DELFT UNIVERSITY OF TECHNOLOGY

MASTERS THESIS

Value-Sensitive Rejection of Machine Learning predictions for Hate Speech Detection

Author:
Philippe Lammerts

Thesis advisor:
Prof. dr. ir. G.J.P.M. Houben
Delft University of Technology

Daily supervisor:
Dr. J. Yang
Delft University of Technology

Co-daily supervisors:
Dr. Y-C. Hsu
University of Amsterdam

P. Lippmann
Delft University of Technology

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Web Information Systems Group - Crowd Computing
Software Technology

August 9, 2022

DELFT UNIVERSITY OF TECHNOLOGY

Abstract

Electrical Engineering, Mathematics and Computer Science
Software Technology

Master of Science

Value-Sensitive Rejection of Machine Learning predictions for Hate Speech Detection

by Philippe Lammerts

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Contents

List of Abbreviations

BERT	**b**idirectional **e**ncoder **r**epresentations from **t**ransformers
ECE	**e**xpected **c**alibration **e**rror
BOW	**b**ag-**o**f-**w**ords
CNN	**c**onvolutional **ne**ural **n**etwork
DL	**d**eep **l**earning
FN	**f**alse **n**egative
FP	**f**alse **p**ositive
LDA	**l**atent **d**irichlet **a**llocation
LR	**l**ogistic **r**egression
LSTM	**l**ong **s**hort-**t**erm **m**emory
ML	**m**achine **l**earning
POS	**p**art-**o**f-**s**peech
SVM	**s**upport **v**ector **m**achine
TF-IDF	**t**erm **f**requency-**i**nverse **d**ocument **f**requency
TN	**t**rue **n**egative
TP	**t**rue **p**ositive
VSD	**v**alue-**s**ensitive **d**esign

Chapter 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 (Balayn et al., 2021; Council of Europe, n.d.). There are many news articles about events where hate spread on online platforms lead to acts of violence (Ingram, 2018; Mashal et al., 2022; Mozur, 2018; Müller & Schwarz, 2021). 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 (Müller & Schwarz, 2021). Governmental institutions and social media companies are becoming more aware of these risks and are trying to combat 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 (European Commission, 2016). This Code of Conduct requests companies to prohibit hate speech and report their progress every year (European Commission, 2016). 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 (European Commission, 2016). However, one article found in internal communication from Facebook that this percentage is much lower, around 3-5% (Giansiracusa, 2021). Therefore, hate speech detection remains a hard problem that even large institutions have not solved yet.

Currently, people rely on reactive and proactive content moderation methods to detect hate speech (Klonick, 2018). Reactive moderation is when social media users are flagging (also known as reporting) hateful content (Klonick, 2018). Proactive moderation is either done automatically using detection algorithms or manually by a group of human moderators (Klonick, 2018). 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 (Balayn et al., 2021; Fortuna & Nunes, 2018). These algorithms can range from traditional ML methods such as Support Vector Machine or Decision Tree to Deep Learning algorithms (Fortuna & Nunes, 2018).

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 (Balayn et al., 2021). 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 have shown to be unreliable as they often perform poor on deployment data (Balayn et al., 2021; Gröndahl et al., 2018). One study found that the F1 scores reduce

significantly (69% F1 score drop in the worst case) when training a hate speech detection model on one dataset and evaluating it using another dataset (Gröndahl et al., 2018). Furthermore, one paper found that most research in hate speech detection overestimates the performance of the automated detection methods (Arango et al., 2019). The authors found that the performance drops significantly when the detection algorithms are trained on one dataset and evaluated on another (Arango et al., 2019).

This thesis research will tackle the problems of proactive moderation by focusing on the concept of *human-machine co-creation* (Woo, 2020) where the advantages of both humans (cognitive abilities and ability to make judgements) and machines (automation and performance) are combined. So humans and machines should work together to detect hate speech. ML models should detect hateful content automatically and humans should make the final decisions (*human-in-the-loop*) when the model is not confident enough (Woo, 2020). 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 (Hendrickx et al., 2021). There are several advantages. First, the utility of the ML model 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 prediction 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.

In this work, we focus on *value-senstive* rejection. There are gains of accepting correct predictions (positive value) and costs of accepting incorrect or rejecting predictions (negative value). 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, accord- ing to the task of hate speech detection and incorporate them in the design of the hybrid human-AI system (Sayin et al., 2021). We will mainly focus on the human values from the perspective of the social media users since they are the most affected by the consequences of hate speech.

The idea of most ML models with rejection is that we reject predictions when the model's confidence is too low. Therefore, we need a metric that measures the total value of ML models with a reject option. We can use the resulting metric to determine when to reject/accept predictions by maximizing the total value. Second, we need to find out how we can define the user-centred values in the context of hate speech detection. We will attempt to retrieve the value ratios since it is hard to come up with the absolute cost values in the hate speech domain. By value ratios, we mean to figure out, for example, the ratio between an FP and an FN prediction. Therefore, our research questions are as follows:

This leads to the following research questions:

RQ How can we reject predictions of Machine Learning models in a value-sensitive manner for hate speech detection ?

- **SRQ1** How can we measure the total value of Machine Learning models with a reject option?
- **SRQ2** How can we determine the value ratios between rejections and True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) predictions?

Here comes a list of contributions

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

Chapter 2

Related work

In this chapter, we first define hate speech in section 2.1 and explain why it is such a challenging topic to tackle, especially from a computer science perspective. Then, we give an overview of the state-of-the-art solutions for automatic hate speech detection in section 2.2. In section 2.3, we discuss the different methods of ML with rejection. Section 2.4 discusses the main challenges of assessing the values of (in)correct and rejected predictions in the hate speech domain. Finally, we discuss the shortcomings of standard machine metrics in section 2.5, such as accuracy, to evaluate detection systems and why human-centred metrics such as ours are promising.

2.1 Hate speech: definition and challenges

Different types of online conflictual languages exist, such as cyberbullying, offensive language, toxic language, or hate speech, and come with varying definitions from domains such as psychology, political science, or computer science (Balayn et al., 2021). We can broadly define *hate speech* as *"language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group"* (Balayn et al., 2021; Davidson et al., 2017). It differs from other conflictual languages since it focuses on specific target groups or individuals (Balayn et al., 2021).

Balayn et al. (2021) identified the mismatch between the formalization of hate speech and how people perceive it. Many factors influence how people perceive hate speech, such as the content itself and the characteristics of the target group and the observing individual, such as gender, cultural background, or age (Balayn et al., 2021). We can identify this mismatch in other related work from which there appears to be low agreement among humans regarding annotating hate speech (Fortuna & Nunes, 2018; Ross et al., 2017; Waseem, 2016). Ross et al. (2017) found low inter-rater reliability scores (Krippendorff's alpha values of around $0.2 - 0.3$) in a study where they asked humans about the hatefulness and offensiveness of a selection of tweets. They also found that the inter-rater reliability value does not increase when showing a definition of hate speech to the human annotators beforehand. Waseem (2016) found a slight increase in the inter-rater reliability when considering annotations of human experts only, but it remained low overall.

In the hate speech domain, we must be careful with creating biased detection systems trained on biased datasets. Hate speech datasets such as Waseem and Hovy (2016) or Basile et al. (2019) collected their data using specific keywords that can introduce *sample retrieval* bias and annotated their data using only three independent annotators that might result in *sample annotation* bias (Balayn et al., 2021). Automated classification models will likely become biased in their predictions if we train them on biased datasets (the *garbage in, garbage out* principle). This phenomenon becomes most notable when applying pre-trained classification models to new and

unseen data. For example, Gröndahl et al. (2018) and Arango et al. (2019) report significant drops in F1 scores when training a hate speech classification model on one dataset and evaluating it on another. Gröndahl et al. (2018) found that the F1 score reduces by 69% in the worst case and that the model choice does not affect the classification performance as much as the dataset choice. Arango et al. (2019) replicated several state-of-the-art hate speech classification models and found that most studies overestimate the classification performance. These results further strengthen our stance that we should not detect hate speech solely by machines but rather by a human-in-the-loop approach.

2.2 Automatic hate speech detection

This section will list the literature's state-of-the-art Natural Language Processing (NLP) techniques for automatic hate speech detection. Several excellent surveys outlined the different detection methods (Fortuna & Nunes, 2018; Schmidt & Wiegand, 2019). First, we will discuss the different features used in the classification models. Then, we will state the most used classification models ranging from supervised to unsupervised learning.

Commonly used features are bag-of-words (BOW) (Greevy & Smeaton, 2004), character/word N-grams (Waseem & Hovy, 2016), lexicon features (Xiang et al., 2012), term frequency-inverse document frequency (TF-IDF) (Badjatiya et al., 2017; Davidson et al., 2017; Rodriguez et al., 2019), part-of-speech (POS) (Greevy & Smeaton, 2004), sentiment analysis (Rodriguez et al., 2019), topic modelling (e.g. Latent Dirichlet Allocation (LDA)) (Xiang et al., 2012), meta-information (e.g. location) (Waseem & Hovy, 2016), or word embeddings (Agrawal & Awekar, 2018; Badjatiya et al., 2017). Greevy and Smeaton (2004) found that the classification performance is higher with BOW features than with POS features. Waseem and Hovy (2016) found that character N-gram achieves higher classification performance than word N-gram. They also found that using demographic information such as the location does not improve the results significantly. Xiang et al. (2012) used a lexicon feature (whether a social media post contains an offensive word or not) and the topic distributions from an LDA analysis. Rodriguez et al. (2019) used TF-IDF and sentiment analysis to detect and cluster topics on Facebook pages that are likely to promote hate speech. Badjatiya et al. (2017) experimented with different word embeddings: fastText[1], GloVe[2], and random word embeddings. They found that using pre-trained word embeddings such as GloVe does not result in better classification performance than using random embeddings.

Most studies use supervised learning techniques that range from traditional ML to deep learning (DL) classification models, and a few use unsupervised learning techniques to cluster social media posts. Support Vector Machine (SVM) (Davidson et al., 2017; Greevy & Smeaton, 2004; Xiang et al., 2012) and Logistic Regression (LR) (Davidson et al., 2017; Waseem & Hovy, 2016) are the most popular traditional ML techniques for hate speech detection. Davidson et al. (2017) found that SVM and LR perform significantly better than other traditional ML techniques such as Naive Bayes, Decision Trees, and Random Forests. Badjatiya et al. (2017) experimented with various configurations of word embeddings and two DL models: a convolutional neural network (CNN) and a long short-term memory (LSTM) model. They

[1] https://fasttext.cc/
[2] https://nlp.stanford.edu/projects/glove/

found that CNN performs better than LSTM. Given the recent popularity of Bidirectional Encoder Representations from Transformers (BERT) models (Devlin et al., 2018) in the NLP field, studies such as Alatawi et al. (2021) found that BERT models achieve slightly better classification performance than DL models. Rodriguez et al. (2019) use the unsupervised learning method, K-means clustering, to cluster social media posts for identifying topics that potentially promote hate speech. Based on the findings of these studies, we will experiment with three models in our project: LR, CNN, and DistilBERT (a lightweight version of BERT (Sanh et al., 2019)).

2.3 Machine Learning with rejection

Several related studies promoted the concept of rejecting ML predictions when the risk of producing an incorrect prediction is too high so that a human gives the final judgement instead (Hendrickx et al., 2021; Sayin et al., 2021; Woo, 2020). Hendrickx et al. (2021) identified three ways of rejecting ML predictions: *separated*, *integrated*, and *dependent*. A separated rejector decides beforehand whether a data sample needs to be handled by the classification model or not (Hendrickx et al., 2021). An integrated rejector forms one whole with a classification model that we often train simultaneously (Hendrickx et al., 2021). A dependent rejector analyzes the output of the classification model to determine whether to reject a prediction or not (Hendrickx et al., 2021). Several studies have applied the reject option using one of the abovementioned architectures (Coenen et al., 2020; De Stefano et al., 2000; Geifman & El-Yaniv, 2017, 2019; Grandvalet et al., 2008).

Coenen et al. (2020) developed a *separated* rejector that rejects data samples before passing them to the classification model. They used different outlier detection techniques, such as the one-class Support Vector Machine (SVM), to detect data samples unfamiliar with the training data (Coenen et al., 2020).

Dependent rejectors are the most commonly used (De Stefano et al., 2000; Geifman & El-Yaniv, 2017; Grandvalet et al., 2008). Grandvalet et al. (2008) experimented with support vector machines (SVMs) with a reject option. Geifman and El-Yaniv (2017) developed a dependent rejector that rejects data samples based on a predefined maximum risk value and the coverage accuracy of the classification model (Geifman & El-Yaniv, 2017). De Stefano et al. (2000) were among the first to develop a dependent rejector for neural networks. The authors developed a confidence metric for determining the optimal rejection threshold (De Stefano et al., 2000). This threshold is calculated based on a set of predictions with their corresponding confidence values and a set of cost values: the cost of incorrect, correct, and rejected predictions (De Stefano et al., 2000).

Geifman and El-Yaniv (2019) developed an *integrated* rejector by extending their work from Geifman and El-Yaniv (2017). They integrated the reject option in the training phase of a DL classification model by including a selection function in the last layer of the DL model.

In this work, we apply the dependent way since it allows for applying the reject option to any existing classification model (Hendrickx et al., 2021). The most relevant work is from De Stefano et al. (2000) since their confidence metric considers the value of (in)correct and rejected predictions. While their metric measures only the effectiveness of the reject option and is based on the values of correct, incorrect, and rejected predictions, our metric measures the total value of the ML model with the reject option and is based on the values of TP, TN, FP, FN, and rejected predictions.

While they experimented with a range of different cost values, we go further by employing a value-sensitive approach, which determines the cost values based on how users feel regarding machine predictions using a survey study with crowd workers. Therefore, we obtain a rejection threshold that captures the implications of machine predictions from a human perspective.

2.4 Value assessment

Fjeld et al. (2020) outlined eight principles of AI systems, such as *fairness and discrimination* (e.g. preventing algorithmic bias), *human control of technology* (e.g. the system should request help from the human user in difficult situations), and *promotion of human values* (e.g. we should integrate human value in the system). Sayin et al. (2021) and Casati et al. (2021) suggest we should identify context-specific *values* and incorporate them in the design of a hybrid human-AI system. We adhere to the suggestions of these studies in our project since we develop a hate speech classification model with a reject option that incorporates human value.

As explained in the Introduction, we have costs of incorrect and rejected predictions and gains of correct predictions. We can express the costs of incorrect (FP and FN) and rejected predictions as negative values and the gains of correct (TP and TN) predictions as positive values. We should weigh these values according to the task of case hate speech detection (Sayin et al., 2021). However, value is a broad term, and its definition depends heavily on the context.

Several works discuss the value-sensitive design (VSD) approach that describes how different types of value, such as privacy, can be integrated into a socio-technical system's design (Umbrello & Van de Poel, 2021; Zhu et al., 2018). According to the VSD approach, it is critical to understand the system's stakeholders, and we can retrieve their values either *conceptually* (e.g. from literature) or *empirically* (e.g. through survey studies) (Umbrello & Van de Poel, 2021; Zhu et al., 2018).

We consider two different stakeholders: the social media platforms and the users. The goal is to find out whether we can retrieve the value ratios between rejection, FP, FN, TP, and TN predictions from the perspective of both stakeholders. We would like to know whether an FN prediction is, for example, two times worse than an FP prediction. The main challenge is to express all values using a single unit. First, we could define the values using an objective measure, such as time or money spent/saved. Second, we could define the values subjectively, for example, by analyzing people's stance towards the consequence of incorrect predictions in hate speech detection.

In this section, we try to assess the values of both stakeholders empirically and conceptually and explain why we eventually go for an empirical analysis of the values of the social media users only.

2.4.1 Objective assessment

In this section, we explain the difficulties of defining the values of TP, TN, FP, FN, and rejected predictions in hate speech detection using objective measurements. We do this by following the conceptual approach for both stakeholders by looking at some related work to see if the empirical approach is possible.

First, we look at the social media company as a stakeholder. We can retrieve the value of rejection by looking at how much time a human moderator spends on average to check whether some social media post contains hateful content or not. We

can convert this into money by considering the moderator's salary. We could also argue that the value of a TP and a TN prediction is equal to the negative value of rejection since we saved human effort by having the classification model produce a correct prediction. The problem, however, starts to arise when we look at the FP and the FN predictions. How can we express the values of FP and FN predictions regarding money or time saved/spent? The main problem is that most social media companies are not transparent about moderate hate speech (Klonick, 2018). So it is infeasible to assess the values of the social media companies either conceptually or empirically. When looking at the consequences of FN predictions, we can also look at governmental fines. For example, Germany approved a plan where social media companies can be fined up to 50 million euros if they do not remove hate speech in time ("Social media firms faces huge hate speech fines in Germany", 2017). However, this is location-specific, and it is unclear how this applies to individual cases of hate speech. Defining the value of FP predictions is even more difficult. It is unclear how filtering out too much content would affect the company regarding money/time lost. Therefore, we abstain from estimating the values from the companies' perspective as the stakeholder.

Second, we look at the social media users as the stakeholder. Both FP and FN predictions have negative consequences on the users. Having too many FP predictions might violate the value of Freedom of Speech since we are filtering out non-hateful posts and, therefore, we cause suppression of free speech. One paper found through a survey that most people think some form of hate speech moderation is needed, but they also worry about the violation of freedom of speech (Olteanu et al., 2017). Having too many FN predictions might harm individuals or even result in acts of violence (Council of Europe, n.d.). Therefore, we must figure out how to weigh the values of FP and FN predictions accordingly. We abstain from using time as a unit since it does not make sense to express the consequences of hate speech or the benefits of freedom of speech in time. Therefore, we want to look at the value of freedom of speech and hate speech from an economic perspective. However, we noticed a lack of research in this area. There is one paper where they tried to develop an economic model for free political speech by looking at the First Amendment to the United States Constitution (Posner, 1986). The First Amendment restricts the government from creating laws that could, for example, violate Freedom of Speech ("The Constitution", n.d.). The authors explained in Posner (1986) that the lack of research in this area is because most economists do not dive into the legal domain regarding free speech, and free speech legal specialists refrain from doing economic analysis (Posner, 1986). The proposed economic model from the paper includes the cost of harm and the probability that speech results in violence (Posner, 1986). However, the authors do not elaborate on how we can define the probability and the costs. Another paper did speculate on this topic by explaining why doing a cost-benefit analysis of free speech is almost impossible (Sunstein, 2019). The authors explained that there are too many uncertainties (Sunstein, 2019). We can assume that there are values of free speech, but it is too difficult to quantify them (Sunstein, 2019). Terrorist organizations use free speech to recruit people and call for acts of violence online (Sunstein, 2019). At the same time, most other hateful posts will never result in actual acts of violence (Sunstein, 2019). Therefore, value assessment using objective measurements is already tricky for specific cases, let alone in general. There is a nonquantifiable risk that acts of violence will happen in the unknown future (Sunstein, 2019). However, suppose we know this probability, there are still too many uncertainties. To calculate the actual costs of hate speech (the FN predictions), we also need to know the number of lives at risk and how we should quantify the

value of each life (Sunstein, 2019). The authors claim that analyzing the benefits of free speech is even more difficult (Sunstein, 2019). They conclude their work by saying that there are too many problems to empirically evaluate the costs and benefits of hate speech detection (Sunstein, 2019).

Therefore, we believe that using objective measurements, such as money, is impossible to assess the values of predictions for both stakeholders in hate speech detection.

2.4.2 Subjective assessment

From section 2.4.1, we concluded that from related work, it appears that we cannot retrieve the objective values conceptually and empirically. Instead, we will focus on the subjective measurement of values: what is people's stance towards (in)correct and rejected predictions in hate speech detection? We only consider the social media users as the stakeholder in the subjective assessment since they are the most affected by the consequences of hate speech detection. We will empirically assess social media users' value through a survey. In our survey, we ask social media users what their stance (disagree-agree) is towards TP, TN, FP, FN, and rejected predictions in hate speech detection. Conceptual analysis is impossible since no related studies have tackled this problem. The closest work is from Ross et al. (2017), where the authors asked human subjects to rate a selection of tweets on hatefulness using a 6-point Likert scale and to indicate whether they think it should be banned from Twitter or not. Like Ross et al. (2017), we could use the commonly used Likert scales as our measurement scale. However, as we will explain in section 2.4.2, Likert scales are unsuitable for retrieving ratio values. Therefore, in section 2.4.2, we will explain why the Magnitude Estimation technique seems promising for our use case.

Likert

Likert scales are a common choice in academic research for retrieving the opinions of a group of subjects. Likert scales are multiple Likert-type questions (items) where subjects can answer questions with several response alternatives (Boone & Boone, 2012). For example, we could use a bipolar scale with seven response alternatives ranging from 'strongly disagree' to 'strongly agree', including a 'neutral' midpoint. However, there is a lot of discussion in the literature about how we should analyze these Likert scales (Allen & Seaman, 2007; Boone & Boone, 2012; Murray, 2013; Norman, 2010). The scale of the questions is ordinal, which means that we know the responses' ranking, but we do not have an exact measurement of the distances between the response items (Allen & Seaman, 2007). For example, we know that 'strongly agree' is higher in rank than 'agree', but not the exact distance between the two responses and whether it is greater than the distance between the 'neutral' and the 'somewhat agree' responses. Therefore, we technically cannot use parametric statistics, such as calculating the mean, when analyzing the data (Allen & Seaman, 2007). Other papers argue that we can treat a Likert scale that consists of multiple Likert items as interval data and, therefore, applying parametric statistics will not affect the conclusions (Boone & Boone, 2012; Murray, 2013; Norman, 2010). So, we can calculate mean scores for TP, TN, FP, FN, and rejected predictions and compare these with each other. For example, we can then verify that the mean value of FN predictions is smaller than the mean value of FP predictions and conclude that FN predictions are worse than FP predictions. Analyzing Likert scales would at most

provide us with interval data (data for which we know the order, and we can measure the distances, but there is no true zero point (Allen & Seaman, 2007)). However, we need to have ratio data in this project since we want to know the exact value ratios between the TP, TN, FP, FN, and rejected predictions.

Magnitude Estimation

In section 2.4.2, we concluded that Likert scales are unsuitable since they do not provide ratio data. In this research, we want to experiment with the Magnitude Estimation (ME) technique. The ME technique originates from psychophysicists, where human subjects must give quantitative estimations of sensory magnitudes (Stevens, 1956). For example, in one experiment, human subjects are asked to assign any number that reflects their perception of the loudness of a range of sounds (Stevens, 1956). If the human subjects perceive the succeeding sound as twice as loud, they should assign a number to it that is twice as large. Researchers applied the ME technique to different types of physical stimuli (e.g. line length, brightness, or duration) and proved that the results are reproducible and that the data has ratio properties (Moskowitz, 1977). Other works have shown that the ME technique is also helpful for rating more abstract types of stimuli, such as judging the relevance of documents (Maddalena et al., 2017; Roitero et al., 2018), the linguistic acceptability of sentences (Bard et al., 1996), the strength of political opinions (Lodge et al., 1976; Lodge & Tursky, 1979), and the usability of system interfaces (McGee, 2004). Therefore, we think that ME is a promising method for judging the value ratios of the different types of predictions in hate speech detection.

The main advantage of ME is that it provides the ratio scale properties we need. Another advantage is that the scale is unbounded compared to other commonly used response scales, such as Likert. For example, suppose the subject provides a 'strongly disagree' judgment for the first stimulus. Suppose we now present an even worse stimulus. The subject is now limited to the response items in the Likert scale and can only give the same 'strongly disagree' judgement. We do not have this problem using ME because the subject is always free to assign a more significant value of disagreement. However, there are two drawbacks to using ME in our use case. First, we need to normalize the results since each subject uses a different range of values. Second, since ME has not been applied to the hate speech domain before, we need to validate the ME scale to verify that it measures what we want to know.

The data needs to be normalized since each subject can use any value they like. For example, one may give ratings using values of 1, 2, and 10, while another may use 100, 200, and 1000. Geometric averaging is the recommended approach for normalizing magnitude estimates since it preserves the ratio information (Maddalena et al., 2017; McGee, 2004; Moskowitz, 1977). However, as opposed to the unipolar scales (with only positive values) used in Bard et al. (1996), Maddalena et al. (2017), and McGee (2004), we cannot apply geometric averaging to bipolar scales (disagree-agree). By including 0 (neutral) and negative values (disagree), we cannot use geometric averaging anymore because it uses log calculations (Moskowitz, 1977). Using the algorithmic mean is also not an option since it would destroy the ratio scale properties (Moskowitz, 1977). Therefore, we can normalize the magnitude estimates for bipolar scales by dividing all estimates of each subject by the maximum given value (Moskowitz, 1977). This way, all magnitudes estimates are in the range [-100, 100] while maintaining the ratio properties.

Most papers that use the ME method in a new domain apply some form of validation. Cross-modality validation is a technique that is often applied to validate the

ME results (Bard et al., 1996). Psychophysicists compare the magnitude estimates to the physical stimuli by analyzing their correlation (Bard et al., 1996). In the case of estimating line lengths, we can easily vary the line length, for example, by showing a line that is twice as long as the previous line. Subjects can then estimate the line length using a number twice as large. However, this becomes more difficult in the social science and psychology domains. In hate speech detection and other social science and psychology applications, we do not have an exact measure of the stimulus (Bard et al., 1996). However, related work has shown that ME is still a suitable technique for eliciting opinions about different types of non-physical stimuli (Bard et al., 1996; Lodge & Tursky, 1979; Maddalena et al., 2017; McGee, 2004). We can validate the magnitude estimates by adopting the cross-modality technique but instead compare judgements against judgements (Bard et al., 1996; Lodge & Tursky, 1979). Some papers analyze the correlation between different ME scales for validation, such as handgrip measurements or drawing lines (Bard et al., 1996; Lodge et al., 1976). Others compare ME with another validated scale that can be of any type. For example, in Maddalena et al., 2017 which is about judging the relevance of documents, the authors compared the ME scale with two validated ordinal scales for the same dataset (Maddalena et al., 2017). In Roitero et al. (2018), the authors applied cross-modality analysis between a bounded scale that consists of 100 levels (now known as the 100-level scale) and the ME scale and found that they were positively correlated. In our work, we follow the approach from Roitero et al. (2018) as we also validate our findings by checking the correlation between the ME scale and the 100-level scale.

2.5 Evaluation metrics

Most hate speech-related studies evaluate their classification methods using standard *machine* metrics such as accuracy, precision, recall, or F1. Classification models with a reject option are often evaluated by analyzing the model's accuracy and coverage. Nadeem et al. (2009) proposed using accuracy-rejection curves to plot the trade-off between accuracy and coverage so that different classification models with a reject option can be compared. Casati et al. (2021), Olteanu et al. (2017), and Röttger et al. (2020) recognized the shortcomings of machine metrics such as accuracy and found a gap in the evaluation of hate speech detection systems. Röttger et al. (2020) found it hard to identify the weak points of classification models using machine metrics such as accuracy. Therefore, the authors presented a suite that consists of 29 carefully selected functional tests to help identify the model's weaknesses (Röttger et al., 2020). Each test checks criteria, such as coping with spelling variations or detecting neutral content containing slurs (Röttger et al., 2020). Our approach is different since we focus on measuring the value of classification models with a reject option. Olteanu et al. (2017) promote using *human-centred* metrics that measure the human-perceived value of hate speech classification models. They found that for the same precision values, the perceived value changes depending on the user characteristics and the type of classification errors (an offensive tweet labelled as hate (low impact) and a neutral tweet labelled as hate (high impact)) (Olteanu et al., 2017). Casati et al. (2021) propose to develop new metrics for evaluating ML models with a reject option that considers domain-specific values. Our work aligns with both studies since we create a human-centred metric for evaluating hate speech classification models with a reject option that incorporates human value derived from a survey study.

Chapter 3

Value-sensitive rejection

As concluded in the Related work, there is a need for *value-sensitive* metrics for measuring the performance of ML models, especially for social-technical applications such as hate speech detection. We also concluded that manual human moderation is the most effective and that most automatic hate speech detection methods do not perform well on unseen data. Therefore, in this project, we focus on rejecting ML predictions in a value-sensitive manner. We do this by taking the values of TP, TN, FP, FN, and rejected predictions into account. Section 4 will explain how we assess these values. For the remaining part of this chapter, we assume that we know these values.

In this chapter, we explain how we create a dependent rejector by introducing a confidence metric that measures the total value of an ML model with a reject option. In 3.1, we explain how we construct the confidence metric, and in 3.2, we discuss how we apply the metric to some of the state-of-the-art hate speech classification models.

3.1 Value-sensitive metric

The idea of a rejecting ML predictions using a treshold is that for some treshold value τ in the range $[0, 1]$, we accept all predictions with confidence values that are greater than or equal to τ and reject all predictions with confidence values below τ. We use a confidence metric to find the optimal rejection threshold. Here, we introduce our confidence metric as the value function $V(\tau)$ that measures the total value of some ML model with rejection threshold τ. We can determine the optimal rejection threshold by finding the τ value for which $V(\tau)$ is maximum. The value of $V(\tau)$ depends on the values of TP, TN, FP, FN, and rejected predictions and is calculated for a set of predictions with their corresponding confidence values and actual labels. We denote the values of TP, TN, FP, FN, and rejected predictions as V_{tp}, V_{tn}, V_{fp}, V_{fn}, and V_r respectively. From the set of predictions we can derive the subsets of TP, TN, FP, and FN predictions. Note that we consider that V_{tp} and V_{tn} can be expressed as gains and, therefore, non-negative values and that V_{fp}, V_{fn}, and V_r can be expressed costs and, therefore, non-positive values. There is one condition that needs to hold: the reduction of total value by means of a rejection should always be smaller in magnitude than the value reduction of an incorrect prediction. Otherwise adopting the reject option serves no purpose. This condition can be formulated as:

$$\frac{V_{FP} + V_{FN}}{2} < V_R, \tag{3.1}$$

For each τ value in $[0, 1]$, we would like to know whether the model with the reject option is more effective (increased $V(\tau)$) or less effective (decreased $V(\tau)$ value). In general, the metric should take the following conditions into account:

1. Correct accepted predictions should increase the value of $V(\tau)$, while incorrect accepted predictions should decrease the value of $V(\tau)$.
2. Correct rejected predictions should decrease the value of $V(\tau)$, while incorrect rejected predictions should increase the value of $V(\tau)$.

We can convert these conditions into the following equations:

$$\frac{\partial V}{\partial R_{tp}(\tau)} + \frac{\partial V}{\partial R_{tn}(\tau)} >= 0, \qquad \frac{\partial V}{\partial R^r_{tp}(\tau)} + \frac{\partial V}{\partial R^r_{tn}(\tau)} <= 0, \tag{3.2a}$$

$$\frac{\partial V}{\partial R_{fp}(\tau)} + \frac{\partial V}{\partial R_{fn}(\tau)} <= 0, \qquad \frac{\partial V}{\partial R^r_{fp}(\tau)} + \frac{\partial V}{\partial R^r_{fn}(\tau)} >= 0, \tag{3.2b}$$

where $R_t(\tau)$ and $R^r_t(\tau)$ are the fractions of accepted and rejected predictions respectively based on the rejection threshold τ and where $t \in [tp, tn, fp, fn]$. All conditions include greater/smaller than or equal to operators, since we want to allow the values of V_{tp}, V_{tn}, V_{fp}, V_{fn}, and V_r to be equal to zero. We create a linear $V(\tau)$ function and assume that the values V_t are known constants. Subsequently, we can formulate $V(\tau)$ as:

$$V(\tau) = \sum_p (V_p - V_r) R_p(\tau) + \sum_q (V_r - V_q) R^r_q(\tau), \tag{3.3}$$

where $p, q \in [tp, tn, fp, fn]$. Conditions 3.2a are satisfied by default since we assume that V_{tp} and V_{tn} are non-negative and V_r is non-positive. Conditions 3.2b are satisfied under the condition of 3.1 and since we assume that V_{fp}, V_{fn}, and V_r are negative. We can define the R_t and the R^r_t values by computing the integrals over the probability density functions (PDF) of the confidence values of the predictions with type t. We can define R_t by taking the integral over the interval $[\tau, 1]$, and R^r_t by taking the integral over the interval $[0, \tau]$:

$$R_t(\tau) = \int_\tau^1 D_t(x)dx \qquad R^r_t(\tau) = \int_0^\tau D_t(x)dx, \tag{3.4}$$

where D_t is the PDF of all predictions of type t. By inserting the integrals from 3.4 into 3.3, we get our final value function:

$$V(\tau) = \sum_t (V_t - V_r) \int_\tau^1 D_t(x)dx + \sum_t (V_r - V_t) \int_0^\tau D_t(x)dx \tag{3.5}$$

We can now use 3.5 to calculate the total value of some ML model for all $\tau \in [0, 1]$ values. The theoretical optimal rejection threshold is equal to the τ value for which we achieve the maximum value of $V(\tau)$. We can find the optimal rejection threshold τ_O using the following formulation:

$$\tau_O \text{ where } V(\tau_O) = \max\{V(\tau) : \tau \in \mathbb{R} \wedge 0 \leq \tau \leq 1\} \tag{3.6}$$

3.2 State-of-the-art

In this section, we will explain how we will apply the value-sensitive metric from section 3.1 to some of the state-of-the-art automatic hate speech detection models to convert them into dependent rejectors. In this experiment, we aim to find out three things. First, we want to find out how the value-sensitive metric behaves on different models and datasets. Second, we want to know wheter rejecting ML predictions can be beneficial in the hate speech detection domain. Finally, we will compare our value-sensitive metric to machine metrics such as accuracy and check whether they give different results.

3.2.1 Models

We will experiment with three different hate speech detection models based on the findings from related work in section 2.2. The first model is a traditional ML model. We implement the Logistich Regression (LR) model with Character N-gram from Waseem and Hovy (2016) since this model achieved the best performance compared to other traditional ML models (Davidson et al., 2017). We selected the second model, an DL model, based on the findings from Agrawal and Awekar (2018) and Badjatiya et al. (2017). We chose a Convolutional Neural Network (CNN) model initialized with random word embeddings since both studies found that this configuration provides state-of-the-art classification performance. We implemented the CNN model based on the work of (Agrawal & Awekar, 2018). Finally, our third model is a transformer model given its recent popularity in the NLP domain. We selected the DistilBERT model since it's faster and smaller in size compared to BERT models while achieving similar performance (Sanh et al., 2019). We implement all models in Python. We implement the LR model with scikit-learn[1], the CNN model with TensorFlow[2], and the DistilBERT model with a combination of Hugging Face[3] and PyTorch[4]. We use Google Colab[5] to train all models.

3.2.2 Calibration

The problem of most neural network models is that they are often not calibrated (Guo et al., 2017; Sayin et al., 2021). We can define calibrated models as models where the confidence values of the predictions are equal to the probabilities that the predicted labels are correct. However, most neural networks tend to be sensitive to producing both low- and high-confident errors (Guo et al., 2017; Sayin et al., 2021). A well-calibrated model that achieves a low accuracy score can still be valuable since we can reject all low-confident incorrect predictions and only accept the high-confident correct predictions (Sayin et al., 2021). In our project, we aim to have calibrated models since calculating the optimal rejection threshold depends on the confidence values of the predictions.

Guo et al. (2017) experimented with different calibration methods. They evaluated the results using the expected calibration error (ECE), which measures the difference between the expected confidence and accuracy (Guo et al., 2017). They found that the temperature scaling method is the most effective. In temperature

[1] https://scikit-learn.org/
[2] https://www.tensorflow.org/
[3] https://huggingface.co/
[4] https://pytorch.org/
[5] https://colab.research.google.com/

scaling, we divide the model's output logits with a temperature value T to soften the probabilities of the final softmax function in the model's architecture (Guo et al., 2017). This T value is initially set to 1 and optimized by minimizing the negative log-likelihood (Guo et al., 2017). Please note that temperature scaling does not change the model's accuracy but only rescales the distribution of the confidence values (Guo et al., 2017).

As we will experiment with two neural networks (DistilBERT and CNN), we will apply temperature scaling to calibrate both models. Unfortunately, it is also possible that once we calibrated these models with temperature scaling, they can still produce high-confident incorrect predictions and low-confident correct predictions. However, it is still valuable to calibrate the models since it also benefits human interpretation of the confidence values and, therefore, the optimal rejection threshold.

The Logistic Regression model is well-calibrated by default since it optimizes the log-loss function which measures the difference between predicted confidence values and the actual labels. Therefore, we do not have to apply temperature scaling to the Logistic Regression model.

3.2.3 Datasets

We train all models on the Waseem and Hovy (2016) dataset consisting of 16K tweets labelled racist, sexist, or neutral. We converted the 'racist' and 'sexist' labels to 'hate' labels to create a binary classification setting. We split the dataset into a train and test dataset according to an 80:20 ratio, respectively (80% for training and 20% for testing) For the CNN and the DistilBERT models, we split the training set up into a training set and a validation set according to a 75:25 ratio, respectively. We use this validation set to calibrate the trained models by finding the optimal temperature T value for the temperature scaling method. We preprocess the data by tokenizing all URLs, user mentions, and emojis since these do not contain any valuable information. The remaining parts of the preprocessing, such as removing whitespaces and stop words or the tokenization process, are dedicated to the different frameworks we use per model.

We apply the metric to two test datasets: the *seen* dataset and the *unseen* dataset. The seen dataset is the test set from the Waseem and Hovy (2016) dataset. The unseen dataset is a test set from the Basile et al. (2019) dataset that consists of 10K English tweets labelled as either hateful (against immigrants or women) or not hateful. We use the unseen dataset to simulate how the models would perform in a realist use-case when a model is trained on one dataset and applied to a different dataset. We want to analyze the effect of bias and how this affects the results when using our metric for evaluating the models when rejection is adopted. We expect that the accuracy of the predictions on the unseen dataset is significantly lower than on the seen dataset, similar to the findings of related studies: Arango et al. (2019) and Gröndahl et al. (2018). Therefore, we also expect that the output value of our metric for the unseen dataset will be lower and that the optimal rejection threshold will be lower (meaning that we need to reject more predictions).

3.2.4 Probability Density Functions

Since our metric depends on the PDFs of the confidence values of the TP, TN, FP, and FN predictions, we need to emperically estimate these PDFs as we do not know the actual underlying distributions. We use the Kernel Density Estimation (KDE)

method provided by Statsmodels[6] for estimating the PDFs. With KDE we estimate the PDF by weighing each confidence value from a set of predictions using a kernel function, a gaussian density function in our case since it's the most commonly used, for each value in the range $[0, 1]$. If there are many predictions with a confidence value around 0.8, then the KDE estimate will be higher around that point. The kernel fucntion used in the KDE method also depends on a bandwidth (smoothness) value. A small bandwidth value results in an estimated PDF with a lot of variance, while a high bandwidth value results in an estimated PDF with a lot of bias. We use maximum likelihood cross validation for finding the optimal bandwidth value.

3.2.5 Application of the metric

We apply our metric from section 3.1 to calculate the total value ($V(\tau)$ from formula 3.5) of all three models for both the seen and the unseen datasets at all possible rejection thresholds (τ). We determine the optimal rejection threshold τ_O using the formulation from 3.6. Since we have a binary classification setting (hate or not hate) all confidence values will always be greater than 0.5. So if $\tau \in [0.0, 0.5]$, then we accept all predictions and if $\tau = 1.0$, then we reject all predictions. Therefore, we only calculate the total value of all predictions for the range $\tau \in [0.5, 1.0]$.

The first goal is to check the behaviour of the model on different models and datasets. We can analyze this by plotting $V(\tau)$ for the range $\tau \in [0.5, 1.0]$. The second goal is to find out whether the reject option can beneficial for hate speech detection. If the total value of a model for some optimal rejection threshold ($0.5 < \tau_O < 1.0$) is positive, then we know that the reject option can be beneficial for that specific model. The final goal is to compare the metric to machine metrics such as accuracy. We will do this by comparing the $V(\tau_O)$ values and the accuracies between all models.

[6] https://www.statsmodels.org/

Chapter 4

Survey study

4.1 Hypothesis

4.2 Method

4.2.1 Scales

4.2.2 Normalization

4.2.3 Design

Independent variables

Confounding variables

Control variables

Dependent variables

4.2.4 Planned sample

Sample size

Subjects

Explain subject Inclusion and Exclusion Criteria

Explain subject Compensation

4.2.5 Materials

Survey tool

Data

4.2.6 Procedure

4.3 Analysis

4.3.1 Validation

4.3.2 Reliability

Chapter 5

Results

5.1 Survey study

5.1.1 Value ratios

5.1.2 Reliability

5.1.3 Validity

5.2 Value-sensitive rejection

Chapter 6

Discussion

Answer research questions

Discuss future work

6.1 Survey study

6.2 Value-sensitive rejection

6.3 Implications

Explain that Olteanu et al., 2017 claims that we need more human-centred metrics instead of abstract metrics such as precision and we agree with that by introducing our own human-centred metric

6.4 Limitations

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. Ross et al. (2017) found low Krippendorff alpha values in a hate speech survey. So our findings are in line with theirs.

Explain limitations of the metric and the survey study

The rejection threshold is calculated using the test set. This test set needs to be as realistic as possible. Furthermore we need to have calibrated models since we rely purely on the confidence values. This is also hard to realize. Temperature scaling can help, but it is still limited.

6.5 Recommendations

Magnitude Estimation seems promising for future research in HCI.

Personal and demographic characterisitcs might have a big impact. So further analysis on those aspects seem relevant.

Chapter 7

Conclusion

Bibliography

Agrawal, S., & Awekar, A. (2018). Deep learning for detecting cyberbullying across multiple social media platforms. *European conference on information retrieval*, 141–153.

Alatawi, H. S., Alhothali, A. M., & Moria, K. M. (2021). Detecting white supremacist hate speech using domain specific word embedding with deep learning and bert. *IEEE Access*, *9*, 106363–106374.

Allen, I. E., & Seaman, C. A. (2007). Likert scales and data analyses. *Quality progress*, *40*(7), 64–65.

Arango, A., Pérez, J., & Poblete, B. (2019). Hate speech detection is not as easy as you may think: A closer look at model validation. *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, 45–54.

Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep learning for hate speech detection in tweets. *Proceedings of the 26th international conference on World Wide Web companion*, 759–760.

Balayn, A., Yang, J., Szlavik, Z., & Bozzon, A. (2021). 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.

Bard, E. G., Robertson, D., & Sorace, A. (1996). Magnitude estimation of linguistic acceptability. *Language*, *72*(1), 32–68.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F. M. R., Rosso, P., & Sanguinetti, M. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. *Proceedings of the 13th international workshop on semantic evaluation*, 54–63.

Boone, H. N., & Boone, D. A. (2012). Analyzing likert data. *Journal of extension*, *50*(2), 1–5.

Casati, F., Noël, P.-A., & Yang, J. (2021). On the value of ml models. *arXiv preprint arXiv:2112.06775*.

Coenen, L., Abdullah, A. K., & Guns, T. (2020). Probability of default estimation, with a reject option. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 439–448.

The constitution [Visited on 11/04/2022]. (n.d.). *The White House*. https://www.whitehouse.gov/about-the-white-house/our-government/the-constitution/

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

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, *11*(1), 512–515.

De Stefano, C., Sansone, C., & Vento, M. (2000). To reject or not to reject: That is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *30*(1), 84–94.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Fjeld, J., Achten, N., Hilligoss, H., Nagy, A., & Srikumar, M. (2020). Principled artificial intelligence: Mapping consensus in ethical and rights-based approaches to principles for ai. *Berkman Klein Center Research Publication*, (2020-1).

Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, *51*(4), 1–30.

Geifman, Y., & El-Yaniv, R. (2017). Selective classification for deep neural networks. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4885–4894.

Geifman, Y., & El-Yaniv, R. (2019). SelectiveNet: A deep neural network with an integrated reject option. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 2151–2159). PMLR. https://proceedings.mlr.press/v97/geifman19a.html

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

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., & Canu, S. (2008). Support vector machines with a reject option. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2008/file/3df1d4b96d8976ff5986393e8767f5Paper.pdf

Greevy, E., & Smeaton, A. F. (2004). Classifying racist texts using a support vector machine. *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 468–469.

Gröndahl, T., Pajola, L., Juuti, M., Conti, M., & Asokan, N. (2018). All you need is "love" evading hate speech detection. *Proceedings of the 11th ACM workshop on artificial intelligence and security*, 2–12.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. *Proceedings of the 34th International conference on machine learning*, 1321–1330.

Hendrickx, K., Perini, L., Van der Plas, D., Meert, W., & Davis, J. (2021). Machine learning with a reject option: A survey. *arXiv preprint arXiv:2107.11277*.

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

Klonick, K. (2018). The new governors: The people, rules, and processes governing online speech. *Harvard Law Review*, *131*, 1598.

Lodge, M., Tanenhaus, J., Cross, D., Tursky, B., Foley, M. A., & Foley, H. (1976). The calibration and cross-modal validation of ratio scales of political opinion in survey research. *Social Science Research*, *5*(4), 325–347.

Lodge, M., & Tursky, B. (1979). Comparisons between category and magnitude scaling of political opinion employing src/cps items. *American Political Science Review*, *73*(1), 50–66.

Maddalena, E., Mizzaro, S., Scholer, F., & Turpin, A. (2017). On crowdsourcing relevance magnitudes for information retrieval evaluation. *ACM Transactions on Information Systems (TOIS)*, *35*(3), 1–32.

Mashal, M., Raj, S., & Kumar, H. (2022). As officials look away, hate speech in india nears dangerous levels [Visited on 07/03/2022]. *The New York Times*. https://www.nytimes.com/2022/02/08/world/asia/india-hate-speech-muslims.html

McGee, M. (2004). Master usability scaling: Magnitude estimation and master scaling applied to usability measurement. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 335–342.

Moskowitz, H. R. (1977). Magnitude estimation: Notes on what, how, when, and why to use it. *Journal of Food Quality*, *1*(3), 195–227.

Mozur, P. (2018). A genocide incited on facebook, with posts from myanmars military [Visited on 07/03/2022]. *The New York Times*. https://www.nytimes.com/2018/10/15/technology/myanmar-facebook-genocide.html

Müller, K., & Schwarz, C. (2021). Fanning the flames of hate: Social media and hate crime. *Journal of the European Economic Association*, *19*(4), 2131–2167.

Murray, J. (2013). Likert data: What to use, parametric or non-parametric? *International Journal of Business and Social Science*, *4*(11).

Nadeem, M. S. A., Zucker, J.-D., & Hanczar, B. (2009). Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In S. Deroski, P. Guerts, & J. Rousu (Eds.), *Proceedings of the third international workshop on machine learning in systems biology* (pp. 65–81). PMLR. https://proceedings.mlr.press/v8/nadeem10a.html

Norman, G. (2010). Likert scales, levels of measurement and the laws of statistics. *Advances in health sciences education*, *15*(5), 625–632.

Olteanu, A., Talamadupula, K., & Varshney, K. R. (2017). The limits of abstract evaluation metrics: The case of hate speech detection. *Proceedings of the 2017 ACM on Web Science Conference*, 405–406.

Posner, R. A. (1986). Free speech in an economic perspective. *Suffolk University Law Review*, *20*, 1.

Rodriguez, A., Argueta, C., & Chen, Y.-L. (2019). Automatic detection of hate speech on facebook using sentiment and emotion analysis. *2019 international conference on artificial intelligence in information and communication (ICAIIC)*, 169–174.

Roitero, K., Maddalena, E., Demartini, G., & Mizzaro, S. (2018). On fine-grained relevance scales. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 675–684.

Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., & Wojatzki, M. (2017). Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

Röttger, P., Vidgen, B., Nguyen, D., Waseem, Z., Margetts, H., & Pierrehumbert, J. B. (2020). Hatecheck: Functional tests for hate speech detection models. *arXiv preprint arXiv:2012.15606*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sayin, B., Yang, J., Passerini, A., & Casati, F. (2021). The science of rejection: A research area for human computation. *arXiv preprint arXiv:2111.06736*.

Schmidt, A., & Wiegand, M. (2019). A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain*, 1–10.

Social media firms faces huge hate speech fines in germany [Visited on 11/04/2022]. (2017). *BBC News*. https://www.bbc.com/news/technology-39506114

Stevens, S. S. (1956). The direct estimation of sensory magnitudes: Loudness. *The American journal of psychology*, *69*(1), 1–25.

Sunstein, C. R. (2019). Does the clear and present danger test survive cost-benefit analysis? *Cornell Law Review*, *104*, 1775.

Umbrello, S., & Van de Poel, I. (2021). Mapping value sensitive design onto ai for social good principles. *AI and Ethics*, *1*(3), 283–296.

Waseem, Z. (2016). Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. *Proceedings of the first workshop on NLP and computational social science*, 138–142.

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. *Proceedings of the NAACL student research workshop*, 88–93.

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

Xiang, G., Fan, B., Wang, L., Hong, J., & Rose, C. (2012). Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. *Proceedings of the 21st ACM international conference on Information and knowledge management*, 1980–1984.

Zhu, H., Yu, B., Halfaker, A., & Terveen, L. (2018). Value-sensitive algorithm design: Method, case study, and lessons. *Proceedings of the ACM on human-computer interaction*, *2*(CSCW), 1–23.