

Data Annotation and Analysis of Fine-tuned twitter-roberta-base-sentiment model on the Uncleaned and Cleaned data of the hatespeech

Sai Hemanth Kilaru,
skilaru@arizona.edu

– > [GitHubRepository](#) < –

Abstract

This research analyses and evaluates the performance of the twitter-roberta-base-sentiment model, one of the best transformer-based model fine-tuned for hate speech detection, using both uncleaned and cleaned versions of a dataset which is created and annotated by myself Using a zero-shot methodology. Here, we observe the model's ability to predict hate speech without the need for any domain-specific training or fine-tuning. The model's performance was evaluated using metrics such as accuracy and F1-score. The results show that good data, data-driven text cleaning, significantly improves the model's performance on the hate speech detection task than the raw data, demonstrating the importance of data quality for the machine learning tasks.

1 Introduction

Detecting the Hate/Offensive speech has become an essential task for moderating content on social media platforms such as X,instagram,etc.The rise of this kind of harmful speech online has led to the development of various machine learning models that aim to detect and flag these hate speech in real time cases like the tweets/comments etc.. While pre-trained models like twitter-roberta-base-sentiment model have shown very good prediction results in this area, their performance can vary significantly depending on the quality of the data they are being tested on. This study shows how the twitter-roberta-base-sentiment model performs on both uncleaned and cleaned datasets. The main goal is to evaluate how much good data and it's observed text cleaning impacts model performance, especially in the case of zero-shot approach, where the model has not been fine-tuned on the specific task at hand. In this project, we use two datasets: one with raw, uncleaned data and the other cleaned

to remove irrelevant content, such as links and special characters and evaluates the model performance on both of these using accuracy and F1 scores.

2 Challenges Faced

During the project,several challenges were faced that impacted both the performance and compatibility.One of the main difficulties was selecting proper tweets from the X website. I have used x search for few rows and searched for few hashtags to collect the proper suitable data which i can label it according to mine and also i made sure that the proper data sampling is maintained between my '0' and '1'labels.any Another challenge came, particularly when extracting the scores for LABEL 1(NEUTRAL),LABEL 0(NEGATIVE/HATE/OFFENSIVE) of the language model . This resulted in occasional indexing errors,data type errors,etc.I have solved that by careful error analysis and looked for the issue in the web sources like stackoverflow. Selecting the right pretrained language model was also a challenge, as I chose the normal hate-BERT model earlier , which resulted in very poor accuracy scores and model didn't even properly analysed my given data and labels . Then , i looked for the pre-trained hate speech model on the web and found the twitter-roberta-base-sentiment model,and i chose it as the model's pre defined labels matched my intended labels and the model gave very good accuracies through my zero-shot approach (but not 100).

3 Methodology

Here is the methodology for my entire project:

3.1 Data Annotation

The first step was the annotation of the dataset. I have collected 200 tweets from the official website of X(twitter formerly), ensuring a balanced distribution of hate/offensive speech and neutral speech

labels. The dataset was manually annotated into two labels based on my annotation guidelines, they are:

- '1': HATE/OFFENSIVE: Tweets containing harmful or offensive language.
- '0': NEUTRAL: Tweets without any offensive language.

I, as Annotator-1, annotated the dataset, while Annotator-2 (my roommate, Sriram Theerdh Manikyala), also followed the same annotation guidelines, annotated the data. This annotation process was carried out to ensure that the dataset was labeled perfectly according to the guidelines. The labels were applied to each tweet by carefully reviewing the content for any form of offensive or hate speech.

3.2 Inter-Annotator Agreement Evaluation (Cohen's Kappa Score)

To assess the consistency between the annotations made by Annotator-1 and Annotator-2, I calculated the Cohen's Kappa score as it is a scikit-learn metric, the process was simple. This score measures the level of agreement between the two annotators. It yielded a score of 0.9099, indicating almost perfect agreement between the two annotators. This confirms the reliability of the labeled dataset and gives a sign to proceed to further steps. So, from the mutual agreement of the annotators, the gold labels were made as the new column and those labels are the final labels for my data.

3.3 Data Cleaning

In our case, based on the clear observation of the data, my data doesn't contain any null/weak columns. So, I proceeded with the text cleaning. This involved removal of unnecessary things like string(LINK), url's, @mentions and the case-insensitive scenarios of the words, etc. They were done in order to get the cleaned dataset for my project and safely stored the clean rows in a new data file. I haven't removed the hashtags because few contain the hate words; which might disturb the model's predictions, so I kept them.

3.4 Model Setup and Zero-shot training

After searching for the hate speech models in the hugging face, I found a model named 'cardiffnlp/twitter-roberta-base-sentiment' from the hugging face, in which the labels almost

matches my data labels and after thorough study of the model documentation, I have finalized this model for the classification task. Here, without any fine tuning on the specific labels. The model was directly fed with my 150 rows considering its own pre-defined labels named 'LABEL1(NEUTRAL)', 'LABEL0(NEGATIVE/HATE/OFFENSIVE)' (similar to my labels). It generated the hate scores and non hate scores for each row based on that. If the hate score > non hate score, then it will be labelled as '1' else it will be labelled '0'; this matches my style of labelling and stored those predicted labels in a new dataframe and saved that in a dataset for both the uncleaned and cleaned data.

3.5 Model Evaluation and comparison

After getting the model's labels, my labels and model's labels get compared and the evaluation metrics are generated for both the uncleaned and the cleaned data.

4 Evaluation Metrics

To evaluate the performance of the model on the datasets, two primary metrics were used:

- **Accuracy:** The proportion of correct predictions (i.e., how many times the predicted label matched the true label) is calculated.
- **F1 Score:** The F1 score balances both the precision and recall, was used to evaluate the model's ability to classify both the labels correctly. This metric is particularly useful in binary classification tasks, where class imbalance might disturb the results.

Both metrics were calculated for the uncleaned and cleaned datasets, and the model's performance is compared on both the cases.

5 Results and Discussion

The performance of the twitter-roberta-base-sentiment model was evaluated on both uncleaned and cleaned datasets. The scores are generated on the basis of my 200 samples. The table below shows the results based on Accuracy and F1 Scores:

The results demonstrate that the cleaned dataset slightly performs better than the uncleaned d

The results demonstrate that the cleaned dataset performs slightly better than the uncleaned

Metric	Uncleaned Dataset	Cleaned Dataset
Accuracy	0.920	0.925
F1 Score	0.9191	0.9238

Table 1: Comparison of model accuracies and F1 scores for cleaned and uncleaned datasets

data.so, this shows that proper data cleaning plays a key role in the model evaluation. However, during the evaluation, I encountered indexing errors and unexpected output structures when extracting scores for the labels. These errors typically occurred when the output from the model was not consistent. To address this, I implemented error-handling mechanisms, such as the try-except block, to catch such issues and ensure that the program could continue running smoothly without crashing. This helped resolve inconsistencies and allowed me to proceed with the analysis. After overcoming such kind of issues, I successfully finished my project.

6 Conclusion

This study evaluated the performance of the twitter-roberta-base-sentiment model for hate speech prediction using a zero-shot approach on both uncleaned and cleaned datasets. The results showed a clear improvement in performance after cleaning the data. For the uncleaned dataset, the model achieved an accuracy of 0.920 and an F1 score of 0.9191. After data cleaning, the model's performance slightly increased, with an accuracy of 0.925 and an F1 score of 0.9239.

These results highlight the significant impact of model selection and data cleaning on model performance on data, demonstrating the importance of proper research before selecting a model for your task and then removing irrelevant elements, such as mentions and URLs, helps the model to focus only on the content that matters. While the zero-shot learning approach provided useful insights, further improvements could be made through fine-tuning the model or employing more advanced data preprocessing techniques. Overall, this study reinforces the importance of clean, well-annotated data in achieving reliable results in hate speech prediction tasks and model selection phase also matters in the instances like this.

References

- [1] tweets. Retrieved from <https://x.com/home>
- [2] Hate-speech-cardiffnlp. *twitter-roberta-base-sentiment model*. Retrieved from <https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment>
- [3] Yash Jain. (2021). *Text Cleaning Using Regex in Python*. Retrieved from <https://medium.com/@yashj302/text-cleaning-using-regex-python-f1dded1ac5bd>