Building a smart rejector for detecting hate speech

Philippe Lammerts

Building a smart rejector for detecting hate speech

by

Philippe Lammerts

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday January 1, 2013 at 10:00 AM.

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TUDelft

Abstract

Acknowledgements

Philippe Lammerts
Delft, January 2013

Contents

1

Introduction

The amount of hateful content spread online on social media platforms remains a significant problem. Ignoring its presence can harm people and even result in actual violence and other conflicts [1, 4]. There are many news articles about events where hate spread on online platforms lead to acts of violence [9, 12–14]. One research paper found a connection between hateful content on Facebook containing anti-refugee sentiment and hate crimes against refugees by analyzing social media usage in multiple municipalities in Germany [14]. Governmental institutions and social media companies are becoming more aware of these risks and are trying to tackle hate speech. For example, the European Union developed a Code of Conduct on countering illegal hate speech in cooperation with large social media companies such as Facebook and Twitter [2]. This Code of Conduct requests companies to prohibit hate speech and report their progress every year [2]. The most recent report from 2021 stated that Twitter only removed 49.5% of all hateful content on their platform. Facebook is most successful in removing hate speech as they claim to have removed 70.2% of all hateful content in 2021 [2]. However, one article found in internal communication from Facebook that this percentage is much lower, around 3-5% [7]. Therefore, hate speech detection remains a hard problem that even large institutions cannot do well yet.

Currently, people rely on reactive and proactive content moderation methods to detect hate speech [10]. Reactive moderation is when social media users are flagging (also known as reporting) hateful content [10]. Proactive moderation is either done automatically using detection algorithms or manually by a group of human moderators [10]. There exist different methods for automatically detecting hateful content. Most use Machine Learning (ML) algorithms since these tend to be the most promising for their detection performance at a large scale [4, 6]. These algorithms can range from traditional ML methods such as Support Vector Machine or Decision Tree to Deep Learning algorithms [6].

However, both proactive and reactive moderation methods have their limitations. Proactive manual moderation of hateful content is still the most reliable solution but is simply infeasible due to the large amount of content generated by the many users [4]. Reactive moderation solves this problem since the users can report hate speech themselves. Although, the problem stays that hateful content is exposed to the users for some time. Proactive automatic moderation using automated detection algorithms allow for large amounts of data to be checked quickly without the involvement of humans. However, these algorithms can be unreliable as they often perform poor on deployment data [4]. Furthermore, one paper found that most research in hate speech detection overestimates the performance of the automated detection methods [3]. The authors found that the performance drops significantly when the detection algorithms are trained on one dataset and evaluated on another [3].

This thesis research will tackle the problems of proactive moderation by creating a *human-machine co-creation* [16] system that combines the advantages of both humans (cognitive abilities and ability to make judgements) and machines (automation and performance). A system where humans and machines work together to detect hate speech more effectively. This system is a *machine-assisting-human* system where ML models help humans detecting hateful content automatically and where hu-

mans can make the final decisions (*human-in-the-loop*) when the model is not confident enough [16]. Here come ML models with a reject option in place. The goal of the reject option is to reject an ML prediction when the risk of making an incorrect prediction is too high and to defer the prediction task to a human [8]. There are several advantages. First, the utility of the ML increases as only the most confident (and possibly the most correct) predictions are accepted. Second, less human effort is necessary as the machine is handling all predictions tasks, and only a fraction needs to be checked by a human. To the best of our knowledge, ML with rejection has not been used in hate speech detection before. Therefore, the goal of this thesis project is to build the first *smart rejector for detecting hate speech*. This leads to our main research question:

> **RQ** How can we apply Machine Learning with a reject option to detect hate speech more effectively?

The idea of most ML models with rejection is that we reject predictions when the model's confidence is too low. However, we need to tackle two underlying problems to be able to answer our main research question. First, we need to figure out when the ML model is not confident enough by measuring the effectiveness of the reject option according to the task of hate speech detection. Second, we need to determine how we can detect the high confidence errors.

The first problem is about the trade-off between how much we trust the predictions produced by the ML model and how much we involve humans to make the judgements. There are gains of accepting correct predictions, costs of accepting incorrect predictions, and costs of rejecting samples. More specifically, we should weigh cost values for False Negative (FN), labelling something as non-hateful when it is, and False Positive (FP) predictions, labelling something as hateful when it is not, according to the task [15]. So, we first need a metric that measures the cost-effectiveness of the reject option. We can use the resulting metric to determine when to reject/accept predictions by maximizing cost-effectiveness. Second, we need to find out how we can define these cost values in the context of hate speech detection. Therefore, our first sub research question is as follows:

> **SRQ1** How can we determine when the Machine Learning model is not confident enough?
> - **SRQ1.1** How can we measure the cost-effectiveness of the reject option?
> - **SRQ1.2** How can we determine the costs of rejections and True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) predictions?

The second problem is about detecting the high confidence errors. These high confidence errors are also called *unknown unknowns* [11]. When we would only rely on the confidence of the ML model to determine when to reject/accept predictions, then we would accept a subset of incorrect predictions with high confidence. We need to find a way to recognize these unknown unknowns. Once detected, we can reject these samples so that a human moderator makes the final judgement. Doing so would further improve the effectiveness of our smart rejector for detecting hate speech. So, our second sub research question is as follows:

> **SRQ2** How can we detect the unknown unknowns?

Here comes a list of contributions

Here comes a short description of the structure of the thesis report

2

Related work

This section gives some background information about ML with rejection, hate speech detection, and unknown unknown detection

2.1. Machine Learning models with rejection

Explain the different architectures of ML with rejection

Explain the different types of confidence metrics

Provide examples of ML models with rejection from other domains

2.2. Hate speech detection

Give some examples of existing hate speech detection methods from literature that for example use traditional Machine Learning algrorithms or Bag of Words

2.3. Unknown unknown detection

Give examples of existing unknown unknown detection methods from literature

3

Methods

3.1. Hate speech detection

3.1.1. Model

Explain the model's architecture using the original paper from Agrawal and Awekar

3.1.2. Calibration

Explain what model calibration is and why it's necessary

3.2. Cost-effectiveness metric
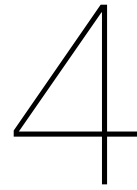
Explain the original metric from De Stefano

Explain and proof our modified version of the metric from De Stefano

3.3. Unknown unknowns

Explain our method for detecting unknown unknowns

Implementation

4.1. System architecture

Explain design of the smart rejector and how the different methods are combined

4.2. Phases

4.2.1. Training

The system will probably support a training and deployment phase. Explain here the training phase of the smart rejector. During this phase, the preparations are done for training the model, determining the optimal rejection threshold, and preparing things for detecting the unknown unknowns

4.2.2. Deployment

Explain how the smart rejector works in the wild and is detecting hate speech in new unlabelled data.

5

Evaluation

5.1. Cost values

5.1.1. Setup

Describe the experimental setup

5.1.2. Method

Explain the method for retrieving the cost values for hate speech detection

5.1.3. Results

5.2. Smart rejector

5.2.1. Setup

Describe the experimental setup

5.2.2. Method

Explain which experiments are conducted

Explain how the results are analyzed. Things to consider: Accuracy-Rejection curves, accuracy of accepted predictions, rejection rates, acception rates

5.2.3. Results

6

Discussion

Answer research questions

The rejection threshold is calculated using the test set. This test set needs to be as realistic as possible.

Hate speech is difficult domain as there tend to be a lot of disagreement between people about what is considered hate speech and what not. Most datasets are binary labeled but perhaps it's better that hate speech datasets use an ordinal scale to define how hateful a text sample is.

Explain difficulties in coming up with numerical cost/gain values of (in)correct predictions and rejections

Discuss future work

7

Conclusion

Bibliography

[1] Hate speech and violence. *European Commission against Racism and Intolerance (ECRI)*. URL `https://www.coe.int/en/web/european-commission-against-racism-and-intolerance/hate-speech-and-violence`. Visited on 19/01/2022.

[2] The eu code of conduct on countering illegal hate speech online. *European Commission*, May 2016. URL `https://ec.europa.eu/info/policies/justice-and-fundamental-rights/combatting-discrimination/racism-and-xenophobia/eu-code-conduct-countering-illegal-hate-speech-online_en`. Visited on 07/03/2022.

[3] Aymé Arango, Jorge Pérez, and Barbara Poblete. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 45–54, 2019.

[4] Agathe Balayn, Jie Yang, Zoltan Szlavik, and Alessandro Bozzon. Automatic identification of harmful, aggressive, abusive, and offensive language on the web: A survey of technical biases informed by psychology literature. *ACM Transactions on Social Computing (TSC)*, 4(3):1–56, 2021.

[5] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 512–515, 2017.

[6] Paula Fortuna and Sérgio Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018.

[7] Noah Giansiracusa. Facebook uses deceptive math to hide its hate speech problem. *Wired*, Oct 2021. URL `https://www.wired.com/story/facebooks-deceptive-math-when-it-comes-to-hate-speech/`. Visited on 07/03/2022.

[8] Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: A survey. *arXiv preprint arXiv:2107.11277*, 2021.

[9] Mathew Ingram. Facebook now linked to violence in the philippines, libya, germany, myanmar, and india. *Columbia Journalism Review*, Sep 2018. URL `https://www.cjr.org/the_media_today/facebook-linked-to-violence.php`. Visited on 07/03/2022.

[10] Kate Klonick. The new governors: The people, rules, and processes governing online speech. *Harv. L. Rev.*, 131:1598, 2017.

[11] Anthony Liu, Santiago Guerra, Isaac Fung, Gabriel Matute, Ece Kamar, and Walter Lasecki. Towards hybrid human-ai workflows for unknown unknown detection. In *Proceedings of The Web Conference 2020*, pages 2432–2442, 2020.

[12] Mujib Mashal, Suhasini Raj, and Hari Kumar. As officials look away, hate speech in india nears dangerous levels. *The New York Times*, Feb 2022. URL `https://www.nytimes.com/2022/02/08/world/asia/india-hate-speech-muslims.html`. Visited on 07/03/2022.

[13] Paul Mozur. A genocide incited on facebook, with posts from myanmar's military. *The New York Times*, Oct 2018. URL `https://www.nytimes.com/2018/10/15/technology/myanmar-facebook-genocide.html`. Visited on 07/03/2022.

[14] Karsten Müller and Carlo Schwarz. Fanning the flames of hate: Social media and hate crime. *Journal of the European Economic Association*, 19(4):2131–2167, 2021.

[15] Burcu Sayin, Jie Yang, Andrea Passerini, and Fabio Casati. The science of rejection: A research area for human computation. *arXiv preprint arXiv:2111.06736*, 2021.

[16] Wai Lok Woo. Future trends in i&m: Human-machine co-creation in the rise of ai. *IEEE Instrumentation & Measurement Magazine*, 23(2):71–73, 2020.