Neural Network Hate Deletion: Developing a Machine Learning Model to Eliminate Hate from Online Comments

Joni Salminen[1,2], Juhani Luotolahti[2], Hind Almerekhi[1], Bernard J. Jansen[1], Soon-gyo Jung[1]

[1] Hamad Bin Khalifa University, Doha, Qatar; and [2] University of Turku, Turku, Finland
jsalminen@hbku.edu.qa, mjluot@utu.fi, hialmerekhi@hbku.edu.qa, bjansen@hbku.edu.qa, sjung@hbku.edu.qa

Abstract. We propose a method for modifying hateful online comments to non-hateful comments without losing the understandability and original meaning of the comments. To accomplish this, we retrieve and classify 301,153 hateful and 1,041,490 non-hateful comments from Facebook and YouTube channels of a large international media organization that is a target of considerable online hate. We supplement this dataset by 10,000 Reddit comments manually labeled for hatefulness. Using these two datasets, we train a neural network to distinguish linguistic patterns. The model we develop, Neural Network Hate Deletion, computes how hateful the sentences of a social media comment are and if they are above a given threshold, it deletes them using a language dependency tree. We evaluate the results by comparing crowd workers' perceptions of hatefulness and understandability before and after transformation and find that our method reduces hatefulness without resulting in a significant loss of understandability. In some cases, removing hateful elements from online comments has the potential of improving understandability by reducing the linguistic complexity of the comment. In addition, we find that the network can satisfactorily retain the original meaning on average but is not perfect in this regard. The research has practical and theoretical implications. For example, automation could be used in real systems to suggest more neutral use of language to agitated online users.

Keywords: Online hate; Toxic comments; Hate deletion; Neural networks

1 Introduction

Hateful remarks and comments are prevalent in online communities, including YouTube, Reddit, Facebook, Instagram and various other social networks and online discussion forums [1]. But why is hate so prevalent? One reason is due to its contagious nature, spreading from one individual to another [2, 3]. Following this logic of previous research, the more we can reduce the initiation of hate, the more we can reduce the overall hate taking place online. Reducing hate, however, is to a large degree sociological problem, dealing with manners, upbringing, culture, patience, and non-provocative behavior [5]. While these principles are well-known and often considered as common sense, they are not to be taken for granted in the online environment. Following the proper netiquette [6] and acting politely would be ideal, but unfortunately, people do not behave in an idealistic manner, as illustrated by the high degree of online hate taking place in social media [7]. Even if we acknowledge the proposition that the ultimate solution to online hate warrants a non-technical solution, it may be possible to combine social and computational techniques as hybrid solutions that reduce the number of hateful comments.

More particularly, sociological theory can be used here to evoke two principles: (1) giving individuals choices for different actions, and (2) urging them to consider their harmful behavior [8]. In the context of online hate, a social network could recommend ways to modify a detected hateful comment before posting it online. The decision remains with the human, but the machine can give options– a digital analogy advising an angry person to wait 10 seconds before speaking up. While the detection algorithms for online hate have developed rapidly [4, 9–11], there is almost no extant work on suggesting such alterations to hateful comments.

In this research, we tackle the problem of online hate, defined as offensive use of language, by modifying hatefulness of social media comments. We formulate the research objective as follows: *Remove hateful elements from a comment without eliminating the understandability or meaning of the comment*. To achieve this objective, we propose a methodology called Neural Network Hate Deletion (NNHD). This research represents exploratory and experimental work, and we note that rewriting hateful comments is a part of larger research effort, requiring more research. The purpose of this work is to provide a "minimum viable product" [12] and to show that further research on this problem is feasible from a technical point of view. In the following sections, we briefly review the related literature, the present our proposed solution, and evaluate the results with crowd experiments and manual ratings. Finally, we discuss the findings and future avenues for research.

2 Related literature

We searched for related literature in academic databases, including Google Scholar and Science Direct. The search phrases included ["online hate"], [+hateful "social media"], ["toxic comments"], and other relevant key phrases. We then manually evaluated if the found articles match our research objective. Overall, we found several thousand papers covering this topic. For example, "online hate" alone brings 2,120 results in Google Scholar. To narrow down the scope, we focus on studies that introduce computational solutions to offensive or hateful commenting in online environments. Overall, social media platforms, such as Twitter, Facebook, Reddit, and YouTube have been the most common source for data collection of online hate studies [7, 10, 13–15].

In one of the earlier studies, Sood et al. [16] focused on separating off-topic negative comments from on-topic negative comments. They collected 1,655,131 comments from Yahoo! Buzz, a news website, labeled 6,500 of them according to the presence or absence of insults and profanity, along with topics of hate. The topics were world, business, entertainment, politics, news, and general, and the target was either an author of a previous comment or a third party. The comments were labeled using Amazon Mechanical Turk, instructing the workers to focus on the intention of the comment writer to successfully label clear profanity as well as a disguised one.

The approach by Mondal et al. [10] was based on created a sentence structure that reveals hate, target, and intensity: “I <intensity> <userintent> <hatetarget>”. This allowed them to identify a number of explicit hate targets, e.g. minority groups and women. Intent is the emotion of the user; intensity is the level of emotion, and a hate target is the group receiving the emotion of dislike, animosity or hate [15]. To determine hate targets, Mondal et al. [10] designed two templates: (1) Placing a specific word before people in order to specify hate targets – black people, Mexican people, stupid people, and (2) using targets from the Hatebase, a dictionary with 1,078 hate words. Using this strategy, the most common hate categories were: race, behavior and physical.

Much of the previous work utilizes keyword dictionaries to detect hateful comments. However, one shortcoming of the dictionary-based models is that a keyword used as a hate indicator in one community may not represent hate in another community. The same concern has been raised by Sood et al. [17] whose model, using the profanity list from no-swearing.com, was able to detect 40.2% of profanity terms, the conclusion is that even the best list could not achieve good performance at detection of profanity. Keywords may also miss sarcasm, negations, and humor, as these forms of language are particularly challenging to classify [18]. Moreover, Nobata et al. [11] note that blacklists (collections of hateful words and insults) require constant updates. Sood et al. [17] point out new terminology of profane terms as a major challenge.

To overcome these challenges, Saleem et al. [14] used Labeled Latent Dirichlet Allocation (LLDA) to learn the community-specific hate topics, which were compared to baseline language from Reddit. Saleem et al. [14] achieved good performance with Naïve Bayes and Logistic Regression. The classifier trained on the community-based data achieved a better performance than a keyword-based classifier, being 10–20% more accurate on hate detection. Another finding from the literature is that combining several features tends to improve classification accuracy of online hate. For example, Nobata et al. [11] detected abusive language consisting of profanity and derogatory phrases, using four types of features: N-grams, Linguistic, Syntactic, and Distributional Semantics. The combined application provided the best performance [11]. Similar results were obtained by Salminen et al. [4] using the same feature types.

There is also a nascent line of work utilizes neural networks. For example, Djuric et al. [9] detected hateful comments using a two-step approach: 1) Paragraph2vec with a bag-of-words (BOW) neural language model, and 2) embeddings-based a binary classifier distinguishing hateful and non-hateful comments. Paragraph2vec appeared as very useful and precise, discovering some non-obvious swearing words. This method also obtained the higher area under the curve (AUC) than BOW models [9]. Badjatiya et al. [19] classified tweets into three categories: racist, sexist, neither, using deep neural networks. Benchmark dataset of 16,000 tweets was analyzed, and 3,383 were labeled as sexist, 1,972 as racist, and the remaining were labeled as neither. The proposed methods (CNN, LSTM, FastText) were better than the baseline methods (Character n-grams, TF-IDF, Bag of Words). The best accuracy was obtained when deep neural networks were combined with Gradient Boosted Decision Trees [19]. In a similar vein, Park and Fung [20] detected abusive language using three convolutional neural network (CNN) models: CharCNN, WordCNN, and HybridCNN, with character and word features. The best performance was achieved with HybridCNN and the worst with

CharCNN. When two logistic regressions were combined, they performed as well as the one-step HybridCNN, and better than a one-step logistic regression classifier [20].

Overall, there is a substantial number of previous work focusing on the detection and classification of online hate. However, there is almost no extant work on *changing the hateful comments* from the state of hateful to neutral or positive. In fact, we could not locate any prior study solely focusing on this. In previous studies, the objective is detecting and/or deleting the whole comment. However, at the same time, limiting the freedom of speech is noted as a concern in this approach [7, 16], especially when automatic moderation is applied without giving the end user a chance to take corrective action in reformulating his message. Our study aims to improve this condition by *not deleting the whole comment but only the hateful elements within it*. In other words, the social challenge is to balance moderating offensive language with the freedom of speech, while the computational challenge is altering the sentences without losing their understandability and meaning.

3 Methodology

3.1 Data collection and validation

We collect data from a major news and media company with an international audience from over 200 countries. This media organization has several channels on several platforms. For this research, we retrieve all the comments from two channels for two platforms. By using Facebook and YouTube application programming interfaces (APIs), we pull comments from videos posted on YouTube and Facebook, including 1,342,597 comments from the period of December, 2013 to January, 2018. The comments originate from all the Facebook posts and YouTube videos of the two channels in the two platforms (several thousands of content pieces). They are written by social media followers of the respective channels. Some posts or videos did not have comments since adding comments was disabled or the comments were deleted. Nevertheless, this provides both longitudinal spectrum (close to five years of commenting) and breadth (more than a million comments). Table 1 shows the breakdown of comments by platform.

Table 1. Number of retrieved comments by platform and channel.

Source	Number of retrieved comments
Channel 1 (YouTube)	780,138
Channel 2 (YouTube)	235,315
Channel 1 (Facebook)	75,002
Channel 2 (Facebook)	252,142
Total	1,342,597

In the dataset of 1,342,597 comments, 327,144 (24%) are from Facebook, 1,015,453 (76%) from YouTube. Initially, while exploring this large dataset, we observed that it contains many hateful comments. This prompted us to find ways to detect and classify them, for automated or computer-aided moderation. Overall, these comments make a useful dataset for dealing with online hate using computational means, and demonstrate a real problem touching online media producers. To make the dataset useful for classification, we automatically label these comments for hate content. To accomplish this, we first build a hateful keyword dictionary. We use a combination of two techniques to generate the dictionary of hateful phrases: (1) open coding, i.e., reading the comments and noting down phrases that are frequently used in a hateful sense [21], and (2) manually screening existing online hate dictionaries[1] for additional hateful words. Using these sources, we collected 1,098 hateful phrases. However, not all of them are present in the dataset based on mining the comments. In addition, the use of hateful expressions has been found to vary by the community using them [14]. Here, the context of the videos is diverse, but most hate, based on our open coding, revolves around political topics (e.g., Israel-Palestine, Police brutality, US presidential elections, etc.). That is why we must adjust the dictionary using open coding, i.e. manually finding hateful expressions that are typical for this dataset. Table 2 includes examples of hateful phrases and their appearance in the comments.

Table 2. Examples from the created hate dictionary. Dictionary includes 203 phrases.

Example key phrase	Example comment (hateful expressions bolded)
stupid	"Because Denmark is getting smart and Sweden is still **stupid**.
should be killed	however the rapist's **should be killed**.... lets give them a bit of their sharia law they so long for. the **fucking pigs**...."
fuck	"All the libs making a big deal of it. When in reality if it was Hillary's **bitch ass**, you **cowards** would be almost certain you had been shot. Right away blaming the 2nd amendment. Soft **pussy fucks**."
zionist	"The **zionist scumbags** are losing the battle. #FreePalestine"

Second, we apply this dictionary to social media datasets to retrieve hateful comments. Keyword matching and keyword-based features have been utilized in a similar fashion in several previous works to detect hateful comments [15, 17]. For example, Salminen et al. [4] found that simple term-level features were among the best predictors when detecting hateful online comments. As mentioned, the keyword list was compiled using a combination of open coding and existing hate keywords lists. After retrieving the hateful comments via keyword matching, we label 301,153 hateful comments and 1,041,490 non-hateful comments.

Hateful comments are automatically labeled as 1, non-hateful as 0, corresponding to a classical binary classification task. To ensure that this labeling approach works, we

[1] We compiled the list of hate words by combining open coding done on our dataset and lists of profanity or swear words available online: http://www.bannedwordlist.com/lists/swearWords.txt; https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words/blob/master/en; https://www.frontgatemedia.com/a-list-of-723-bad-words-to-blacklist-and-how-to-use-facebooks-moderation-tool/; http://onlineslangdictionary.com/lists/most-vulgar-words/

conduct a "sanity check" on a sample of 300 comments we expect to be hateful based on our automatic coding, and 300 comments we expect to be non-hateful. In the sanity check, we use crowd workers, asking them if the comment was hateful or not (options given: 'hateful', 'not hateful'). We provide the workers with examples of both categories through using test questions that also protect against unethical workers [22]. Each comment in each sample is evaluated by three crowd workers, and we use majority voting to assign the final label (e.g., the label with the highest number of ratings is chosen as the final label). The results are shown in Table 3.

Table 3. Validation of training data using crowd ratings.

	Hateful (*observed*)	Non-hateful (*observed*)
Hateful (*expected*)	84.47%	15.53%
Non-hateful (*expected*)	19.80%	80.20%

The sanity check yields 84.47% of the automatically coded hateful comments as hateful (using majority voting, where 2/3 raters agree), and 80.20% of the non-hateful as non-hateful. The observed sentiment of the comments is thus in line with the expected sentiment. However, because the ratings show some number of false positives and negatives (diagonals in Table 3), we also utilize another dataset that contains 10k manually labeled Reddit comments, to train the network. This dataset was obtained by retrieving comments from AskReddit (a sub-group of Reddit, the popular online social network) covering all the posts between January 2008 and August 2017, and then asking the crowd workers to label if a comment is hateful or not. The 10k comments from the Reddit corpus were randomly selected from a total of 16M Reddit comments. By using a combination of the above data containing hateful and non-hateful online comments, we proceed to the next step, which is training and model development.

3.2 Model development

We develop a model dubbed Neural Network Hate Deletion (NNHD). A simple example of hate deletion is as follows: "you fail once again stupid" (high hatefulness) → "you fail once again" (decreased hatefulness). We thus aim to reduce the hatefulness of the comments while retaining their understandability.

Figure 1 describes the network built for this research. In this model, we are most interested in the network activations after the time-distributed dense layer that correspond to the final classification and should correspond to the hatefulness of the tokens in the input. These will be used to score the tokens in the comments by their hatefulness'. Using this approach, each token in the tree is scored by the neural model. Before pruning the tree, we set a threshold for pruning the tree. From the tree, we remove all tokens with a score above the threshold and all tokens which depend on it or are part of a subtree with the removed token as its head.

The aim of the scoring in the ordinary text simplification task is to score the tokens by their meaning [23]. In this research, however, we score them by their hatefulness and compress the sentence to reduce its degree of hatefulness. We follow an approach in which tokens of the comment texts are scored using a neural model trained on a mixture of hand-annotated and automatically generated online comment data. The model is trained on 10,000 comments of hand-annotated Reddit comments and 10,000 of automatically keyword matched news comments. The preloaded embeddings used were 300-dimensional fastText vectors trained on wiki- and news-texts and distributed by Facebook research[2]. We set dropouts at 0.5 to reduce model overfitting.

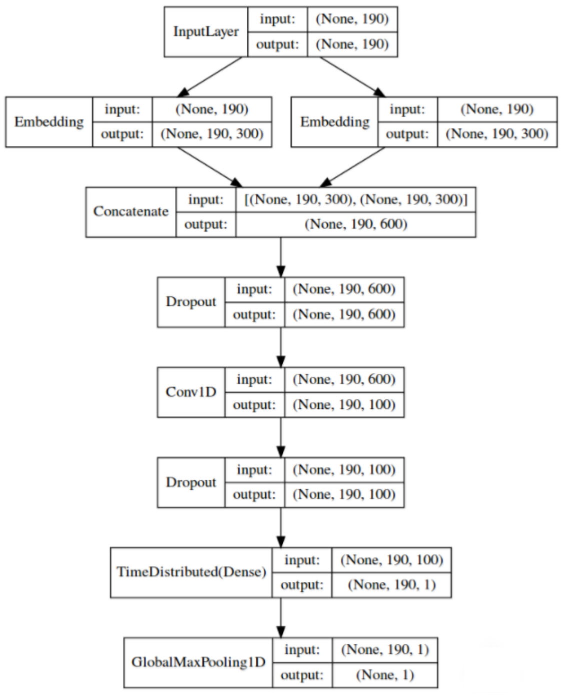


Figure 1. Description of NNHD.

Overall, we approach the problem as a text simplification task. In a text simplification task, the aim is to compress text so that it keeps its meaning as much as possible, while the output text is compressed in length while remaining grammatically intact. A widely used approach to text simplification is extractive text compression [24]. Extractive text compression achieves the aim of compression by extracting parts of the original text that contain its core meaning. The grammaticality of the output is often maximized using a language model, or dependency trees [23].

To improve the grammaticality of the output, we use dependency tree pruning, i.e. removing subsets of the sentences from hateful comments. This is because subtrees of a dependency tree are usually grammatically separate subsets of the sentence. For

[2] fastText is a library for learning of word embeddings and sentence classification created by Facebook.

example, in Figure 2, "In Ramadi" is a potentially prunable part of the example sentence. The dependency trees are parsed with UDPipe with UDv2 English model.

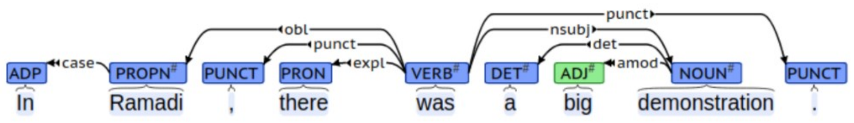


Figure 2. An Example of Dependency Tree from UD-English Treebank

To improve grammaticality, we follow Filippova and Strube [23] and produce the result text by pruning dependency trees based on the scoring produced by the model. For this approach to perform well, we build a neural model to obtain scores for the tokens. These scores are then used to prune the dependency trees for maintaining grammar. To obtain the hate score for the tokens in the sentence, we build a simple neural network classifier on our dataset to differentiate between hateful and non-hateful comments and then use activations of this model to score the tokens.

The comments are dependency parsed, part-of-speech (POS) tagged and tokenized with UDPipe [25] as a preprocessing step. UDPipe is a processing pipeline for tokenization, morphological analysis and dependency parsing with pretrained models trained on Universal Dependencies data. The network takes as its input a sequence of token indices, fetches two embeddings, one pre-trained and frozen and one trained during the model training. They are concatenated and fed through one-dimensional convolution with a window size of one. This sequence is then fed to a time-distributed dense neural network layer with sigmoid activation and its result is max-pooled to give the final prediction. Because of the max-pooling layer at the end of the network, the network predictions are based on the maximum scoring token, and as such, the model is not meant to give predictions on the comments themselves, but the model creates token weights indicating hatefulness.

4 Evaluation of results

4.1 Manual evaluation

We conduct a three-staged evaluation: (a) manual evaluation, (b) quantitative analysis of variance (ANOVA), and (c) rating of original meaning retention. First, we manually check the results of the transformation. We can see that the results vary. While not perfect, the network performs surprisingly well. This leads us to believe our method is providing satisfactory results. Table 4 contains examples from our manual evaluation.

Table 4. Examples of transformed comments. Threshold score in parentheses.

Hateful comment	Transformed comment
“another narrow minded offensive point of view from an ignorant jerk who wants to spread hate wants to make people feel low as he is garbage”	“another narrow minded offensive point of view” [0.5]
“without killing most americans in asia it’s hard to achieve peace”	“it’s hard to achieve peace” [0.7]
“fucking america why america is trying to have an impact in europe”	“why america is trying to have an impact in europe” [0.9]

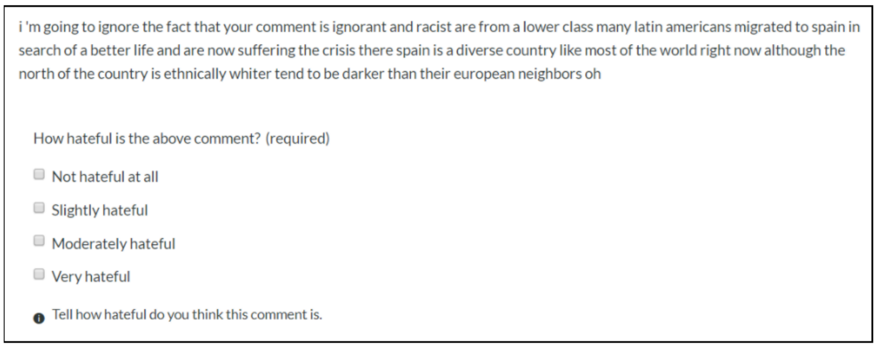
In addition, we review the hate scores of the words and find that they seem indeed highly toxic. Table 5 shows the ten most hateful words learned by the network. It is worth noting that these words were learned from the training data without any other supervision than the general label “hateful”. The algorithm also deleted some comments completely – we call such comments ‘beyond repair’ because the comment has no other meaning to retain but hate. On average, the share of completely removed comments was 8.52% for the 0.5, 6.03% for the 0.7, and 4.96% for the 0.9 thresholds.

Table 5. Hateful words learned by the network without supervision. Showing top 10.

Hate Score	Term	Type (based on POS-tagging)
0.99793506	'scum'	noun
0.9974718	'coward'	noun
0.9974164	'fuckin'	adposition
0.99735606	'cunts'	noun
0.9971541	'dumbass'	noun
0.9971535	'fuckers'	noun
0.9969387	'greedy'	adjective
0.9955591	'bastards'	noun
0.9946931	'motherfucker'	noun
0.99336666	'bitch'	verb

4.2 Crowd experiments

Second, to evaluate, we want to know how many comments will keep their meaning after applying neural network hate deletion. This measures the successfulness of fixing the hateful comments. We opt for crowd evaluation, as done in prior literature dealing with online hate ratings [10, 15]. Here, we use CrowdFlower, a platform that offers good options for crowdsourced tasks. We measure two types of perceptions of crowd workers: (1) hatefulness (“*How hateful is this comment?*”) and (2) understandability (“*Is this comment easy to understand?*”). Figure 3 illustrates the created tasks.



i 'm going to ignore the fact that your comment is ignorant and racist are from a lower class many latin americans migrated to spain in search of a better life and are now suffering the crisis there spain is a diverse country like most of the world right now although the north of the country is ethnically whiter tend to be darker than their european neighbors oh

How hateful is the above comment? (required)

- ☐ Not hateful at all
- ☐ Slightly hateful
- ☐ Moderately hateful
- ☐ Very hateful

Tell how hateful do you think this comment is.

Figure 3. Example question asked from the crowd labelers on hatefulness of the comments.

We evaluate both the original hateful comments and the manipulated comments with three different thresholds of hate scoring (0.5, 0.7, and 0.9). Overall, we carry out eight data collection tasks: *Baseline Hateful* (original hateful comments), *Baseful Senseful* (original hateful comments), *Hateful_05*, *Hateful_07*, *Hateful_09* (each measuring hatefulness with transformation at a given threshold of hate score), *Senseful_05*, *Senseful_07*, and *Senseful_09* (each measuring understandability with transformation at a given threshold of hate score). The tested comments were randomly sampled from the transformed comments, pairing the original comments with their transformed versions at each threshold level. For both tasks, we use a 4-point scale, so that a comment is 'Very hateful' (4), 'Moderately hateful' (3), 'Slightly hateful' (2), or 'Not hateful at all' (1); and 'Very easy to understand' (4), 'Quite easy to understand' (3), 'Slightly difficult to understand' (2) or 'Difficult to understand' (1).

We undertook several measures to ensure the quality of crowd ratings:

- **Test questions:** We created 8 test questions with known values for each task, according to the recommendation of the platform. Test questions quiz the raters randomly; answering correctly makes workers eligible for the job, whereas failing to answer them correctly removes a worker from the task.
- **Max Judgments / Rater:** We chose 50, so that with 300 x 5 = 1500 ratings, there is 50/1500 = 3.33% maximum impact per rater, thus avoiding individual bias affecting the results.
- **Minimum time for a rater to complete a page of work:** 25 seconds; a page has 5 comments and we expect it would take a rater at least 5 seconds to read and rate each (5 x 5 = 25 seconds).
- **Quality Level:** Level 2 ("Higher Quality"), consisting of a smaller group of more experienced, higher accuracy raters.
- **Judgments per Row:** 5, to get more information on the preferences, as is suggested for more subjective classification tasks [26].
- **Price per Judgment:** 4 cents (USD), a 33% increase in the price suggested by the crowdsourcing platform.

4.3 Statistical analysis

We then compare if the perceptions are significantly different between the group that saw the original comment and the group that saw the comment modified by the algorithm. The purpose is to evaluate if the neural network changed the comment successfully to remove hate while retaining the understandability of the comment. In other words, the threshold is the hate score of the algorithm we want to test (experimental condition), comparing to comments that received no transformation.

Testing with a multivariate analysis of variance (MANOVA) model, we find that understandability does not have significant changes between transformed and non-transformed comments, or between the threshold levels. However, for hatefulness, applying transformation results in a very significant decrease in perceived hatefulness. The 0.7 and 0.9 thresholds are statistically similar, with the 0.5 threshold showing the largest decrease in terms of hatefulness. In other words, lowering the threshold for transformation yields a significant decrease of hatefulness without resulting in discernible changes to the understandability of the comment.

The dataset was analyzed using ANOVA tests to determine which questions had statistical different responses by group, followed by Tukey's HSD post-hoc tests to determine which groups significantly differed from the others [27]. Although the dependent variables only have a range of 4 points, research has shown that it is acceptable to use parametric methods in these scenarios [28]. The omnibus test indicated that significant differences between levels could be found for at least one variable (Roy's Largest Root = 0.271, F(3, 1048) = 94.500, $p < 0.001$). As such, this was followed-up by individual analysis for both sense-making and hatefulness. These are presented in Table 6, which compares the baseline to the various thresholds.

Table 6. Effects of transformation on understandability and hatefulness.

Level	Sense-making	Hatefulness
Threshold = 0.5	-0.011 (0.070)	-1.631 *** (0.103)
Threshold = 0.7	-0.015 (0.070)	-1.461 *** (0.103)
Threshold = 0.9	-0.038 (0.070)	-1.417 *** (0.103)
Baseline (No transformation)	-	-
R2	0.000	0.213

Note: Standard errors are in parenthesis. * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

First, it becomes evident that transformation has no impact on the degree of sense-making (F(3, 1048) = 0.128, $p = 0.944$). On the other hand, hatefulness presents a substantial decrease once transformation is applied (F(3, 1048) = 94.350, $p < 0.001$). This effect becomes more pronounced as the threshold is lowered; at a 0.9, hatefulness is 1.417 points lower on average than the baseline, which becomes 1.461 points lower at a 0.7 threshold, and finally 1.631 points lower at a 0.5 threshold. Post-hoc tests, using

Tukey's HSD, were conducted to ascertain if these differences were statistically significant. The results are reported in Table 7.

Table 7. Comparison of effect by level.

Threshold (i)	Threshold (j)	Mean Difference (i – j)
	Threshold = 0.7	-0.170
Threshold = 0.5	Threshold = 0.9	-0.213
	Baseline	-1.630 ***
	Threshold = 0.5	0.170
Threshold = 0.7	Threshold = 0.9	-0.043
	Baseline	-1.460 ***
	Threshold = 0.5	0.213
Threshold = 0.9	Threshold = 0.7	0.043
	Baseline	-1.417 ***
	Threshold = 0.5	1.630 ***
Baseline (No transformation)	Threshold = 0.7	1.460 ***
	Threshold = 0.9	1.417 ***

Note: * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

Based on these results, it is possible to observe that although all levels of threshold differ from the baseline ($p < 0.001$), they do not statistically differ among themselves, despite some numerical differences being noted. As such, we can conclude that transformation does not impact sense-making, and only minimal transformation needs to be conducted to obtain optimal hatefulness reduction.

Finally, we evaluate how well the modified comments retain the original meaning. This is done by three trusted raters each independently coding a sample of 100 comment pairs (modified–non-modified) for each threshold, in total 300 comment pairs per rater. For each pair, the raters read the original and modified comment and rate it on a scale of 1 to 5, as explained below:

- 1 = Not at all the same meaning
- 2 = Very different meaning
- 3 = Moderately the same meaning
- 4 = Very similar meaning
- 5 = Exactly the same meaning

The raters were explained the purpose of the task, namely that of evaluating if automatically changed comments have their original meaning. They were instructed to give their objective evaluation of each comment pair. In the borderline cases, raters were advised to choose conservatively, that is, if they hesitate, then rate the meaning to have changed. For each comment pair, we calculate the average original meaning rating given by the three raters. Figure 4 shows the average scores by threshold. We can see that as the threshold for deleting the hateful parts decreases, the comments start to lose their original meaning (3.73 for 0.9 threshold versus 3.29 for 0.5 threshold). The reason for this is logical since, on one hand, the number of false positives increases, and, on the other hand, as more hateful content is deleted it is not simultaneously replaced by a more neutral text, leaving some of the comments "crippled". As such, the result shows the limits of the current approach, although even at the lowest threshold, the comments retain "moderately the same meaning" on average.

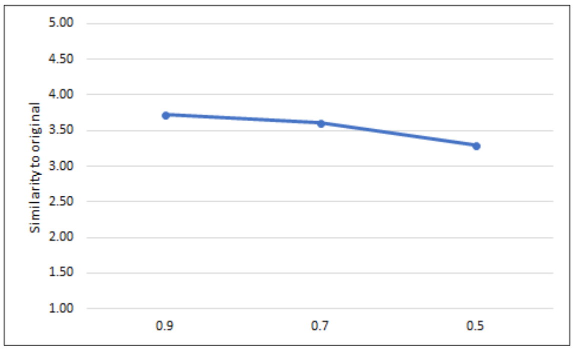


Figure 4: Evaluation of original meaning retention.

5 Conclusion and discussion

We have demonstrated an approach of neural network modeling to remove the hateful elements from comments. The approach significantly reduces the hatefulness of the comments while not significantly resulting in the loss of their understandability. The results on the loss of meaning are mixed: there seems to be a certain trade-off between removing hateful elements and losing the meaning, but on average the comments seem to keep their original meaning when applying the NNHD algorithm.

The positive results on understandability can be explained in different ways. First, it may be that the human understanding is robust to simple deletion of parts of comments, so that when one sentence of the comment gets reduced or deleted, the other sentences are used as context to fill in the gaps. Second, we speculated that if the network removes mostly adjectives and adverbs from the sentences, it is logical that the sentence will remain understandable because these word types are used to enhance a point, unlike verbs and nouns that are central for the understandability. However, a POS analysis revealed that the network, in fact, removes most often nouns (5,870 deletions) and verbs (2,785), while adjectives rank only on third place (1,790). Thus, this explanation, although logical, cannot be seen valid. Thus, we postulate that, as a side effect, removing hateful elements from online comments can reduce the linguistic complexity of the comments and thus make them more understandable.

Regarding the limitations of our work, we found three types of comments when looking at the results our algorithm: (1) **beyond repair**, i.e., comments that cannot be transformed without losing everything, (2) **doable but hard**, i.e., comments that a human can alter but that are difficult for machine to transform, and (3) **easy fix**, i.e., comments that can be transformed by simple removal of hateful expressions. The algorithm works well for the last type, but some comments it completely deletes, and some comments may lose their original meaning. Thus, more work is needed to develop automatic hate correction algorithms, especially utilizing more sophisticated methods for language generation, not only removal of hateful elements. This is needed because, as our findings show, sometimes the removal results in the loss of meaning. For example, in some cases, NNHD may disrupt the syntactic structure of the sentence. Thus, more advanced models for word replacement are needed to generate fluent, non-hateful language that retains the original meaning of the sentence.

In terms of practical implications of this work, we maintain that providing choice to end users to opt for non-hateful self-expression is a potentially fruitful approach in decreasing online hate. For example, it is possible to develop a system that recommends users ways to fix their hateful comments. Even though a user study is beyond the scope of this research, our approach provides a starting point for testing how receptive agitated online users are to this type of conditioning and whether they would change their behavior when exposed to hate-free forms of expression.

References

1. Burnap, P., Williams, M.L.: Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Science. 5, 11 (2016).
2. Del Vicario, M., Vivaldo, G., Bessi, A., Zollo, F., Scala, A., Caldarelli, G., Quattrociocchi, W.: Echo Chambers: Emotional Contagion and Group Polarization on Facebook. Scientific Reports. 6, 37825 (2016).
3. Kramer, A.D.I., Guillory, J.E., Hancock, J.T.: Experimental evidence of massive-scale emotional contagion through social networks. PNAS. 111, 8788–8790 (2014).
4. Salminen, J., Almerekhi, H., Milenković, M., Jung, S., An, J., Kwak, H., Jansen, B.J.: Anatomy of Online Hate: Developing a Taxonomy and Machine Learning Models for Identifying and Classifying Hate in Online News Media. In: Proceeding of The International AAAI Conference on Web and Social Media (ICWSM 2018). , San Francisco, California, USA (2018).
5. Wright, L., Ruths, D., Dillon, K.P., Saleem, H.M., Benesch, S.: Vectors for Counterspeech on Twitter. In: Proceedings of the First Workshop on Abusive Language Online. pp. 57–62 (2017).
6. Scheuermann, L., Taylor, G.: Netiquette. Internet Research. 7, 269–273 (1997).
7. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated Hate Speech Detection and the Problem of Offensive Language. In: Proceedings of Eleventh International AAAI Conference on Web and Social Media. , Québec, Canada (2017).
8. Bamberg, S.: Changing environmentally harmful behaviors: A stage model of self-regulated behavioral change. Journal of Environmental Psychology. 34, 151–159 (2013).
9. Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., Bhamidipati, N.: Hate Speech Detection with Comment Embeddings. In: Proceedings of the 24th International Conference on World Wide Web. pp. 29–30. ACM, New York, NY, USA (2015).
10. Mondal, M., Silva, L.A., Benevenuto, F.: A Measurement Study of Hate Speech in Social Media. In: Proceedings of the 28th ACM Conference on Hypertext and Social Media. pp. 85–94. ACM, New York, NY, USA (2017).
11. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive Language Detection in Online User Content. In: Proceedings of the 25th International Conference on World Wide Web. pp. 145–153. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2016).
12. Ries, E.: The Lean Startup. Penguin Books Ltd (2011).

13. Mohan, S., Guha, A., Harris, M., Popowich, F., Schuster, A., Priebe, C.: The Impact of Toxic Language on the Health of Reddit Communities. In: SpringerLink. pp. 51–56. Springer, Cham (2017).
14. Saleem, H.M., Dillon, K.P., Benesch, S., Ruths, D.: A Web of Hate: Tackling Hateful Speech in Online Social Spaces. arXiv:1709.10159 [cs]. (2017).
15. Silva, L., Mondal, M., Correa, D., Benevenuto, F., Weber, I.: Analyzing the Targets of Hate in Online Social Media. In: Proceedings of Tenth International AAAI Conference on Web and Social Media. , Palo Alto, California (2016).
16. Sood, S., Antin, J., Churchill, E.: Profanity Use in Online Communities. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1481–1490. ACM, New York, NY, USA (2012).
17. Sood, S.O., Churchill, E.F., Antin, J.: Automatic identification of personal insults on social news sites. J. Am. Soc. Inf. Sci. 63, 270–285 (2012).
18. Rajadesingan, A., Zafarani, R., Liu, H.: Sarcasm detection on twitter: A behavioral modeling approach. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. pp. 97–106. ACM (2015).
19. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep Learning for Hate Speech Detection in Tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 759–760. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2017).
20. Park, J.H., Fung, P.: One-step and Two-step Classification for Abusive Language Detection on Twitter. arXiv preprint arXiv:1706.01206. (2017).
21. Strauss, A., Corbin, J.: Grounded theory methodology. Handbook of qualitative research. 273–285 (1994).
22. Geiger, D., Seedorf, S., Schulze, T., Nickerson, R., Schader, M.: Managing the Crowd: Towards a Taxonomy of Crowdsourcing Processes. Americas The. 1–11 (2011).
23. Filippova, K., Strube, M.: Dependency Tree Based Sentence Compression. In: Proceedings of the Fifth International Natural Language Generation Conference. pp. 25–32. Association for Computational Linguistics, Stroudsburg, PA, USA (2008).
24. Alguliev, R., Aliguliyev, R.: Evolutionary algorithm for extractive text summarization. Intelligent Information Management. 1, 128 (2009).
25. Straka, M., Hajic, J., Strakova, J.: UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. Presented at the Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016) , Portorož, Slovenia (2016).
26. Alonso, O., Marshall, C.C., Najork, M.: Debugging a Crowdsourced Task with Low Inter-Rater Agreement. In: Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries. pp. 101–110. ACM, New York, NY, USA (2015).
27. Norušis, M.J.: IBM SPSS Statistics 19 Statistical Procedures Companion. Prentice Hall (2011).
28. Norman, G.: Likert scales, levels of measurement and the “laws” of statistics. Adv Health Sci Educ Theory Pract. 15, 625–632 (2010).