# Towards Interpretable Detection of Hate Speech in Twitter

**Yi-Ju Lin**
Georgetown University
yl1290@georgetown.edu

## 1 Introduction

With the growing volume of social media, the detection of hatred content online has become more critical. While most of the research (Liu et al., 2019; Seganti et al., 2019) in online hate speech detection has focused on using large scale pre-trained language model to achieve higher performance, explainable AI is getting considerable attention recently. As noted by Bender et al. (2021), researchers have to be aware of the potential risk of these big models (e.g. BERT), including the unintended bias (Sap et al., 2019) picked up by the model during the training process. To deal with the problem of lacking interpretability, common ways in hate speech detection include explaining model's predictions (Belinkov and Glass, 2019) or using models that can produce more interpretable results (e.g. Logistic Regression, SVM) (Waseem and Hovy, 2016; Xiang et al., 2021). Other approaches (Xiang et al., 2021) identified toxic spans (tokens) that give rise to the toxicity in the text, which explores more on how machines make decisions.

The main goal of this study is to improve the interpretability of hate speech detection model. To this end, a logistic regression model that could produce more interpretable results will be built. Apart from that, a public hate speech dataset (OLID (Zampieri et al., 2019a)) that contains multiple aspects of hate speech annotations is used as training data to provide more explainability to hate speech. More importantly, by taking advantage of the transparency of tf-idf vectors, this study aims to understand the reasoning behind model's prediction by investigating significant tokens in a tweet that can affect model's decisions. I hope this approach could provide insights into machine's understanding of hate speech.

## 2 Related Work

As online content continues to grow, so does offensive content in social media. Therefore, the detection of online hate speech has become a topic of immense interest in recent years.

Most of the work was done to improve model performances. For example, one of the top-performing system description papers (Liu et al., 2019) in SemEval-2019 Task 6 (Zampieri et al., 2019b) focused on applying deep learning as well as ensemble models to deal with the task, providing approximately 82% of F1 score on capturing the presence of hate speech in tweets. Although the result looks amazing, it may not be as exciting as it sounds like when we consider the fact that the baseline approach (choosing all predictions to be non-offensive) already yields 42% on F1 score. Furthermore, the official task report of SemEval-2019 Task 6 (Zampieri et al., 2019b) also showed that a traditional machine learning approach (SVM) produces nearly 70% of F1 score on the same task, suggesting that a simple model is effective enough to provide competitive results. To this end, I see an advantage of using a simpler, yet equally effective model in dealing with hate speech.

Another advantage of using a simpler model rather than state-of-the-art deep learning methods is that the latter ones are said to suffer from the problem of lacking interpretability. Although it has been shown in the literature (Zampieri et al., 2020) that contextual word embedding like BERT (Devlin et al., 2018) is advantageous in hate speech detection, there are debates (Bender et al., 2021) on the risks of using these large language models, including the possibility of these large and unfathomable models picking up bias (Sap et al., 2019). To provide interpretability of the model, traditional ways include explaining prediction results (e.g. plotting confusion matrix). Another

simple yet effective way involves using models that are considered to be highly interpretable (e.g. Logistic Regression, SVM) (Waseem and Hovy, 2016; Xiang et al., 2021) rather than large but unfathomable language models like BERT or deep learning methods. On hate speech detection, recent approach (Xiang et al., 2021) involves identifying toxic spans (at the token level) in a text. In the present study, in addition to common ways (error analysis of the model, confusion matrix) of dealing with the problem, the spans (tokens in this study) that can significantly influence model's judgments were also analyzed to help understand model's predictions.

As highlighted in Mathew et al. (2020), providing hate speech on a more comprehensive level in training data helps improve model's performance and reduce unintended bias. While most of the work in hate speech detection (Davidson et al., 2017; Waseem and Hovy, 2016) has focused on identifying potentially offensive messages, datasets for hate speech with hate type (targeted or untargeted) and hate target annotations that provide more explainability to hate speech are indeed few. The dataset used in this study is OLID (Offensive Language Identification Dataset) (Zampieri et al., 2019a), the first large-scale dataset of English tweets with high-quality annotation of the target and type of offenses. Taking advantage of a hierarchical annotation schema, OLID provides annotations on the presence, categories, and target of hate speech at one time, offering a more generic picture of hate speech.

# 3 Methodology

## 3.1 Dataset

Systems are trained and evaluated on OLID (Offensive Language Identification Dataset), which is an official dataset used for SemEval-2019 Task 6 that contains 14,100 English tweets annotated with ground truth offense types and offense target labeled by human annotators. All the tweets are annotated following a hierarchical annotation, as each tweet contains up to 3 levels (or subtasks), where level A is to determine whether the content is offensive language (OFF) or not (NOT), level B is to identify whether the tweet is targeted insults (TIN) or untargeted (UNT), and finally level C is to see either the post is targeting at individual (IND), Group (GRP), or others (OTH). Only posts that are labeled as OFF in level A would be included in level B, and only those labeled as TIN in level B would be passed on to level C. Due to the limited scope of this study, I will only use data from level A and B, which consists of 13,240 offensive posts (further split up to 4,400 targeted posts and 8,840 untargeted posts) and 860 non-offensive posts.

## 3.2 Pre-processing

In order to see the effect of different pre-processing techniques on the model, a variety of pre-processing steps on the tweets are performed. First, all the tweets are lower-cased and tokenized. Next, `sklearn` collections of English stop words are used and removed them from the tokens. I also experimented on keeping the stop words. Interestingly, the result shows that task B (determining whether the tweet is targeted or untargeted) benefits from keeping stop words, while task A (whether the tweet is offensive) benefits from removing stop words. My hypothesis is that certain stop words (e.g. you, they) are crucial in predicting whether the post is a targeted insult, whereas they may be irrelevant in determining whether the post is offensive or not.

## 3.3 Model

For this study, Logistic Regression (LR) with tf-idf vectors is used for evaluation and interpretability evaluation. The model is built with the solver function set with `liblinear`, which is the best choice for a small dataset like OLID. LR model is being considered as a highly interpretable model (Xiang et al., 2021) that transparently assigns scores to each input feature that can be used to justify the model's decision (see Section 6 for how LR could be used with tf-idf features). Although there are other state-of-the-art models that use attention mechanisms to make the models more explainable (Rogers et al., 2020), LR is preferred here because previous literature (Waseem and Hovy, 2016; Xiang et al., 2021) has proved LR to be effective in offensive language detection. Furthermore, unlike LR that naturally provides explicit explanations for the predictions, attention mechanisms usually need post-processing to perform interpretation, which is less straightforward and intuitive compared to LR.

# 4 Results

To evaluate the effectiveness of LR models, precision (P), recall (R), and F1 score (F) are considered. Table 1 shows the performance of the

proposed LR model compared to the baseline in Semeval-2019 task 6 (Zampieri et al., 2019b). As can be seen from Table 1, LR model provides satisfying results, given that it is a simple model without any fine-tuning on the dataset. Interestingly, micro average scores perform better than macro scores. This can be explained by the fact that the dataset is strongly imbalanced (Zampieri et al., 2019b), with offensive tweets being almost twice the number of non-offensive tweets. The relatively well performance of micro average scores suggests that the LR model performs better in the majority class than in the small class, as micro-average score gives more weight to the majority class.

### 4.1 Exploring on different features

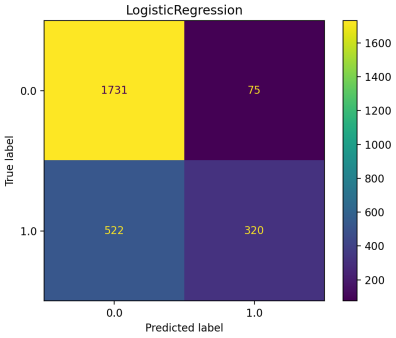
In addition to tf-idf vectors, pre-extracted features such as "sentiment", "subjectivity", and "profanity usage" are added into the model for Task A. The effect of "occurrence of @USER" is also investigated in Task B. The results of the model after adding these features are shown in Table 2. From Table 2, almost no improvement (slight improvement, if any) on the LR model is observed in Task A after adding pre-extracted features in it. This is expected, as the python package used to calculate profanity usage, `profanity-check` [1], is said to have a hard time picking up on variants of swear words like "f4ck you" or "you b1tch", which is relatively common in Twitter. In addition, the subjectivity tool, `TextBlob` [2], is trained on a large corpus of IMDb movie reviews, meaning that it is not appropriate for analyzing subjectivity in tweets.

In contrast, there is a slight increase in F1 score in Task B after adding user name features in the model. The feature is calculated by counting the presence of "@USER" in a tweet and then divided by the word count of the tweet, showing the "percentage" of @USER in a tweet. This result aligns with our assumption, which believes that people would usually mention the name of an individual or the group if they want to attack or insult someone.

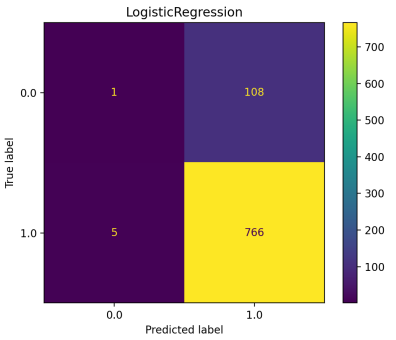
## 5 Error Analysis

In order to understand the errors LR models make, The predictions against the gold labels are analyzed both quantitatively and qualitatively. First, the confusion matrix of task A and B are shown in Figure 1a and 1b.

[1]https://github.com/vzhou842/profanity-check
[2]https://github.com/sloria/TextBlob




(a) Confusion matrix for task A




(b) Confusion matrix for task B

Figure 1: Confusion matrix for both tasks

As can be seen from Figure 1a, our LR model makes much more Type II error (False Negatives) than Type I errors (False Positives) in task A, which is reflected in high precision and rather low recall in Task A (see Table 1). This means that our model did a great job on correctly identify hateful tweet out of all the "real" offensive content. Conversely, our model in Task B is "sensitive" enough to capture all the targeted hateful content in the real world, as reflected in the low precision and rather high recall in Table 1. In order to find out the reason for this difference, it is necessary to take a closer look at where and why the models make mistakes.

**False Negatives** Hate speech detection could be hard for humans, not to mention machines. Such cases include hateful content in subtle form, for example, hate speech that requires outside context or those that do not contain explicit words that express hatefulness. Consider those real examples from OLID dataset:

- *@USER @USER @USER i have heard he is making waves*
- *@USER That's what happens when you let a liberal get a hold of a gun. That would be a*

| | Task A | | | |
|---|---|---|---|---|
| | P | R | F | Baseline |
| NOT | 0.77 | 0.96 | 0.85 | |
| OFF | 0.81 | 0.38 | 0.52 | |
| Micro av. | 0.77 | 0.77 | **0.77** | |
| Macro av. | 0.79 | 0.67 | 0.69 | 0.42 (ALL NOT)/ 0.22 (ALL OFF) |
| Weight av. | 0.78 | 0.77 | 0.75 | |
| | Task B | | | |
| | P | R | F | Baseline |
| TIN | 0.88 | 0.99 | 0.93 | |
| UNT | 0.17 | 0.01 | 0.02 | |
| Micro av. | 0.87 | 0.87 | **0.87** | |
| Macro av | 0.52 | 0.50 | 0.47 | 0.47 (ALL TIN)/ 0.1 (ALL UNT) |
| Weight av. | 0.79 | 0.87 | 0.82 | |

Table 1: Evaluation results for LR model, compared to baseline macro F1 score provided by SemEval-2019

| | Task A |
|---|---|
| Base | 0.7745 |
| add sentiment | -0.0056 |
| add subjectivity | -0.0094 |
| add profanity | +0.0046 |
| | Task B |
| Base | 0.8716 |
| add count(@USER) | +0.0216 |

Table 2: Micro average F1 score of LR after adding additional features

*good place to start gun control. Ban liberals from getting guns*

- *More bad news for Democrats MAGA URL*

The first example does not contain any explicit word/phrases that would be considered as hateful. In fact, this is an ambiguous case even for humans, as the speaker may not even intend to be toxic. The hatred in the second and third example lies in the context outside, which may be easy for humans but hard for machines to detect.

**False Positives** A major source of False Positives comes from taking negative words (e.g. gun, damn, killing, hype) as strong indicator of hate speech, which is often not the case in real world. For example:

- *@USER Reagan also signed the first gun control bill as governor of CA*
- *10th duet with my cutiee angel @USER is damn awesomee in this her expressions just killing love to do this only for u #mybubblygirl #anupama #dreamgirl #BCO love u my cutiee doll*
- *@USER Dont believe the hype*

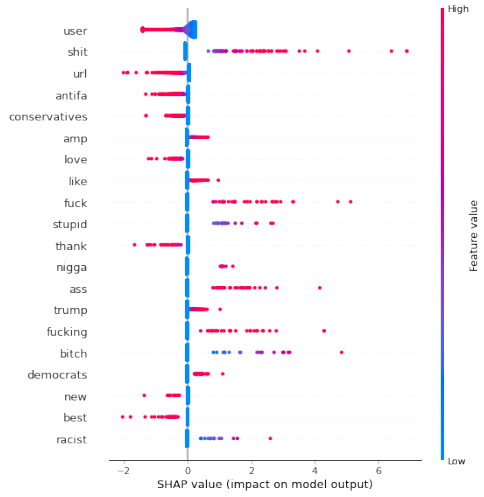
The first and the third example are neutral descriptions without malicious intent. However, machines treat them as hate speech by overestimating the negative effect of the word "gun" in "gun control" or the negative word "hype". The second tweet is a good example of how machines mistreat the seemingly negative words in the sentence, where they are in fact used to emphasize something.

To sum up, the rather low recall of task A could be attributed to the various forms of hate speech, including those in subtle form. This has made capturing all forms of hate speech more difficult. As

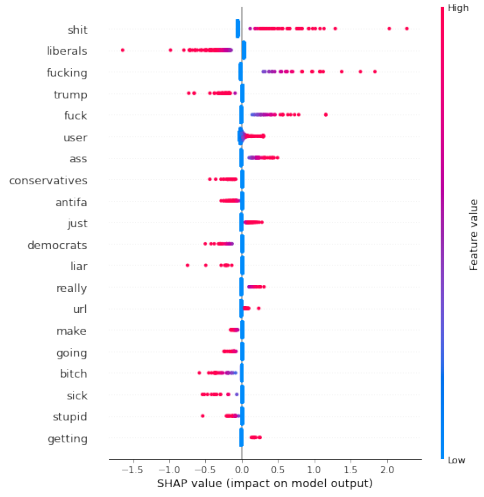
for the low precision of task B, one of the reasons may come from overestimating the negative effect of the negative word.

## 6 Interpretability

In this section, the interpretability of our model would be examined. Since tf-idf vectors transparently assign tf-idf scores to each input feature (tokens in this case), the relative importance of the token when making a prediction could be analyzed through a python package called `shap` [3]. As a result, we could explore the significant tokens in a tweet that can affect model's prediction. Figure 2a and 2b show us how a single feature (or word) affects the output of the model. The color represents the feature value (red high, blue low), the higher the value, the more important it is in affecting model's output. In Figure 2a, positive shap value in X axis means the features are actually helping raise the chance of a hateful tweet, while the negative features are lowering the chance. In Figure 2b, positive shap value means the features help raise the chance of untargeted tweet, while negative shap values raise the chance of targeted tweet.




(a) Feature impact on LR model (task A)




(b) Feature impact on LR model (task B)

Figure 2: Feature impact on LR model

The result shown here is interesting, in that we can see how each word in the tweet contributes to the model's output. Figure 2a tells us the features the model relies on to determine whether a tweet is offensive. It is revealed in the plot that most of the positive (or neutral) words like "user", "URL", "conservative", "love", "thank", "new", "best" are important in predicting non-offensive tweet, while the swear words shown in the plot help predict offensive tweet. Figure 2b shows that the model depends on curse words to predict an untargeted offensive tweet. On the other hand, possible target of offense like "Trump", "democrats", "liberals", "conservatives", "liar", "sick", "stupid" are shown to be effective in predicting targeted offensive tweet.

Taken together, our results show that explicit features such as swear words and positive words are relied on for machines to make predictions. While this may help machines to achieve satisfying results (as shown in Table 1), it is dangerous for machines to solely rely on these explicit features to make predictions, as we may miss some hate speech in subtle form (as discussed in Section 5).

## 7 Conclusion

In this paper, a simpler hate speech detection model that produces more interpretable results is built. This approach is proved to yield competitive results compared to the benchmark results set for the same task in SemEval 2019. By taking advantage of the transparency of tf-idf vectors, a more explainable form of text embedding than pre-trained word embeddings (word2vec, GloVe, BERT), I generated better explanations of the significant spans (tokens) in a text that can affect model's prediction. This provides more explainability to the results, improving our understanding of how models identify hate speech. As shown in the error analysis, however, our simple model is not able to capture all forms of hate speech, including those that require outside context to understand or those in subtle form.

[3] https://github.com/slundberg/shap

This suggests that investigating models or methods that can take implicit hate speech into account is a promising line of research, which is the future direction for this study.

## References


Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ping Liu, Wen Li, and Liang Zou. 2019. Nuli at semeval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 87–91.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1668–1678.

Alessandro Seganti, Helena Sobol, Iryna Orlova, Hannam Kim, Jakub Staniszewski, Tymoteusz Krumholc, and Krystian Koziel. 2019. Nlpr@ srpol at semeval-2019 task 6 and task 5: Linguistically enhanced deep learning offensive sentence classifier. *arXiv preprint arXiv:1904.05152*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Tong Xiang, Sean MacAvaney, Eugene Yang, and Nazli Goharian. 2021. Toxccin: Toxic content classification with interpretability. *arXiv preprint arXiv:2103.01328*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.