Hate Speech and Offensive Language Detection with Bayesian Networks

Susana Benavidez
sbenavid@stanford.edu

Pratyusha Javangula
pjavan@stanford.edu

Andy Lapastora
awlapas@stanford.edu

Derek McCreight
dmccreig@stanford.edu

Abstract

With the proliferation of social media and anonymity of users, the identification of both offensive and hate speech is timely and important in creating spaces that promote the exchange of factual and socially appropriate information. To that end, we explored the application of Bayesian Networks to the task of fine-grained sentiment analysis of a set of tweets from Twitter. We found that Bayesian networks perform better than baseline linear classifiers, but worse than a Naive Bayes model in terms of recall for classification of hate speech. We conducted an in-depth error analysis to discover why Bayesian Networks performed this way.

1 Introduction

We proposed a novel hate-speech and offensive language detection system that used a Bayesian network to infer whether a given tweet is hate speech, offensive language, or neither. Specifically, we implemented a Bayesian network whose Directed Acyclic Graph **(DAG)** structure was learned with a score-based approach using a graph structure search technique called Chow-Liu. To perform classification, we applied a variational inference algorithm called loopy belief propagation to infer the posterior distribution over hate speech classification labels given evidence (in this case, tweet feature variables). Then, we selected the highest-posterior-probability label as the network-predicted classification. We chose this particular architecture because it lends itself well to an exploratory analysis of the interesting conditional dependence relationships between features, if they exist. Once identified, these relationships (or lack thereof) can be analyzed and leveraged to pinpoint the causes and symptoms of hate speech and offensive language.

2 Related Work

Lee et al. [2018] explored classifying abusive speech on Twitter using traditional Machine Learning methods (Naive Bayes, Logistic Regression, Support Vector Machine (SVM), etc.) as well as deep learning methods (Convolutional Neural Networks and Recurrent Neural Networks) as the first comparative study on the relatively substantive (100K samples) dataset created by Founta et al. [2018]. They classified according to labeling tweets as Normal, Abusive, Hateful, or Spam and found the best performing models were RNNs that integrated Latent Topic Clustering , achieving an F1 score of .805. They were also able to use the context of tweets as features but found that context did not improve the performance of the top models.

Davidson et al. [2017a] were the first to approach the task of offensive and hate speech labeling outside of a binary model by adding a third category of neither. Their focus was on classifying without relying on explicit hate keywords. The features they employed included uni/bi/trigrams

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Class label	Mean unigrams	Mean bigrams	Mean characters
Hate speech	15.89	14.89	4.34
Offensive language	16.11	15.11	4.04
Neither	17.45	16.45	4.43

Table 1: Text statistics in dataset.

Class	Number of samples
Hate speech	1430
Offensive language	19190
Neither	4163

Table 2: Classes and their count in dataset.

weighted by its TF-IDF, binary and count indicators for hashtags, mentions, retweets, and URLs. To capture syntactic structure information they used NLTK and Penn Part-of-Speech (POS) taggings. They trained a logistic regression with L2 regularization model on the entire dataset and then used it to predict the label for each tweet using a one-versus-rest framework with: 0.91 precision, 0.90 recall, and .90 F1 score. However, almost 40% of hate speech was misclassified: the precision and recall scores for the hate class were 0.44 and 0.61 respectively which suggested that the model is biased towards classifying tweets as less hateful or offensive than the human coders.

We hoped to exemplify the type of analysis done by Lee et al. [2018] on the dataset created by Davidson et al. [2017a] by including Bayesian Networks in the type of model used on the task of identifying and classifying offensive and hate speech on Twitter.

3 Methodology

3.1 Data

We used the 3-class dataset of tweets containing hate speech, offensive language, or neither compiled by Davidson et al. [2017a]. The data was obtained by first collecting a hate speech lexicon comprised of words and phrases from *Hatebase.org*, Tuckwood [2017] (Hatebase), an online, crowd-sourced repository of structured, multilingual, usage-based hate speech. Then, the Twitter API was used to obtain 85.4 million tweets from 33,548 users, of which 24,783 tweets were selected to make up the final dataset.

To obtain ground truth labels for this data, a crowd-sourcing website was used. Annotators were provided with a formal definition of hate speech and asked to label each tweet as hate speech, offensive but not hate speech, or neither offensive nor hate speech. Every tweet was labeled by at least three annotators, and mean inter-annotator agreement was 92%.

An example tweet labeled as "hate speech" is the following:

(1) Think it's okay to take my property and break it? F*** you b****

Now, consider the following example tweet labeled as "offensive but not hate speech":

(2) I'm bringing booty back. Go ahead and tell them skinny b****** that.

Finally, here is an example tweet that is neither hate speech nor offensive:

(3) This 8 yr old on #masterchefjunior made my chicken look like trash.

While the first two tweets are inappropriate, the first constitutes directed, intentional aggression towards a specific individual, whereas the second uses profanity to express a mostly innocuous point about body type desirability without implicating a marginalized group or specific individual. The third tweet, while implicating a child and using the word "trash", is not offensive or hateful, as the item that is "trash" is chicken rather than a person or group.

3.2 Pre-processing and Data Statistics

Along with the entire text of the tweet itself, each sample in Davidson et al. [2017b] came with a tweet ID, the number of raters who rated the tweet, the number of raters who rated the tweet in each class, and the class label. Tweets were tokenized using `nltk`'s *TweetTokenizer*, allowing tweet-specific constructions like emoji to be treated as individual tokens.

We did not remove punctuation from the tweets, as punctuation in tweets is used more liberally than in textual prose, and is often an indicator of strong emotion on the part of the author. Similarly, we did not lowercase the tweets, as capitalization is also often used to as a written proxy to prosody. Prosody is defined as the patterns of stress and inflection of spoken language, and has been shown to have an impact on speaker meaning outside of pragmatics and compositional lexical semantics Carlson [2009]. Finally, we did not remove stop words, since most tweets did not contain very many stop words.

Table 1 presents the mean number of unigrams per (single tokens) per sample, the mean number of bigrams (pairs of tokens) per sample, and the mean number of characters per token, for each of the three classes in the dataset. Table 2 gives the number of tweets in the dataset of each class. In the next section, we discuss how we extracted features from the data to serve as nodes in the Bayesian network.

3.3 Features

We opted to employ a variety of linguistic features which take into account the complexity of the language commonly used in tweets. We implemented 33 hand-built features.

We included a number of indicator features: if a tweet is a retweet, the tweet contains a mention, the tweet contains a Hatebase word, the tweet contains a hashtag, the tweet has no positive words, the tweet contains a word that has nonalphanumeric characters contained within it, the tweet contains profanity, the tweet contains a URL, and the tweet contains racist language. For features that would otherwise require a count, we discretized by comparing the count of the feature in a tweet to the quartile values of that feature count over the entire dataset. We used this approach for the number of tokens, swear word count, emoji count, misspelling count, mention count, and judgment disagreement count in a tweet. Judgment disagreement is when the original classifiers of the dataset disagreed on the label that should be assigned to a tweet. To catalog profane and racist words we leveraged the von Ahn [2009] Offensive/Profane Word List of over 1300+ English words that might potentially be seen as offensive.

Using the lexicon of Hatebase words, we were able to identify whether a tweet contained a word or words targeting specific categories which include gender, class, race, ethnicity, disability, religion, and sexual orientation. These were all included as binary features. We utilized Hatebase in addition to the von Ahn lexicon to give us more breadth in our definition of hateful and offensive words.

While the Hatebase and von Ahn [2009] lexicons were useful in identifying hateful words, clearly hateful tweets may not contain vulgarity at all but still direct hate and abuse towards a particular user or group of people. To try and account for this we included a number of features that tried to capture the nuance of these types of tweets, including sentiment and lexical features. By including these features, we hoped to capture the context surrounding the potentially hateful or offensive sentiment of these sorts of tweets since they do not have explicitly offensive terms.

We utilized the Novak et al. [2015] emoji sentiment dictionary to identify the sentiment of emojis in tweets. We also used the Hu and Liu [2004] sentiment lexicon to find the overall sentiment of each tweet, which we measured by raw count of positive versus negative words. Additionally, we included several lexical features including counting the number of uppercase letters to see if they outnumbered lowercase letters to try and capture "yelling" in text, if the tweet had multiple punctuation marks in a row, the number of misspellings, if the tweet started with a second or third person pronoun, and if the tweet had more second or third person pronouns than first.

3.4 System Architecture

In this section, we describe the algorithm that we utilized in order to learn the optimal structure of a Bayesian network containing nodes corresponding to the features we extracted from each of the tweets. In addition, we describe the inference algorithm we used in order to infer a posterior distribution over class labels, given all other nodes' values, $\mathbb{P}(Label|X_1 = x_1, X_2 = x_2, ...X_n = x_n)$. We classified tweets by inferring the posterior distribution over labels for each featurized tweet and selecting the highest-probability label from the resulting distribution.

3.4.1 Structure Learning of Bayesian Network

Because we lack domain expertise, we decided to learn the structure of the Bayesian network from data using a score-based approach. The specific score we selected was minimum description length (MDL). It is the default structure score given by `pomegranate`, a probabilistic modeling package in Python Schreiber [2017]. MDL is given by the following expression:

$$MDL(G|D) = LL(D|G)\frac{\log N}{2}|G| \tag{1}$$

For the present work, D corresponds to the dataset of tweets, N corresponds to the number of tweets in the dataset, $|G|$ is the number of nodes (i.e. tweet features), and G is the proposed graph structure. $LL(D|G)$ is a term for the log-likelihood of the dataset given the graph structure. Then, the optimal graph structure $G*$ that we will use for inference and subsequent classification will be found according to the following expression:

$$G* = argmax_G MDL(G|D) \tag{2}$$

The second component of a score-based structure learning algorithm is an efficient search over possible graph structures, given a set of nodes. We learned the structure of the hate speech detection Bayesian Network using the **Chow-Liu** algorithm, which searches for a maximum-weight spanning tree (i.e. a graph wherein there exists a path between any two feature variables in the final graph structure). In the next section, we discuss the inference algorithm that we utilized in order to infer a posterior distribution over classification labels.

3.4.2 Inference to Classify Tweets

We used loopy belief propagation to infer the posterior distribution over the hate speech classification labels, provided by the `pomegranate` package Schreiber [2017].

Loopy belief propagation is a variant of the belief propagation algorithm designed by Pearl (1988). Belief propagation computes the marginal probability distribution $\mathbb{P}(x_t)$ for all nodes x_t in a probabilistic graphical model (in the present work, a Bayesian network). The algorithm takes the form of a message-passing procedure between nodes. Specifically, at each iteration i of belief propagation, each node t's outgoing message to each outgoing neighbor s is updated as a function of the incoming messages from each incoming neighbor u on the previous iteration $i-1$. Convergence is reached when the outgoing message from each node t to its outgoing neighbors s is the same from iteration i to iteration $i+1$, or when $|m_{ts}^i(x_s) - m_{ts}^{i+1}(x_s)| < \lambda$, for some threshold hyperparameter λ.

Loopy belief propagation differs from belief propagation in that it can be applied to arbitrary graphical models. Although there is some possibility that this algorithm will not converge after a fixed set of iterations, in practice it is often the case that the algorithm reasonably approximates marginal distributions, conditional on observed evidence. The current research derives a model-predicted hate speech classification label by selecting the label with the highest-probability in the resulting posterior distribution.

4 Final Results

For our baseline models, we implemented a Logistic Regression (LR), Support Vector Machine (SVM), and a Naive Bayes (NB) model with word count, swear word count, whether or not the tweet has a mention and whether or not the tweet is a retweet as baseline features. We ran 10-fold cross validation on a combination of the training and development set. We focused on recall on the hate speech class as the primary metric of interest, as we wanted to see how well models correctly

identified the examples of hate speech in the data set. For the Bayesian network, we learned structure and parameters using the Chow-Liu approach on 90% of the training and validation sets combined, and performed inference on the remaining 10%. We did not perform K-fold cross validation on the Bayesian Network. After training the baseline models and the Bayesian network, we ran all models on a held-out test set of 2500 tweets. The results of running all models on the test set can be found in Tables 3, 4, and 5, and 6.

Table 3: Logistic Regression Baseline

	Precision	Recall	F1-score
Hate Speech	0.44	0.13	0.2
Offensive	0.90	0.93	0.91
Neither	0.67	0.71	0.69
Accuracy			0.85
Macro avg	0.67	0.59	0.60
Weighted avg	0.83	0.85	0.84

Table 4: SVM Baseline

	Precision	Recall	F1-score
Hate Speech	0.00	0.00	0.00
Offensive	0.90	0.93	0.91
Neither	0.65	0.79	0.72
Accuracy			0.85
Macro avg	0.52	0.57	0.54
Weighted avg	0.81	0.51	0.83

Table 5: NB Baseline

	Precision	Recall	F1-score
Hate Speech	0.17	0.70	0.27
Offensive	0.96	0.72	0.82
Neither	0.63	0.67	0.65
Accuracy			0.71
Macro avg	0.58	0.70	0.58
Weighted avg	0.86	0.71	0.76

Table 6: Bayesian Network Performance

	Precision	Recall	F1-score
Hate Speech	0.47	0.20	0.28
Offensive	0.89	0.92	0.90
Neither	0.65	0.67	0.66
Accuracy			0.84
Macro avg	0.67	0.60	0.62
Weighted avg	0.82	0.84	0.83

The fact that Naive Bayes by far outperformed the Bayesian Network seems to indicate that our assumption that conditional dependencies would exist in the feature set we created was not well founded. Naive Bayes assumes conditional independence of all features, and this assumption underlies the best performing model.

5 Error Analysis

Our bayesian network model had 402 classification differences where the prediction of our model did not match the true label of the tweet, as labeled by the Davidson and crowdflower team. 39% of these tweets were in the true label class offensive, 33% in true label class neither, and the remaining 28% in the hate speech class.

We believe that **our model failed in 59% of these classifications but disagree with the true label of 41% of missed tweets**. Of the 41%, we posit that $< 4\%$ of the 402 tweets were incorrectly labeled and incorrectly predicted by our model and that 96% of incorrectly labeled were correctly predicted by our model. This makes it difficult to gauge how well our model truly did as we are constrained by a dataset that is unbalanced and with which we disagree on the labeling accuracy.

We took a deep dive into the labeling disagreements and detail below common patterns and explanations as to why a prediction failed. In addition we discuss the problem with subjectivity in the labeling process and list the tweets where we disagree with the true label and agree with our predicted label.

5.1 True Label: Hate Speech

Our model has a recall of 20% for hate speech. This compares to our 33% recall in human performance used for our oracle.

Of the 113 instances of hate speech in missed tweets, we agree with the labeling of 59 tweets, meaning we disagree with almost half of all tweets labeled as hate speech. Of these, 100% of incorrectly labeled tweets were correctly predicted by our model.

- Of the 48% of mislabeled tweets, our model correctly predicted 11% of them into the neither class. These tweets were self-referential, in-group statements, and contained quotes around the offending term to make commentary against it.
  - In-group statement: 'Giving you that trailer park trash. transformthursday ladykimora vegasqueens vegasshowgirls http://t.co/5BbMhApHrD'
  - In-group statement and quotes around the offending term: 'RT @SnowThaProduct: i can do somethin good but im always gon be a "wetback" coo.. 9996;'
- The remaining 89% of the tweets were correctly predicted as offensive. Of these, several instances contained quotes around the offending term / statement and was used to reference it as a disbelief or commentary against it.
  - Expressing shock of offending statement in quotes: '"Let's kill cracker babies!". WTF did I just hear???????? WOW.'

Of the 52% of tweets that our model missed, the following groups were targeted:

- 33% African Americans; of these it is concerning that our model fails to detect them as hate speech as 84% of these tweets contained some variation of the full spelling of ni**er. We could fine tune our feature that detects the full spelling of offending terms and treats them differently than shorter spellings of the term (making ni**er more offensive than nigguh) to bias them towards hate speech. Some of the tweets are on the borderline of hate speech and offensive given that it is difficult to tell if they are spewing hate speech or pointing it out, such as:
  - '8220;@iamkrause: No need to thank me, killing niggers is it's own reward.8221; Dis my pal from our graduation when we was 26'

  which contains the offending statement in quotes. Given the extremely hateful content, we decided to err on the conservative side and agree with the hate speech label over our prediction of offensive.
- 4% Chinese; distressingly, every tweet contained a variation of the term 'chink' which highlights the need to identify words that have dual meanings. Implementing word embeddings could help contextualize these terms. Since we were limited to discrete features for our model, we could make a feature that identifies the usage of the term and checks for swear words.

- 11% Female with one instance threatening sexual violence. Many of the tweets would be difficult to identify due to their general hate statement but some of the tweets that targeted women contained a mention, so combining a feature that checks if it offends women + contains mention could help identify these tweets.
- 26% Gay community with two instances threatening sexual violence. We could implement a feature that checks for homophobic terms and checks for a mention to properly identify these tweets as hate speech.
- 19% White people, with two instances also including political hate speech. Similar to terms used for the Chinese community, terms with dual meanings were used to target white people, such as 'crackers'. There is a need to identify 'white trash' as one term so that all features that we run our data through provides an analysis on that term and not 'white' and 'trash' separately.
- $< 4\%$ targeting Latinos, $< 4\%$ Jews, and the remaining being instances of general racism and targeting Muslims. It is concerning that our model did not identify the tweets targeting Latinos and Jews as hateful as they all contained negative terms. There is a need to identify 'half breed' and it's variations as one term. But both tweets contain other hateful terms, making it hard to understand why the model labeled them incorrectly:
  - 'Spics are half breed trash. No filthy native should be allowed to speak to any European.'
  - 'Why do so many filthy wetback half-breed spic savages live in LosAngeles? None of them have any right at all to be here.'

  Similarly, the tweets targeting Jews were startingly hateful:
  - Note one of the instances referringto Jews is masked in 'ovenjew': '@InTheOvenJews @SlaveCatcher88 @marylene58 we should hang out ovenjew. Have a lot in common. People waste time on nigs. Jews r r problem.'
  - A feature identifying white nationalist slogans could help identify hate speech such as: '@MenachemDreyfus @NatlFascist88 @waspnse Shit your ass and shabbat your moms pussy u Jew bastard. Ur times coming. Heil Hitler!'

5.1.1 Model Prediction: Neither

$.04\%$ of tweets with true labels of hate speech were predicted to be neither by our model. We agree that our model failed in 38% of these tweets. We disagree with the true label for 33% of the tweets where we firmly believe that our model appropriately classified them as neither.

We missed two tweets targeting the gay community. In both instances, this seems to be a result of parsing. We failed to decipher the use of 'faggotsgt;' and retards 'retardsgt;' as the hash tag and the right tag symbol were used to disguise the words.

It should be noted that the tweets that we believe we correctly classified as neither all depended on in-group context such as 'This nigglet 10084;65039;', 'RT @FAAMMoverALL: This nigguh Chris Paul', etc. showing how our model's context features were able to correctly identify the in-group statements as neither offensive nor hate speech.

5.1.2 Model Prediction: Offensive

23% of true label hate speech tweets were labeled as offensive by our model. Of the 23% we believe that 54% of the tweets' true label was incorrectly labeled as hate speech and our model correctly tagged it as offensive. Thus our model failed to recall 46% of hate speech tweets.

The missed tweets targeting African Americans all included some form of 'nigg*r'. The high usage of the word in in-groups makes it difficult to distinguish as a hate word.

5.2 True Label: Offensive Speech

39% of tweets with true label offensive were labeled differently by our model.

5.2.1 Model Prediction: Neither

45% of tweets in the class offensive were correctly predicted by our model as neither. These tweets were self-referential:

- '8220;@MarieLeVisual: if your wardrobe consist of mostly leggings, you's a hoe8221; guilty'

Or quoting a song:

- '"This is for my ghetto motherfuckers" -Missy Elliott'

Or in quotes as commentary on the offensive speech:

- '"captain save a hoe" is the most annoying phrase'

Of the 55 tweets that our model incorrectly labeled as neither, 60% targeted women, showing a bias of our model predicting offensive speech as neither when concerning women.

5.2.2 Model Prediction: Hate Speech

Of the offensive class, our model predicted hate speech for 20% of these tweets. There are subjective to interpretation as the usage of the term 'faggot' to emasculate a man was present in every tweet that we predicted hate speech for but we erred on the side of the label.

We believe 48% of the tweets in the offensive class are incorrectly labeled with 32% of those correctly predicted by our model as hate speech and the remaining 16% with actual label of neither.

Tweets we believe should be hate speech include:

- '@Justi_Baez shut the fuck up faggot ass Knicks cock sucker your opinion is irrelevant fag'
  This tweet is targeted to a user and the use of the term 'faggot' and 'cock sucker' is meant to emasculate and humiliate.
- '@MANIAC3X buck beaners amp; security started pushing him amp; that jetsgreen faggot around'
  We think 'buck beaners' is a misspelling of 'fuck beaners,' which should is considered hateful in any context.

5.3 True Label: Neither Hate or Offensive Speech

33% of tweets with true label 'neither hate speech nor offensive' were classified differently by our model. We agree that 83% of the neither class was misclassified by our model but disagree with the true label of 17% of the neither class where we believe our model correctly labeled the tweets as offensive.

5.3.1 Prediction: Offensive

Highlighting the bias our model has toward the offensive (majority) label, 99% of mislabeled neither tweets were classified as offensive. For all tweets, the prediction makes sense as they contained words like "kill", "coconut" (a term that also has a derogatory term meaning brown on the outside, white on the inside), "bitch", etc. An example of an instance that is difficult to understand is the tweet '@louiev_ lol just getting this..early bird catches the worm!' which was classified as offensive. Five misclassified tweets had some variation of 'early bird catches the worm!'.

We believe our model was able to correctly label tweets as offensive (although these could also be considered hate speech), including examples like:

- 'RT @JohnnyFootballI: Yeah Kaepernick might have biceps like a Greek God, but the dude looks like he was conceived by a Proboscis monkey http8230;'
  The model correctly picks up on monkey being used as an insult to Kaepernick
- ' @mis_sarahd @basedpapi1017 tranny '
  The word 'tranny' is offensive in almost any context;

- 'Is dare any colored players in hockey?'
  The model identifies the phrase 'colored players' as offensive.

5.3.2 Prediction: Hate Speech

Our model had one instance where it classified a tweet as hate speech when its true label was neither:

- 'I didnt buy any alcohol this weekend, and only bought 20 fags. Proud that I still have 40 quid tbh'

Fag is a slang term for cigarettes in the UK. The model missed the contextual information provided by 'quid' being a non-American term and the alternate meaning of the word. Even with a more diverse dataset, we would be limited by using sentiment analysis derived from an American English lexicon. Interestingly our model failed to label a tweet with the word 'faggot' as hate speech, exemplifying how a word used in hate speech, offensive, and neither classes can down-weigh how offensive the term is.

5.4 Novel Insights and Future work in Error Analysis

While our model was not the best performing in hate speech recall, we stress the importance of labeled datasets validated by a third party expert in the field of hate speech. The level of disagreements between the labels and our belief of what the labels should be made it it difficult to accurately aseess the performance of our features.

We plan to continue our work in this space and our project as we have not found any previous work trying to find conditional dependencies in hate speech and to provide a deep dive into the subjectivity of, and defining characteristics of, hate speech and offensive language.

We have applied and just been granted academic research API access from Twitter and will work on creating new datasets with more identifying information (including only using active Twitter accounts as many of the Twitter handles used in Davidson et al. [2017a] have been suspended, limiting contextual information).

6 Future Directions and Conclusion

We were primarily interested in discovering how well a Bayesian Network would perform on the task of multi-class hate speech detection, and we found that it did not perform very well when compared to our Naive Bayes baseline model. However, it did outperform Logistic Regression and SVM on several metrics, including recall for Hate Speech, which was our primary metric of interest. It is possible that the features that we implemented are not the ones that best model the dependencies that would best predict hate speech. We were limited to discrete features, ruling out word embeddings which may have provided the contextual information needed in tweets with non-explicit hate words.

Another limitation is the data. Davidson et al. [2017a] point out their own disagreements with how the data was labeled (i.e. song lyrics incorrectly identified as hate speech). Further work should incorporate expert opinion when deciding the labels of tweets. Further, the data set was heavily biased toward offensive speech. A more balanced distribution of labels would likely improve the performance of a classifier.

We believe it would be valuable to use an algorithm called **greedy** structure learning developed by Schreiber [2017] to learn the structure of the DAG. This algorithm greedily chooses a topological ordering of the variables for our tree structure. Then, the algorithm optimally identifies the best parents for each variable given this ordering. The Chow-Liu approach makes the simplifying assumption that there exists some path between all nodes in the tree structure, hence vastly reducing the set of possible DAG structures to explore. The greedy approach proposed here does not in fact make this assumption and instead searches through the set of all possible DAGs including DAGs featuring islands of nodes, that is to say, graphs which are not fully connected. Such an approach may reveal a more explainable graphical structure in terms of how features relate to one another.

Additionally, the ability to leverage the Twitter API to incorporate more contextual features about users such as when a user posted a tweet, or the user's gender would be extremely valuable. It's possible that contextual features like these would reveal a much more dependent structure when analyzed with a Bayesian Network.

Link to our github repo: https://github.com/pratyushaj/abusive-language-online

References

Younghun Lee, Seunghyun Yoon, and Kyomin Jung. Comparative studies of detecting abusive language on twitter. *CoRR*, abs/1808.10245, 2018. URL http://arxiv.org/abs/1808.10245.

Antigoni-Maria Founta et al. Large scale crowdsourcing and characterization of twitter abusive behavior. *CoRR*, 2018. URL http://arxiv.org/abs/1802.00393.

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

Christopher Tuckwood. Hatebase: Online database of hate speech. *The Sentinal Project. Available at: https://www. hatebase. org*, 2017.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language, 2017b. URL https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15665.

Katy Carlson. How prosody influences sentence comprehension. *Language and Linguistics Compass*, 3(5):1188–1200, 2009. doi: 10.1111/j.1749-818X.2009.00150.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1749-818X.2009.00150.x.

Luis von Ahn, 2009. URL https://www.cs.cmu.edu/~biglou/resources/bad-words.txt.

Petra Kralj Novak, Jasmina Smailovic, Borut Sluban, and Igor Mozetic. Sentiment of emojis. *CoRR*, abs/1509.07761, 2015. URL http://arxiv.org/abs/1509.07761.

Minqing Hu and Bing Liu. Mining and summarizing customer comments. KDD, 2004.

Jacob Schreiber. Pomegranate: Fast and flexible probabilistic modeling in python. *J. Mach. Learn. Res.*, 18(1):5992–5997, January 2017. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=3122009.3242021.