 0
  - **Predicted Class:** 0
  - **Explanation:** The model correctly predicts this as non-counter since the paragraph, although positive, does not address the derogatory sentiment in the tweet directly but instead focuses on Ronaldo's altruistic behavior during the pandemic, which does not counter the specific hate expressed.

9.2.4 Spanish: Incorrectly Predicted Samples:

- **Tweet:** Newt Gingrich está demente; Cawthorn un apóstol sin sentido.
- **Translation:** Newt Gingrich is insane; Cawthorn a senseless apostle.
  - **Paragraph 1**: El mandato de Newt Gingrich como presidente de la Cámara es un buen reflejo de su carrera política en su conjunto.

- **Translation 1:** Newt Gingrich's tenure as Speaker of the House is a good reflection of his political career as a whole.
- **True Class:** 1
- **Predicted Class:** 0
- **Explanation:** The model incorrectly predicts this as non-counter because the paragraph, intended to provide a neutral or potentially positive overview of Gingrich's career, doesn't explicitly counter the claim of insanity in the tweet. The model likely failed to interpret the subtle implications that a successful tenure could indirectly refute the notion of 'insanity'.
- **Paragraph 2:**Las ventajas de Newt Gingrich: 1) Celebró el poder del cerebro. Promovió un ambiente más intelectual. Dirigió un grupo de ex profesores. 2) Sabía cuándo hacer un trato. Hizo algunos compromisos razonables, incluida la reforma del bienestar social.
- **Translation 2:** The advantages of Newt Gingrich: 1) He celebrated the power of the brain. Promoted a more intellectual environment. Led a group of former professors. 2) Knew when to make a deal. Made some reasonable compromises, including welfare reform.
- **True Class:** 0
- **Predicted Class:** 1
- **Explanation:** The model incorrectly predicts this as a counter because it lists achievements and positive attributes which may seem to oppose the negative connotations in the tweet. The model misinterpreted these positives as direct counters to the claims of madness and senselessness, reflecting a challenge in distinguishing between general positive descriptions and specific rebuttals to negative accusations.

10 Discussions

While working on this model, a major challenge we faced was computation restrictions that led to a massive amount of run time for translating our paragraph-level dataset into the targeted languages. After trying multiple times we realized the paragraph data is too huge to be translated all at once. It was taking more than 12-15 hours, and every time we tried, we were getting a server connection error after this time slot in ORC Hopper. So, we decided to proceed with the translation process in batches. Even after that, we were not able to translate all to 10 targeted languages, due to which we decided to reduce the size of our data and proceed with translation in only two languages for the paragraph level. While our model was taking a lot of time to run, we experimented with various batch sizes. Then we finally ran our paragraph-level experiments for our multilingual model on a subset of our translated data with 18000 records (1 Batch). If we had more time to work on this project, we wanted to look into this challenge so that we could evaluate the performance of our model on the complete dataset for paragraph level. Along with this, in future, we can also work on expanding the dataset by adding real-time hateful twitter speech and potential counter-hate argument data in other languages rather than simply translating our original dataset.

11 Workload Clarification - Checkpoint 2

For checkpoint 2, both of us contributed equally to understanding the task and designing the modifications required. We then collaborated at every step of this project, from the translation of our dataset to figuring out a way to solve the challenges faced during the task. We worked on error analysis enhancements together. We also made the required code modifications in collaboration with each other. After our code was ready to run, we divided the experiments into article and paragraph-level data. Then both of us worked individually on hyperparameter tuning to achieve the best results. We divided the segments of the report and completed the project with equal contributions.

12 Overall Conclusions

To conclude this project, we were able to achieve significantly close results to those in

the published research paper, indicating that we were able to reproduce results present in the paper. For checkpoint 1, we used the same models as mentioned in the research paper. For checkpoint 2, we made several modifications to analyse if it's feasible to make it multilingual. Our modified multilingual model has been trained in 10 languages at the article level and 2 languages at the paragraph level. To enhance the capabilities of our model, we chose to train our model on XLM-R in the second phase of the project. After running several experiments, we analysed our model with the same evaluation metrics as that in checkpoint 1. In conclusion, our model performs equally well in multilingual settings as can be seen clearly in our experimental results.

Here is the GitHub link of our complete re-implementation for both Checkpoint 1 and Checkpoint 2. `https://github.com/swabhipapneja/Implementing_Counter-hate_Paragraph`

References

2023. The 10 most spoken languages on X –Twitter.

Brooke Auxier. 2020. Social media has negative effect on the way things are going.

Brooke Auxier and Monica Anderson. 2021. Social media use in 2021.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, U.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroglu, and Marco Guerini. 2019. Conan - counter narratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2829, Florence, Italy. Association for Computational Linguistics.

Yi-Ling Chung, Serra Sinem Tekiroglu, and Marco Guerini. 2021. Towards knowledge-grounded counter narrative generation for hate speech. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 899–914, Online. Association for Computational Linguistics.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. ACM Comput. Surv., 51(4).

Bing He, Caleb Ziems, Sandeep Soni, Naren Ramakrishnan, Diyi Yang, and Srijan Kumar. 2021. Racism is a virus: anti-asian hate and counterspeech in social media during the covid-19 crisis. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 90–94.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

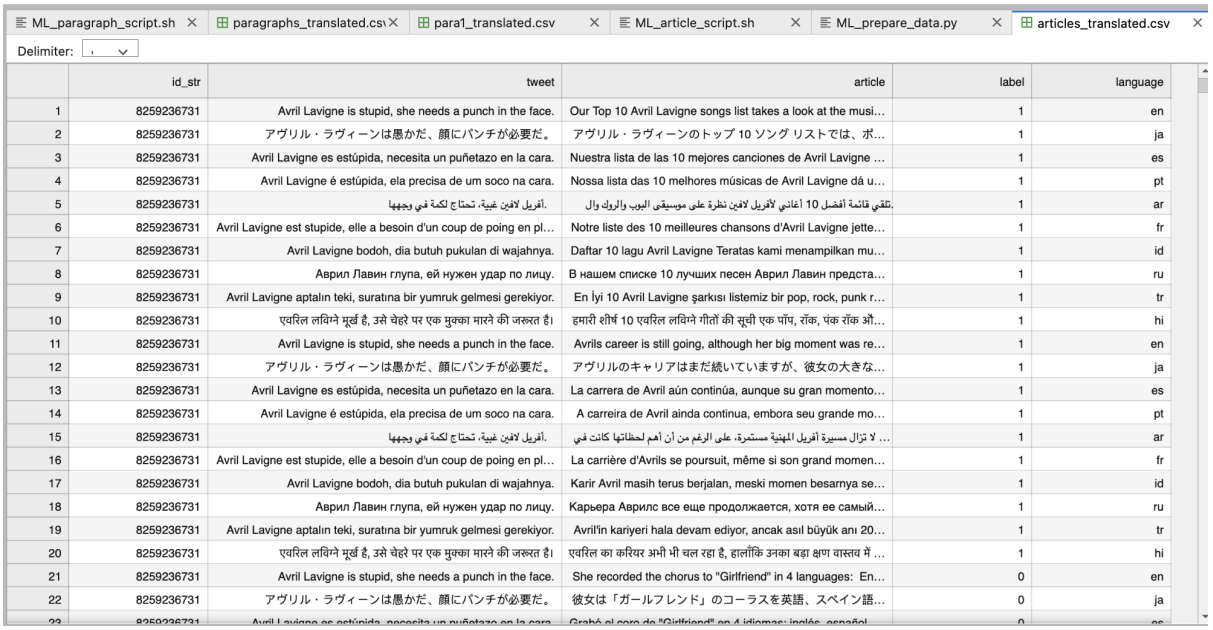
Binny Mathew, Navish Kumar, Pawan Goyal, and Animesh Mukherjee. 2020. Interaction dynamics between hate and counter users on twitter. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, CoDS COMAD 2020*, pages 116–124, New York, NY, USA. Association for Computing Machinery.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4755–4764, Hong Kong, China. Association for Computational Linguistics.

Niam Yaraghi. 2019. How should social media platforms combat misinformation and hate speech?

APPENDIX

1. Showing a snapshot of our translated article dataset for checkpoint 2. Here the data is getting converted to 10 Languages.

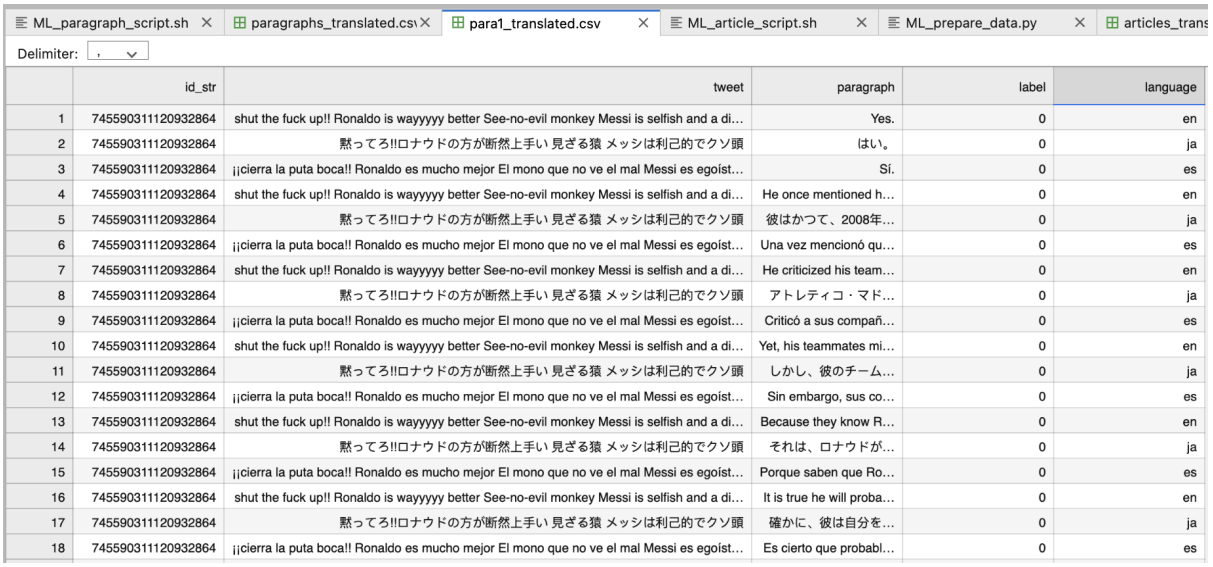


ML_paragraph_script.sh | paragraphs_translated.csv | para1_translated.csv | ML_article_script.sh | ML_prepare_data.py | articles_translated.csv

Delimiter: ,

	id_str	tweet	article	label	language
1	8259236731	Avril Lavigne is stupid, she needs a punch in the face.	Our Top 10 Avril Lavigne songs list takes a look at the musi…	1	en
2	8259236731	アヴリル・ラヴィーンは愚かだ、顔にパンチが必要だ。	アヴリル・ラヴィーンのトップ 10 ソング リストでは、ボ…	1	ja
3	8259236731	Avril Lavigne es estúpida, necesita un puñetazo en la cara.	Nuestra lista de las 10 mejores canciones de Avril Lavigne …	1	es
4	8259236731	Avril Lavigne é estúpida, ela precisa de um soco na cara.	Nossa lista das 10 melhores músicas de Avril Lavigne dá u…	1	pt
5	8259236731	أفريل لافين غبية، تحتاج لكمة في وجهها.	تلقي قائمة أفضل 10 أغاني لأفريل لافين نظرة على موسيقى البوب والروك وال…	1	ar
6	8259236731	Avril Lavigne est stupide, elle a besoin d'un coup de poing en pl…	Notre liste des 10 meilleures chansons d'Avril Lavigne jette…	1	fr
7	8259236731	Avril Lavigne bodoh, dia butuh pukulan di wajahnya.	Daftar 10 lagu Avril Lavigne Teratas kami menampilkan mu…	1	id
8	8259236731	Аврил Лавин глупа, ей нужен удар по лицу.	В нашем списке 10 лучших песен Аврил Лавин предста…	1	ru
9	8259236731	Avril Lavigne aptalın teki, suratına bir yumruk gelmesi gerekiyor.	En İyi 10 Avril Lavigne şarkısı listemiz bir pop, rock, punk r…	1	tr
10	8259236731	एवरिल लविग्ने मूर्ख है, उसे चेहरे पर एक मुक्का मारने की जरूरत है।	हमारी शीर्ष 10 एवरिल लविग्ने गीतों की सूची एक पॉप, रॉक, पंक रॉक औ…	1	hi
11	8259236731	Avril Lavigne is stupid, she needs a punch in the face.	Avrils career is still going, although her big moment was re…	1	en
12	8259236731	アヴリル・ラヴィーンは愚かだ、顔にパンチが必要だ。	アヴリルのキャリアはまだ続いていますが、彼女の大きな…	1	ja
13	8259236731	Avril Lavigne es estúpida, necesita un puñetazo en la cara.	La carrera de Avril aún continúa, aunque su gran momento…	1	es
14	8259236731	Avril Lavigne é estúpida, ela precisa de um soco na cara.	A carreira de Avril ainda continua, embora seu grande mo…	1	pt
15	8259236731	أفريل لافين غبية، تحتاج لكمة في وجهها.	لا تزال مسيرة أفريل المهنية مستمرة، على الرغم من أن أهم لحظاتها كانت في …	1	ar
16	8259236731	Avril Lavigne est stupide, elle a besoin d'un coup de poing en pl…	La carrière d'Avrils se poursuit, même si son grand momen…	1	fr
17	8259236731	Avril Lavigne bodoh, dia butuh pukulan di wajahnya.	Karir Avril masih terus berjalan, meski momen besarnya se…	1	id
18	8259236731	Аврил Лавин глупа, ей нужен удар по лицу.	Карьера Аврилс все еще продолжается, хотя ее самый…	1	ru
19	8259236731	Avril Lavigne aptalın teki, suratına bir yumruk gelmesi gerekiyor.	Avril'in kariyeri hala devam ediyor, ancak asıl büyük anı 20…	1	tr
20	8259236731	एवरिल लविग्ने मूर्ख है, उसे चेहरे पर एक मुक्का मारने की जरूरत है।	एवरिल का करियर अभी भी चल रहा है, हालाँकि उनका बड़ा क्षण वास्तव में …	1	hi
21	8259236731	Avril Lavigne is stupid, she needs a punch in the face.	She recorded the chorus to "Girlfriend" in 4 languages: En…	0	en
22	8259236731	アヴリル・ラヴィーンは愚かだ、顔にパンチが必要だ。	彼女は「ガールフレンド」のコーラスを英語、スペイン語…	0	ja
23	8259236731	Avril Lavigne es estúpida, necesita un puñetazo en la cara.	Grabó el coro de "Girlfriend" en 4 idiomas: inglés, español	0	es

2. Showing a snapshot of our translated paragraph (one of the 10 batches) dataset for checkpoint 2. Here the data is getting converted to 2 Languages.



ML_paragraph_script.sh | paragraphs_translated.csv | para1_translated.csv | ML_article_script.sh | ML_prepare_data.py | articles_trans

Delimiter: ,

	id_str	tweet	paragraph	label	language
1	745590311120932864	shut the fuck up!! Ronaldo is wayyyyy better See-no-evil monkey Messi is selfish and a di…	Yes.	0	en
2	745590311120932864	黙ってろ!!ロナウドの方が断然上手い 見ざる猿 メッシは利己的でクソ頭	はい。	0	ja
3	745590311120932864	¡¡cierra la puta boca!! Ronaldo es mucho mejor El mono que no ve el mal Messi es egoíst…	Sí.	0	es
4	745590311120932864	shut the fuck up!! Ronaldo is wayyyyy better See-no-evil monkey Messi is selfish and a di…	He once mentioned h…	0	en
5	745590311120932864	黙ってろ!!ロナウドの方が断然上手い 見ざる猿 メッシは利己的でクソ頭	彼はかつて、2008年…	0	ja
6	745590311120932864	¡¡cierra la puta boca!! Ronaldo es mucho mejor El mono que no ve el mal Messi es egoíst…	Una vez mencionó qu…	0	es
7	745590311120932864	shut the fuck up!! Ronaldo is wayyyyy better See-no-evil monkey Messi is selfish and a di…	He criticized his team…	0	en
8	745590311120932864	黙ってろ!!ロナウドの方が断然上手い 見ざる猿 メッシは利己的でクソ頭	アトレティコ・マド…	0	ja
9	745590311120932864	¡¡cierra la puta boca!! Ronaldo es mucho mejor El mono que no ve el mal Messi es egoíst…	Criticó a sus compañ…	0	es
10	745590311120932864	shut the fuck up!! Ronaldo is wayyyyy better See-no-evil monkey Messi is selfish and a di…	Yet, his teammates mi…	0	en
11	745590311120932864	黙ってろ!!ロナウドの方が断然上手い 見ざる猿 メッシは利己的でクソ頭	しかし、彼のチーム…	0	ja
12	745590311120932864	¡¡cierra la puta boca!! Ronaldo es mucho mejor El mono que no ve el mal Messi es egoíst…	Sin embargo, sus co…	0	es
13	745590311120932864	shut the fuck up!! Ronaldo is wayyyyy better See-no-evil monkey Messi is selfish and a di…	Because they know R…	0	en
14	745590311120932864	黙ってろ!!ロナウドの方が断然上手い 見ざる猿 メッシは利己的でクソ頭	それは、ロナウドが…	0	ja
15	745590311120932864	¡¡cierra la puta boca!! Ronaldo es mucho mejor El mono que no ve el mal Messi es egoíst…	Porque saben que Ro…	0	es
16	745590311120932864	shut the fuck up!! Ronaldo is wayyyyy better See-no-evil monkey Messi is selfish and a di…	It is true he will proba…	0	en
17	745590311120932864	黙ってろ!!ロナウドの方が断然上手い 見ざる猿 メッシは利己的でクソ頭	確かに、彼は自分を…	0	ja
18	745590311120932864	¡¡cierra la puta boca!! Ronaldo es mucho mejor El mono que no ve el mal Messi es egoíst…	Es cierto que probabl…	0	es

3. Showing a snapshot of our modelling process for Article Level Data (Tweet Only) Experiment. Starting with the Data Preparation Part and running the model training part:

```
[1]: run ML_prepare_data.py --csv-file ../Data/articles_translated.csv --level article --output-dir ../Dataloaders/

[2]: run ML_train.py --data-dir ../Dataloaders/ --level article --output-dir ../Output/
```

Now we will look into the modeling and testing part of few of the experiments.

4. Shows a snapshot of our modeling process for Article Level Data (Tweet Only) Experiment. It shows the EPOCH 1 for the Article Level.

```
======== Epoch 1 / 6 ========
Training...
  Batch    40  of    750.    Elapsed: 0:00:25.
  Batch    80  of    750.    Elapsed: 0:00:47.
  Batch   120  of    750.    Elapsed: 0:01:10.
  Batch   160  of    750.    Elapsed: 0:01:32.
  Batch   200  of    750.    Elapsed: 0:01:55.
  Batch   240  of    750.    Elapsed: 0:02:17.
  Batch   280  of    750.    Elapsed: 0:02:40.
  Batch   320  of    750.    Elapsed: 0:03:03.
  Batch   360  of    750.    Elapsed: 0:03:25.
  Batch   400  of    750.    Elapsed: 0:03:48.
  Batch   440  of    750.    Elapsed: 0:04:10.
  Batch   480  of    750.    Elapsed: 0:04:33.
  Batch   520  of    750.    Elapsed: 0:04:55.
  Batch   560  of    750.    Elapsed: 0:05:18.
  Batch   600  of    750.    Elapsed: 0:05:41.
  Batch   640  of    750.    Elapsed: 0:06:03.
  Batch   680  of    750.    Elapsed: 0:06:26.
  Batch   720  of    750.    Elapsed: 0:06:48.

  Average training loss: 0.35
  Training epoch took: 0:07:05

Running Validation...
  Validation Loss: 0.22
  Validation Accuracy: 0.91
  Validation Precision: 0.82
  Validation Recall: 0.81
  Validation F1-Score: 0.78
  Validation took: 0:00:15
```

5. Showing a snapshot of our modelling process for Article Level Data (Tweet Only) Experiment. It shows the EPOCH 6, Validation and Test for the Article Level.

```
======== Epoch 6 / 6 ========
Training...
  Batch    40  of    750.    Elapsed: 0:00:23.
  Batch    80  of    750.    Elapsed: 0:00:45.
  Batch   120  of    750.    Elapsed: 0:01:08.
  Batch   160  of    750.    Elapsed: 0:01:30.
  Batch   200  of    750.    Elapsed: 0:01:53.
  Batch   240  of    750.    Elapsed: 0:02:15.
  Batch   280  of    750.    Elapsed: 0:02:38.
  Batch   320  of    750.    Elapsed: 0:03:01.
  Batch   360  of    750.    Elapsed: 0:03:23.
  Batch   400  of    750.    Elapsed: 0:03:46.
  Batch   440  of    750.    Elapsed: 0:04:08.
  Batch   480  of    750.    Elapsed: 0:04:31.
  Batch   520  of    750.    Elapsed: 0:04:53.
  Batch   560  of    750.    Elapsed: 0:05:16.
  Batch   600  of    750.    Elapsed: 0:05:38.
  Batch   640  of    750.    Elapsed: 0:06:01.
  Batch   680  of    750.    Elapsed: 0:06:24.
  Batch   720  of    750.    Elapsed: 0:06:46.

  Average training loss: 0.01
  Training epoch took: 0:07:03

Running Validation...
  Validation Loss: 0.14
  Validation Accuracy: 0.98
  Validation Precision: 0.91
  Validation Recall: 0.98
  Validation F1-Score: 0.94
  Validation took: 0:00:15

Training complete!
Total training took 0:43:49 (h:mm:ss)

[3]: run ML_test.py --data-dir ../Dataloaders/ --trained-model-dir ../Output/ --output-dir ../Output/

  Validation Loss: 0.11
  Validation took: 0:00:36
{'Precision Class 0': 1.0, 'Recall Class 0': 0.98, 'F1-Score Class 0': 0.99, 'Precision Class 1': 0.93, 'Recall Class 1': 0.98, 'F1-Score Class 1': 0.96, 'Weighted F1-Score': 0.98}
```

6. Similarly shows the final result we received on Article Level Data (Tweet and Article) Experiment. It shows the Test metrics for the Article Level.

```
[4]: run ML_test.py --data-dir ../Dataloaders/ --trained-model-dir ../Output/ --output-dir ../Output/

      Validation Loss: 0.14
      Validation took: 0:00:37
    {'Precision Class 0': 0.99, 'Recall Class 0': 0.98, 'F1-Score Class 0': 0.99, 'Precision Class 1': 0.92, 'Recall Cla
    ss 1': 0.98, 'F1-Score Class 1': 0.95, 'Weighted F1-Score': 0.98}
```

7. Similarly showing the final result we received on Paragraph Level Data (Tweet Only) Experiment. It shows the Test metrics for the Paragraph Level.

```
[3]: run ML_test.py --data-dir ../Dataloaders/ --trained-model-dir ../Output/ --output-dir ../Output/

      Validation Loss: 0.15
      Validation took: 0:00:27
    {'Precision Class 0': 0.99, 'Recall Class 0': 0.98, 'F1-Score Class 0': 0.99, 'Precision Class 1': 0.69, 'Recall Class 1': 0.78, 'F1-Score Class
    1': 0.73, 'Weighted F1-Score': 0.98}
```