A System to Filter out Unwanted Social Media Content in Real-time on iPhones

Anonymous ACL submission

Abstract

Social media users are often harassed. This paper presents a patented system to filter out harassing content before it reaches the recipient. Our first version is for the iPhone. To detect harassment, we adopted sentiment analysis with a supervised learning approach that combines Machine Learning (ML) text classifiers with a lexicon approach that provides a feedback loop to retrain the ML model with unknown terms. Because good data is essential to obtain the best output of any system, we focused on validating our labeled data. Our results on static and real-time data have an accuracy of, respectively, 90% and 94%. Our labeled data validation allows us to correct labels; we also realized the need to increase the number of sets in our lexicons. Our prototype demonstrates that we are able to build an AI infrastructure to filter out harassment on an iPhone in real-time with good results.

1 Introduction

Social media platforms such as Twitter have massively advanced human connectivity. Every day, there are about 500 million tweets sent globally, or about 6,000 tweets a second.[1] Unfortunately, many people are being exposed to unwanted, harassing, and even threatening content. Harassment is disproportionally aimed at women,[2] people of color,[3] and the LGBTQ+ population (Wilson and Cariola, 2020). Though virtual, this content has had a strong impact. Thirty percent of women journalists have considered leaving their profession,[4] and suicide and self-harm rates are double for adults under 25 who have been victimized by online harassment.[5] The fear of harm and the mental health impact from online harassment are real. The explosive growth of social media has made it difficult for providers to effectively track and remove unwanted content. While companies do attempt to track and remove content, they rely heavily on manual reports from users.[6] Our approach is to filter out harassment using text classifiers and lexicons on the receiver end. For real-time data, the quality of the data is important to obtain good results. Training any models with data that were incorrectly labeled affects the models' performance on real-time data. Therefore, we decided to validate the labeled data. The accuracy of the performance of a model on real-time data needs requires that the training data cover a huge diversity of content. Therefore, we need to expand the model knowledge with the following steps: a lexicon that acts as an adaptive filter to the classifier by catching unknown content to the model; and which searches for content with Search API functions calls. The

[1] See https://www.internetlivestats.com/twitter-statistics/. Accessed: 01-02-2021.

[2] See https://www.amnesty.org/en/latest/news/2017/11/amnesty-reveals-alarming-impact-of-online-abuse-against-women. Accessed: 12-18-2020.

[3] See https://www.pewresearch.org/fact-tank/2017/07/25/1-in-4-black-americans-have-faced-online-harassment-because-of-their-race-or-ethnicity/. Accessed: 12-18-2020.

[4] See https://www.iwmf.org/programs/online-harassment/. Accessed: 12-18-2020.

[5] See https://www.comparitech.com/internet-providers/cyberbullying-statistics. Accessed: 12-18-2020.

[6] See https://blog.twitter.com/en_us/topics/company/2020/An-update-on-our-continuity-strategy-during-COVID-19.html. Accessed: 12-18-2020.

methodology provides the blueprint (see Fig.1) on how to build the different steps of the AI infrastructure in real-time:

- Data Engineering: collecting and labeling data;
- Modeling: training Machine Learning (ML) models and running evaluation metrics;
- Deployment: implementing Representational State Transfer Application Programming Interface (Rest API) and Webhook data transfer, setting authorization requests, uploading models on devices;
- Reports: storing the sender data information into a graph database called Neo4j to evaluate the spread of the harassment among users;
- Analyze Results: evaluating the system in production, writing tests for the multiple components, and providing an evaluation matrix for the results.

The AI infrastructure is a life cycle that allows the system to adjust itself by retraining the models with additional data after obtaining output results in real-time. We have incremented our label data size and validated our label data, identified the underlying patterns that make it possible to use automation to track and filter harassing data in real-time.

2 Background and Prior Art

In a January 19, 2019 interview, Jack Dorsey, one of the founders and the Chief Executive Officer of Twitter revealed how surprised he and his colleagues were at the prevalence of social media harassment: “We weren’t expecting any of the abuse and harassment, and the ways that people have weaponized the platform.” Dorsey explained that they felt “responsible about it.”[7] Social media companies allow users to report abuse and require verification by e-mail addresses, phone numbers, or the identification of pictures to prevent robotic contact attempts. But these mechanisms have proven fruitless to stop the harassment.
Improvements in ML technology allow harassment to be countered.

The Times (London), for instance, partnered in 2016 with a Google-owned technology incubator to score incoming comments by comparing them

[7] See https://www.huffingtonpost.com/entry/jack-dorsey-twitter-interview_us_5c3e2601e4b01c93e00e2a00. Accessed: 12-18-2020.

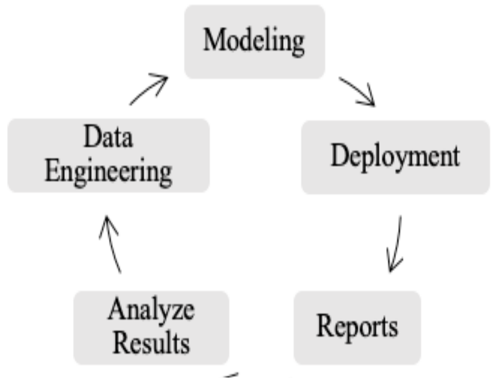

Figure 1: AI Infrastructure

to more than 16 million moderated Times comments going back to 2007.[8] Email Software uses text classifiers to determine whether incoming mail is sent to the inbox folder or the spam folder[9].
Ogudo (2019) uses sentiment analysis and text mining to analyze social media content. Heba (2016), Kolchyna (2015), and Medhat (2014) describe the following three classification types for sentiment mining that were evaluated.

1) K-neighbors, Decision tree, Naïve Bayes and Support Vector Machine (Vinodhini et al., 2013);
2) Passive-Aggressive Algorithm Based Classifier, Language Modeling Based Classifier, Winnow (Cui et al., 2006);
3) Machine learning classifiers for sentiment analysis approaches are the lexicon-based approach and the learning approach build classifier trained with labeled data (Bhuta et al., 2014; Sadia 2018);

Another classifier type is the Maximum Entropy classifier that combines an ensemble of classifier approaches (Perikos et al., 2016; Kharde, 2016).

3 Methodology

The methodology utilized to filter out harassment data on real-time data consists of the following steps:

- data engineering (collect and label the data),
- modeling (train the models with the labeled data, evaluate the model on static data),

[8] See https://www.nytimes.com/2017/06/13/insider/have-a-comment-leave-a-comment.html Accessed: 12-18-2020.
[9] https://developers.google.com/machine-learning/guides/text-classification/?hl=ID-id&skip_cache=false%22.

- deployment (deploy the models onto the iPhone device),
- report how the harassment is spread.

The training of ML text classifiers is done with English labeled data and Italian labeled data. Only our models trained with English data are deployed on an iPhone. We use a lexicon, called a "bag-of-words", to act as a feedback loop for retraining our models with unknown words to the model (see Fig.2). The unknown words to the model and the sender and their friends' names are collected. The Program Collecting Data searches and collects tweets on Twitter using search API with that specific term and/or with the sender name. With bag-of-words and the Program Collecting Data, we expanded our initial set of labeled data to approximatively 70,000 English labeled tweets in order to train the model. Figure 3 describes the system: how incoming data are processed in order to solve the harassing issue on social media. We apply an ML classifier to the incoming content. In the first version for the iPhone, a text classifier model (from Apple Core ML 3) determines if the incoming data is harassing. The text classifier model separates the data into two sets: the harassment data set and the neutral data set. Only the neutral data are displayed to the receiver; the harassing content is filtered out and can be accessed with a different Tabbar.

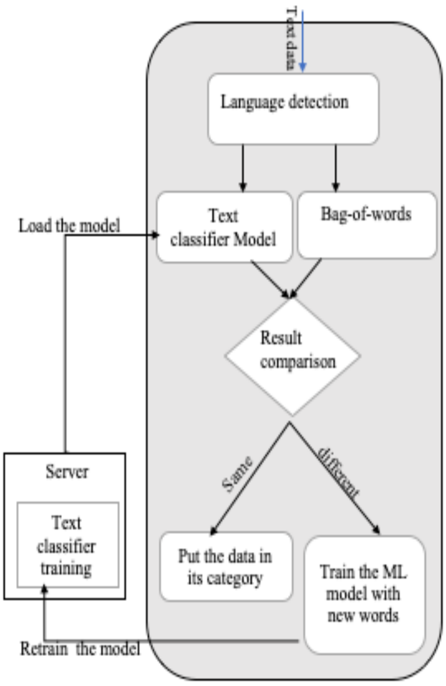

Figure 3: Bag-of-words acting as an adaptive filter to the ML text classifier

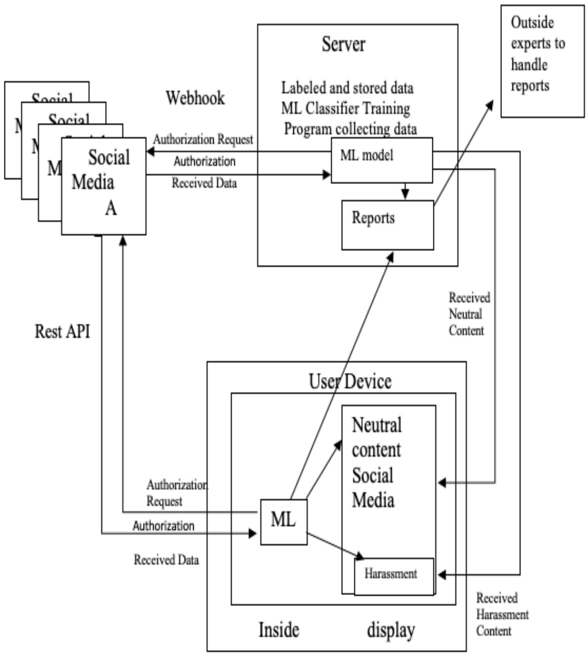

Figure 2: System

3.1 Data

We collected two sets of labeled data, one with English data and the other with Italian data.

3.1.1 English Data

We merged four different available datasets to create a general and comprehensive input dataset by leveraging their annotation schemes into a binary "harassment" and "neutral" classification. The datasets were crowdsourced:

- A corpus of more than 16,000 tweets, annotated with labels such as Racism, Sexism, and Neither (Waseem and Hovy, 2016). The labels conveying harassing content were changed into "harassment" and the "neutral" data was kept as is.
- A corpus of 35,000 tweets, with 15% positive harassment examples and 85% negative examples (Golbeck et al., 2017).
- 7,321 tweets with tweet ID, bullying, author role, teasing, type, form, and emotion labels were all converted into ”harassment" tweets (Xu et al., 2012).
- A corpus of 25,000 tweets is annotated with the labels "hate speech", "offensive language" or "neither" (Davidson et al., 2017).

The system collects text data and labels it internally in two different ways:

- Program Collecting Data is a python program using Twitter Search API to collect content data with specific terms or a specific user. To collect neutral content, we search individuals that are known to have empathetic personalities. The program collects the content of their tweets, screens them, and labels the tweets. For harassing content, the program using Search API searches for specific harassing terms or harassing individuals on Twitter.
- The bag-of-words act as an adaptive filter to increase the data set size by retraining the text classifier with content yet unknown to the model.

3.1.2 Italian Data

Two hundred thousand tweets were collected with distance supervision by allocating "hateful" or "neutral" labels according to the source of the content (Merenda et al., 2018).

3.2 Labeled Data Validation

Our labeled English datasets originated from academic sources and were mainly collected with crowdsourcing. We recognize the implications of merging datasets that have been compiled using different annotation schemes. Some, for example, use a crowd-sourced hate speech lexicon to collect tweets containing hate speech keywords (Davidson et al., 2017). Others use terms in tweets that contain hate speech and references to specific entities (Waseem and Hovy, 2016). Good data are essential to obtain good output results. Training any model with bad data – data that were labeled incorrectly – affects the performance of the model especially with real-time data. Therefore, we decided to validate the accuracy of the labeling before training our models with the labeled data. To assess the quality of the labeled data, we are using the same lexicon that we use as a feedback loop to retrain the models during the deployment. We are evaluating our labeled data against the content of the lexicon lists. At first, we only had one list of harassing words. During the validation, we realized our need to extend the number to at least five different lists of sensitive words and expressions. At a later time, the number of lists might increase depending on the data needs.

The first list consists of hardcore harassing terms. The second list has words evincing a milder harassing tone; the third list has terms that have a double meaning, with one of the meanings being harassing; the fourth list contains phrases connecting the sub-list of "bad action" with the sub-list of the intended recipient of those bad actions; the fifth list contains harassing emojis.

We are evaluating our labeled data against the content of the lexicon lists. In the labeled data set, if any hate-related term is found in tweets labeled as neutral, we changed the label to harassment. On the other hand, if no terms were found in tweets labeled as harassment, we rebalanced the annotation by labeling them as neutral. Following this method, we changed 1,880 labels from "neutral" to "harassing" from the labeled data set.

4 Modeling

The text classifiers train machine learning models that are uploaded to the iPhone onto two applications to classify incoming natural language text. We choose the Core ML 3 platform from Apple [10] and the Auto ML [11] platform from Google to classify the annotated data we had gathered to filter online harassment on incoming social media data. The Apple Core ML 3 text classifier and the AutoML classifier have been trained to recognize a pattern in the text, such as sentiments expressed in a sentence.

Core ML 3 framework provides several fundamental Natural Language Processing (NLP) building blocks such as language identification, tokenization, part of speech tagging, lemmatization, and named entity recognition. Google provides a comprehensive text classifier guideline that allows for the appropriate text classifier to be built. Another interesting feature of these models is that the NLP functionalities are provided across several different languages. For instance, Core ML 3's sentiment analysis API is available in different languages. Core ML 3 uses different classification algorithms. To classify the data, the text classifier algorithm running internally is the MaxEnt algorithm. The MaxEnt combines the following

[10] See https://developer.apple.com/documentation/createml/mltextclassifier. Accessed: 12-31-2020.

[11] See https://cloud.google.com/natural-language/automl/docs. Accessed: 12-31-2020.

classification types: K-neighbors, Decision Tree, Naïve Bayes and Support Vector Machine.

5 Static Data Results

Our results on static data are obtained with two different frameworks; one is the Core ML 3 from Apple and the other is Auto ML from Google.

5.1 Core ML 3 Text Classifier Training & Testing

Core ML 3 framework trains different models and selects using the MaxEnt algorithm . English data consist of 78,533 inputs after removing the duplicates. Our English data sets are not well balanced (see Table 1).

Harassment	Neutral
33%	67%

Table 1: Distribution –English Dataset

The ratio of harassing tweets on the Twitter app is much smaller than 33%, around 3% to 11%. For the Italian data, the input data consist of 199,020 inputs with 50% labeled as harassing content and 50% labeled as neutral content (see Table 2).

Harassment	Neutral
50%	50%

Table 2: Distribution –Italian Dataset

For English data, the MaxEnt training has a training set of 49,873 inputs and a validation set of 12,767 tweets. Each iteration of the MaxEnt training is evaluated on the validation set. The model reached 90.21% accuracy on the test set, consisting of 15,893 English-language tweets (see Table 3). The classifier error on the test data is 9.64%.

The Italian training data consists of 127,177 inputs. The evaluation accuracy on the Italian language data is 88.61% (see Table 3).

English dataset	Italian dataset
90.21%	88.61%

Table 3: Evaluation Accuracy

The evaluation accuracy and the classification error are useful metrics only when the data is well-balanced between categories. The precision and recall on the harassment set (see Tables 4 and 5) reflect more accurately how the model is performing on the harassment set and the neutral set. For instance, the precision and recall for harassment (English language) in Table 4 are, respectively, 84.26% and 85.56%; while for the neutral set they are, respectively, 93.26% and 92.59%. This difference reflects that the model's predictive power is stronger when it comes to finding neutral content.

Class	Precision	Recall	F1
Harassment	84.26%	85.56%	84.90%
Neutral	93.26%	92.59%	92.92%

Table 4: Precision & Recall Core ML 3 English Results

The Italian language results (see Table 5) are similar to the English language model, albeit with a small difference; the English language model detects the neutral tweets better and the Italian language model detects harassment and neutral at a similar ratio.

Class	Precision	Recall	F1
Harassment	89.38%	87.07%	88.21%
Neutral	87.92%	90.10%	89.00%

Table 5: Precision & Recall Core ML 3 Italian Results

The two datasets differ in size, distribution, annotation, and compilation criteria. However, the results we obtained show that the English and Italian results from Tables 3, 4, and 5 are in the same range. The Core ML 3 training of the models took 3.36 seconds for English data and 11.6 seconds for Italian data.

5.2 AutoML Text Classifier Training & Testing

Google Cloud Natural Language API provides content classification, sentiment detection, and extracts entities and syntax analysis. AutoML Natural Language features custom entity extraction and custom sentiment analysis. The training set consists of 62,575 English tweets. The validation and testing set consist of 7,822 labeled tweets each.

The Italian data training set consists of 99,938 inputs. The Auto ML Text classifier is still a beta version and the maximum input data that its structure can take is 100,000 inputs. The Italian testing set consists of 9,994 inputs.

For both languages, the Auto ML text classifier training took from 7 to 12 hours. Table 6 displays the evaluation accuracy of the models training with Auto ML text classifiers. The English data accuracy is 94.36% and the Italian data accuracy is 91.74%. The confusion matrices are shown in Tables 7 and 8. We note that the matrix cells labeled “Harassment/Harassment” have a percentage range from 87% to 95%, respectively, for the English and Italian languages. Tables 9 and 10 show the precision and recall results for the English and Italian data sets.

English dataset	Italian dataset
94.36%	91.74%

Table 6: Evaluation Accuracy

True\Predict	Harassment	Neutral
Harassment	87%	13%
Neutral	2%	98%

Table 7: Confusion Matrix Auto ML English

True\Predict	Harassment	Neutral
Harassment	95%	5%
Neutral	12%	88%

Table 8: Confusion Matrix Auto ML Italian Results

Class	Precision	Recall
Harassment	95.44%	86.88%
Neutral	93.91%	97.99%

Table 9: Precision & Recall Auto ML English Results

Class	Precision	Recall
Harassment	89.42%	95.04%
Neutral	94.47%	88.30%

Table 10: Precision & Recall Auto ML Italian Results

The evaluation accuracy results obtained with Core ML and Auto ML with the English and Italian data sets are in the same range. Table 11 reflects the good results obtained with an evaluation accuracy ranging from 88.61% to 94.36%.

Evaluation Accuracy	English	Italian
Core ML	90.21%	88.61%
Auto ML	94.36%	91.74%

Table 11: Evaluation Accuracy

6 Deployment

Only English Models were deployed with Testing Models application to evaluate the accuracy of the models on real-time data. We first implemented the application for the iPhone because the upload of their classifier models onto the device is a simpler process that has been available since July 2018. Android development will be done at a later time. We upload the English model and the bag-of-words to the iPhone. The bag-of-words acts as an adaptive filter; it catches terms unknown to the model. In the first version, the uploaded bag-of-words on the iPhone is only one set of harassing terms. In the next version the number of sets will increase to five (see §3.2). The bag-of-words filters the data with the following constraints: content defined as harassing has at least one word from the bag-of-words; when no term from the bag-of-words is found in the content, the content is defined as neutral. Fig. 2 shows a flowchart of the bag-of-words serving as an adaptive filter for the model. First, language detection is applied to the data to determine its language. Then, a corresponding text classifier is loaded to process the incoming data. The classifier labels the incoming content as harassing or neutral. In parallel, the data go through the bag-of-words filter. Results from the model and the bag-of-words filter are compared. If the model and filter results are the same, then the data are placed in the corresponding category. If the results differ, we have two possibilities:

1: If a hardcore harassing term from the bag-of-words is detected in tweet content and the model had categorized the tweet as neutral, then the decision of the filter overrides the model.

2: If the model categorizes a tweet as harassment and no harassing term from the bag-of-words is present, the content is defined as neutral.

For the next version, we will integrate the five sets of the bag-of-words such that: the definition of harassing content will have at least one term from any of the following set: hardcore harassing terms (first list), the sub-list of “bad action” with the sub-list of the intended recipient of those bad actions (fourth list); and harassing emojis (fifth list). (See §3.2.)

The neutral content may include words from the second list with moderate words (e.g., the word "stupid") and the third list with double meaning terms. (See §3.2.) We will also modify the second possibility. As modified, if the model categorizes a tweet as harassment, yet no harassing term from any of the first, fourth and fifth list is found in the tweet, then the content is further analyzed and sender history is taken into consideration.

The discrepancy between model and bag-of-words results is reported to the server for further analysis. Program Collecting Data collect tweets containing one or several of those terms and retrain the model with the collected tweets.

7 Real-time Data Results

Figure 4: Model Testing application on an iPhone, Neutral Tweets are displayed with TabBar set to Tweet

The Model Testing application that includes previously English trained models is uploaded on the device. The application purpose is to test our model with real-time data. The application contains a list of 20 user names previously gathered with Program Collecting Data. The user name list is created from different sources. The list of user names contains names from people with diverse backgrounds. The list is composed of the friends of the sender's tweets. The tweets of the senders were previously labeled and the models trained with them in real-time, from the Twitter platform, the Model Testing application requests, and with REST API 120 tweets for each user's list. The tweets are the most recent tweets sent by each user. The previously trained model (not trained with those tweets) filters the tweets into two categories: harassment and neutral. On the device, tweets from the list of names are displayed. The tweets (which are real-time data) were unknown to the Model, the bag-of-words and our development team. As a result, our deployment testing set consists of the last 120 sent tweets from each user's names list. The neutral tweets are displayed on the main screen; while the TabBar allows the harassing content to be accessed. The Model Testing application is a way to evaluate how text classifier is filtering out harassment on real-time data content.

On Twitter, a search for U.S. Congresswoman Maxine Waters shows that she receives a lot of harassing tweets. The names of harassing individuals were collected and added to the user name list.

Figure 5: Model Testing application on an iPhone, Harassing Tweets are displayed with TabBar set to harassment

Fig. 4 displays the neutral tweet content with the TabBar set to Tweet. Fig. 5 is a screenshot of the Model Testing application with TabBar harassment checked. Results output were collected in debug mode with a print console function. On the device, 1,890 tweets were displayed and the accuracy was 94% with a wide margin of error. The accuracy of our models varies with the type of tweets searched. The accuracy is lower for harassing tweets than for neutral ones. The margin of error for accuracy is large given the need to integrate the resulting modification with the validation step into the deployment step. For instance, on the device, the bag-of-words set is only one list and it should be composed of at least five. In Fig. 4, the top arrow points to a tweet with the f-word that was not caught because the word has a different spelling. In Fig.4, the bottom arrow points to another harassing tweet that was not caught by our filtering system; the harassing phrase is of the format of the fourth set of bag-of-words that combines a subset of "bad action" and "recipient". The bad action is "kicking", the recipient is "him". Our aim is to reduce the size of our lexicon by permutating the bad action with different recipients. Even with some errors in detecting harassment, we obtained good results with real-time data. At first, our debugging output test results with real-time data had an accuracy of around 70%; once we trained our models with the new labeled data sets, the accuracy level increased to above 90%.

9 Conclusion

The System demonstrated that a supervised learning technique with hybrid classification and lexicon approaches obtains good results. Our solution was to design and implement an AI infrastructure to filter out harassment on real-time incoming tweets on an iPhone. The life cycle of the system allows us to adjust and retrain our text classifier models with unknown data. We have validated our labeled data because bad data will affect the output of the models. We trained Apple's Core ML3 and Google's Auto ML models; we obtained an accuracy of about 90% on the static data with both models using English and Italian data. For the deployment on the iPhone, we are using the Core ML 3 model on the Model Testing. We improved the quality of the training data with the lexicon adaptive filter. The Apple Core ML 3 documentation recommends that the text classifier is trained with at least one million data inputs to obtain the best results. The first version of the system used an ML model trained with an English language input of 78,533 tweets. The accuracy of the model was improved by increasing the number of inputs with which the model was trained. We expect that enlarging the training data with validated data will improve overall performance. The bag-of-words feedback loop improved the accuracy of the system on real-time data. We obtained an accuracy of 94% on real-time English Twitter data with a large margin of error. Our real-time data results were obtained with data unknown to our model, our bag-of-words, and our developing team.

References

Sagar Bhuta, Avit Doshi, Uehit Doshi, & M Narvekar. 2014. A Review of Techniques for Sentiment Analysis of Twitter Data. in *Issues and Challenges in Intelligent Computing Techniques (ICICT.)*

Hang Cui, Vidhu Mittal, Mayur DaterComparative. 2006. Experiments on Sentiment Classification for Online Product Reviews *AAAI* , vol. 6, pp. 1265-1270, 2006.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. *arXiv preprint arXiv:1703.04009.* https://arxiv.org/abs/1703.04009.

Jennifer Golbeck, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A. Geller, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, Kelly M. Hoffman, Jenny Hottle, Vichita Jienjitlert, Shivika Khare, Ryan Lau, Marianna J. Martindale, Shalmali Naik, Heather L. Nixon, Piyush Ramachandran, Kristine M. Rogers, Lisa Rogers, Meghna Sardana Sarin, Gaurav Shahane, Jayanee Thanki, Priyanka Vengataraman, Zijian Wan, and Derek Michael Wu. 2017. A Large Human-Labeled Corpus for Online Harassment Research. *In Proceedings of the 2017 ACM on web science conference*, pages 229–233. http://www.cs.umd.edu/~golbeck/papers/trolling.pdf.

Ismail Heba, Harous S., Belkhouche Boumediene. 2016. A Comparative Analysis of Machine Learning Classifiers for Twitter Sentiment Analysis. Research in Computing Science. 110. 71-83. 10.13053/rcs-110-1-6.

Vishal A. Kharde, Sheetal Sonawane. 2016. Sentiment Analysis of Twitter Data. A Survey of Techniques *International Journal of Computer Applications* (0975 – 8887), Volume 139 – No.11. https://www.researchgate.net/publication/301335561_Sentiment_Analysis_of_Twitter_Data_A_Survey_of_Techniques.

Olga Kolchyna, Tharsis, T.P. Souza, T. Philip Treleaven, Tomase Aste. 2015. Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination. Department of Computer Science, UCL, Gower Street, London, UK.

Walaa Medhat, Ahmed Hassan, and Hoda Korasshy. 2014. Sentiment analysis algorithms and applications: A survey *Ain Shams Engineering Journal*, Volume 5, Issue 4, pages 1093-1113. https://core.ac.uk/download/pdf/82415645.pdf.

Flavio Merenda, Claudia Zaghi, Tommaso Caselli, and Malvina Nissim. 2018. Source-driven Representations for Hate Speech Detection. In *CLiC-it*. https://core.ac.uk/download/pdf/213589615.pdf.

K. A. Ogudo and D. M. J. Nestor, Sentiment Analysis Application and Natural Language Processing for Mobile Network Operators' Support on Social Media, *2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, 2019, pp. 1-10, doi: 10.1109/ICABCD.2019.8851052.

Isidoros Perikos, Ioannis Hatzilygeroudis, 2016. Recognizing emotions in text using ensemble of classifiers, Engineering Applications of Artificial Intelligence, Volume 51, 10.1016/j.engappai.2016.01.012.

Azeema Sadia, Fariha Khan, and Fatima Bashir. 2018. An Overview of Lexicon-Based Approach For Sentiment Analysis. *3rd International Electrical Engineering Conference*, Karachi, Pakistan. https://ieec.neduet.edu.pk/2018/Papers_2018/15.pdf.

Gopalakrishnan Vinodhini, and Ramaswamy Chandrasekaran. 2013. Performance Evaluation of Machine Learning Classifiers in Sentiment Mining *International Journal of Computer Trends and Technology (IJCTT),* vol. 4, no. 6.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, June 2016. https://pdfs.semanticscholar.org/df70/4cca917666dace4e42b4d3a50f65597b8f06.pdf

Clare Wilson and Laura A Cariola. 2020. LGBTQI+ Youth and Mental Health: A Systematic Review of Qualitative Research. *Adolescent Research Review*, 5(2): pages 187–211. https://www.researchgate.net/publication/333301317_LGBTQI_Youth_and_Mental_Health_A_Systematic_Review_of_Qualitative_Research.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from Bullying Traces in Social Media. *In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 656–666. https://dl.acm.org/doi/pdf/10.5555/2382029.2382139.