Co-trained Ensemble Models for Weakly Supervised Cyberbullying Detection

Elaheh Raisi
Department of Computer Science
Virginia Tech
elaheh@vt.edu

Bert Huang
Department of Computer Science
Virginia Tech
bhuang@vt.edu

Abstract

Social media has become an inevitable part of individuals' social and business lives. Its benefits come with various negative consequences. One major concern is the prevalence of detrimental online behavior on social media, such as online harassment and cyberbullying. In this study, we aim to address the computational challenges associated with harassment detection in social media by developing a machine-learning framework with three distinguishing characteristics. (1) It uses minimal supervision in the form of expert-provided key phrases that are indicative of bullying or non-bullying. (2) It detects harassment with an ensemble of two learners that co-train one another; One learner examines the language content in the message, and the other learner considers the social structure. (3) It incorporates distributed word and graph-node representations by training nonlinear deep models. The model is trained by optimizing an objective function that balances a co-training loss with a weak-supervision loss. We evaluate the effectiveness of our approach using post-hoc, crowdsourced annotation of Twitter data, finding that our deep ensembles outperform previous non-deep methods for weakly supervised harassment detection. We also evaluate on a new benchmark to measure the sensitivity of the detectors to language describing particular social groups.

1 Introduction

The advent of social media has revolutionized human communication. Social media owes its increasing popularity to its uncountable positive influences on individuals' social and business lives. It makes people closer to each other, provides access to enormous real-time information, and eases marketing and business. Despite these benefits, social media has amplified some detrimental aspects of society. Online harassment and cyberbullying are among the major adverse consequences of social media's popularity. According to the American Academy of Child and Adolescent Psychiatry [1], victims of bullying can be suffer interference to social and emotional development and even be drawn to extreme behavior such as attempted suicide. Any widespread bullying enabled by technology represents a serious social health threat.

In this paper, we consider a machine-learning approach for harassment-based cyberbullying detection. We approach to the cyberbullying detection problem from different angle than many machine-learning algorithms proposed thus far. Most machine learning methods for this problem consider supervised text-based cyberbullying detection, classifying social media posts as "bullying" or "non-bullying." In these approaches, crowdsource workers annotate the data, and then a supervised classifier is applied to classify the posts. There are, however several challenges related to these approaches. Fully annotating data requires human intervention, which is costly and time consuming. And without considering social context, differentiating bullying from less harmful behavior is difficult due to complexities underlying cyberbullying and related behavior. Our approach aims to encode such complexities into an efficiently learnable model.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

We use machine learning with weak supervision, which significantly alleviates the need for human experts to perform tedious data annotation. Our weak supervision is in the form of expert-provided key phrases that are highly indicative of bullying. We refer to our proposed framework as the *co-trained ensemble* method, which trains two detectors to extrapolate from the weak supervision to form a rich, multi-faceted model. One detector identifies bullying by examining the language content in messages; another detector considers the social structure to detect bullying. Each detector is using different body of information, and the individual detectors co-train one another to come to an agreement about the bullying concept. They seek consensus on whether examples in unlabeled data are cases of cyberbullying or not.

We represent the language and users as vectors of real numbers with embedding models. For example, word2vec [15, 16] is a popular word-embedding model that represents words with low-dimensional vectors. And node2vec [6] is a framework for learning continuous feature representations for nodes in networks. We use word and user vectors as the input to language-based and user-based classifiers, respectively. We examine two strategies when incorporating vector representations of words and users. First, using existing doc2vec [13]—an extension of word embedding—models as inputs to the learners. Second, creating new embedding models specifically geared for our specific task of harassment detection, which we train in an end-to-end manner during optimization of the model, incorporating the unsupervised doc2vec and node2vec loss function into our co-training objective.

To train the model, we construct an optimization problem made up of a regularizer and two sets of loss functions: a co-training loss that penalizes the disagreement between the deep language-based model and the deep user-based model, and a weak-supervision loss that is the classification loss on weakly labeled messages.

We evaluate our approach on Twitter data, which is one of the public-facing social media platforms with the most frequency of cyberbullying. We use a human-curated list of key phrases indicative of bullying as the weak supervision, and assess the precision of detections by variations of the framework. We evaluate the effectiveness of our approach using post-hoc, crowdsourced annotation of Twitter. We quantitatively demonstrate that our weakly supervised deep models improves precision over a non-deep variation of the model. In addition, we measure how biased and discriminative the proposed algorithm is against particular targeted groups including but not limited to race, gender, religion, and sexual orientations. Our experiments show that our proposed framework—which combines of weak supervision, co-training, and deep, nonlinear detectors—performs better than a model lacks any one of these three characteristics.

2 Related Work

Many researchers have proposed computational methods for automated online harassment and cyberbullying detection. Most methods developed so far use supervised classification algorithms to classify messages as "bullying" or "non-bullying" by extracting language features. Some proposed gender-specific language features to classify users into male and female groups to improve the discrimination capacity of a classifier for cyberbullying detection [4]. Others applied a lexical syntactic feature (LSF) [3] approach to detect offensive content in social media and users who send offensive messages. Others focused on detecting of textual cyberbullying in YouTube comments by manually labeling 4,500 YouTube comments and applying binary and multi-class classifiers [5]. Another approach used the number, density and the value of offensive words as features for cyberbullying identification on the Formspring service [21]. There have been many contributions in designing special features: using features learned by topic models as well as curse words weighted by TF-IDF [17], using sentiment features [24], applying vulgar language expansion using string similarity [19], extracting features based on association rule techniques [14], and using static, social structure features [12]. Additionally, some studies have involved firsthand accounts of young persons, yielding insights on new features for bullying detection and strategies for mitigation [2]. Hosseinmardi et al. [7, 8, 9, 10] conducted several studies analyzing cyberbullying on different online platforms, with findings that highlight cultural differences among the platforms.

Our work directly builds off a recent paper that introduced the *participant-vocabulary consistency* (PVC) method [20], which uses a similar paradigm of viewing the learning tasks as seeking consensus between language-based and user-based perspectives of the problem. PVC uses simple key-phrase presence and a two-parameter user characterization, scoring how much a user tends to bully and how much they tend to be victimized, as its vocabulary and participant detectors, respectively. Our approaches replaces these with richer classifiers.

Recent reactions to a Google Jigsaw-released tool for quantifying toxicity of online conversations (see e.g., [22]) have highlighted an important aspect of any automated harassment or bullying detection: fairness, especially in the context of false positives. A serious concern of these detectors is how differently they flag language used by or about particular groups of people. We begin to address this issue with a benchmark analysis in our experiments.

3 Co-Trained Ensembles

Our learning framework uses co-trained ensembles of weakly supervised detectors. In this section, we first describe them generally, then we describe the specific instantiations we use in our experiments. For social media data, we consider a set of users U and a set of messages M. Each message $m \in M$ is sent from user $s(m)$ to user $r(m)$. In other words, the functions s and r return the sender and receiver, respectively, of their input message. The input data takes on this form, with some of the messages annotated with weak supervision.

General Framework We define two types of classifiers for harassment detection: message classifiers and user-relationship classifiers (or user classifiers for short). Message classifiers take a single message as input and output a classification score for whether the message is an example of harassment, i.e., $f : M \mapsto \mathbb{R}$. User classifiers take an ordered pair of users as input and output a score indicating whether one user is harassing the other user, i.e., $g : U^2 \mapsto \mathbb{R}$. For message classifiers, our framework accommodates a generalized form of weakly supervised loss function ℓ (which could be straightforwardly extended to also allow full or partial supervision). Let Θ be the model parameters for the combined ensemble of both classifiers. The training objective is

$$\min_{\Theta} \underbrace{\frac{1}{2|M|} \sum_{m \in M} \left(f(m;\Theta) - g\left(s(m), t(m); \Theta\right)\right)^2}_{\text{consistency loss}} + \underbrace{\frac{1}{|M|} \sum_{m \in M} \ell\left(f(m;\Theta)\right),}_{\text{weak supervision loss}} \tag{1}$$

where the first loss function is a consistency loss, and the second loss function is the weak supervision loss. The consistency loss penalizes the disagreement between the scores output by the message classifier for each message and the user classifier for the sender and receiver of the message.

We experiment with different variations of this framework that arise from instantiating the weak supervision loss and different classification models for the user and message classifiers.

Key-Phrase Weak Supervision Loss Our weak supervision relies on annotated lists of key-phrases that are indicative or counter-indicative of harassment. For example, various swear words and slurs are common indicators of bullying, while positive-sentiment phrases such as "thanks" are counter-indicators. Let there be a set of indicator phrases and a set of counter-indicator phrases for harassment. Our weak supervision loss ℓ is based on the fraction of indicators and counter-indicators in each message, so for a message containing $n(m)$ total key-phrases, let $n^+(m)$ denote the number of indicator phrases in message m and $n^-(m)$ denote the number of counter-indicator phrases in the message. Our weak supervision loss is then

$$\ell(y_m) = -\log\left(\min\left\{1, 1 + (1 - \tfrac{n^-(m)}{n(m)}) - y_m\right\}\right) - \log\left(\min\left\{1, 1 + y_m - \tfrac{n^+(m)}{n(m)}\right\}\right). \tag{2}$$

Models For the message classifiers, we use a randomly hashed bag of n-grams (BoW) model with 1,000 hash functions [23], a linear classifier based on the pre-trained doc2vec vector of messages trained on our Twitter dataset [13], a custom-trained embedding model with each word represented with 100 dimensions (emb), and a recurrent neural network (LSTM) with 2 hidden layers of 100 dimensionality (RNN). The emb and RNN models are trained end-to-end to optimize our overall loss function, and the vector-based models (BoW, doc2vec) are trained to only adjust the linear classifier weights given the fixed vector representations for each message.

For user classifiers, we use a linear classifier on concatenated vector representations of the sender and receiver user nodes. For our first user classifier (vec), we pre-train a node2vec [6] representation of the communication graph, which uses an algorithm designed to find vector representations that organize nodes based on their network roles and communities they belong to. Our other user classifier (emb) directly trains vector embeddings of the nodes to optimize our objective function in an end-to-end manner.

4 Experiments

Mirroring the setup initially used to evaluate PVC [20], we construct our weak supervision signal by collecting a dictionary of 3,461 offensive key-phrases (unigrams and bigrams) [18]. We augment this with a list of positive opinion words in [11]. The offensive phrases are our weak indicators and the positive words are our counter-indicators.

We use the data collected by Raisi & Huang [20]. They collected data from Twitter's public API, extracting tweets containing offensive-language words posted between November 1, 2015, and December 14, 2015. They then extracted conversations and reply chains that included these tweets. They then used snowball sampling to gather tweets in a wide range of topics. After some preprocessing, the Twitter data contains 180,355 users and 296,308 tweets.

4.1 Precision Analysis

We use post-hoc human annotation to measure how well the outputs of the algorithms agree with annotator opinions about bullying. We asked crowdsourcing workers from Amazon Mechanical Turk to evaluate the discovery of cyberbullying interactions of all methods. First, we average the user and message classification score of each message. Then, we extract the 100 messages most indicated to be bullying by each method. Finally, we collected the full set of messages sent between the sender and receiver of these messages. We showed the annotators the anonymized conversations and asked them, "Do you think either user 1 or user 2 is harassing the other?" The annotators indicated either "yes," "no," or "uncertain." We collected five annotations per conversation.

In Fig. 1, we plot the precision@k of the top 100 interactions for all the combinations of message and user detectors. We compare these methods with each other and against PVC [20]. The precision@k is the proportion of the top k interactions returned by each method that the majority of annotators agreed seemed like bullying. For each of the five annotators, we score a positive response as +1, a negative response as -1, and an uncertain response as 0. We sum these annotation scores for each interaction, and we consider the interaction to be harassment if the score is greater than or equal to 3.

Considering the BoW message detector, the best precision is when BoW message detector is combined with the node2vec user detector. Interestingly, BoW by itself does quite well; its precision is slightly lower than BoW with the node2vec user learner. The BoW message learner with the embedding user detector does not perform well. The embedding message detector, after interaction 20, with and without the user detector does about the same, but noticeably much better than PVC. The RNN message detector by itself has high precision, similarly to when combined with the node2vec user detector, which leads to a slightly lower precision. RNN when combined with the embedding user detector has the lowest precision, similar to PVC. The word2vec message detector when combined with the node2vec user learner has the best precision, better than the word2vec message detector alone. The precision of word2vec combined with the embedding user detector is worse than PVC.

To sum up, the BoW and word2vec message detectors when combined with the node2vec user detector had better precision than when combined with the embedding user detector and not having a user learner at all. The RNN message detector has the best precision without any user learner, slightly better than when combined with the node2vec user learner. The precision of the embedding message learner is almost the same for all user learners. For all of the methods, there is a significant improvement over PVC. It is worth mentioning that the embedding user learner has the best performance when combined with embedding message learners, otherwise its precision is poor.

4.2 Identity Statements

We create a corpus of sentences using the combination of some sensitive keywords describing different attributes: sexual orientation, race, gender, and religion. We generate statements of identity (e.g., "I am a black woman.") that are not harassment. An ideal, fair language-based detector should treat these keywords equitably, not estimating higher scores for any keyword over any other.

Score-Based Comparison Using different combination of message and user learners (12 methods in total), we computed the average score of sentences containing each keyword. Since none of these statements are in fact reasonable examples of harassment, the ideal score should be low; any high scores are false positives. The rnn_emb combination has the lowest score of 0.147. Next, rnn_node2vec (0.257) and emb_none (0.381) also have reasonable low scores. Methods that returned

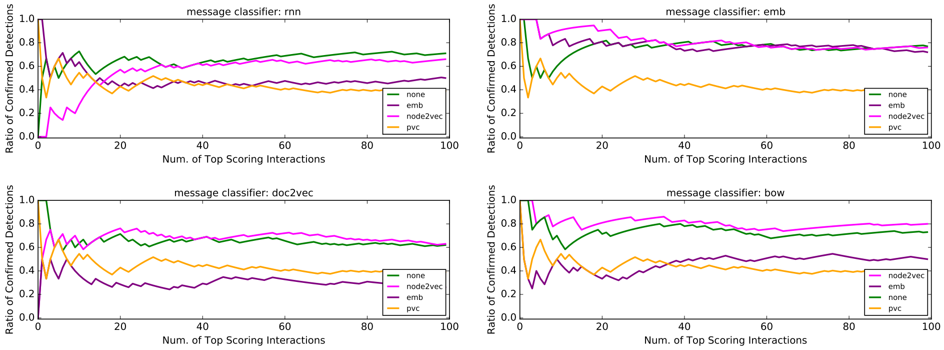

Figure 1: Precision@k for bullying interactions on Twitter using the combination of message and user learners, and PVC.

high average scores include bow_node2vec (0.536), bow_none (0.543), and emb_node2vec (0.588). The RNN message detector when combined with a user detector (node2vec or embedding) are among the least sensitive to these identity statements. The BoW message detector generally (with and without a user learner) and the embedding message detector combined with the node2vec user learner have higher scores. We believe these differences may illustrate the fact that the RNN model attempts to consider the sequence of language and its structure more than the orderless bag-of-words representation, but more study is necessary to make definitive conclusions.

Keyword Score Comparisons We compute the average score of sensitive keywords according to each method to find out which keywords have highest score (more falsely positive), and which keywords have the lowest score. We show the results for two methods: rnn_node2vec and rnn_none. In Fig. 2, three keywords "black," "boy," and "queer" have the highest score by both methods. The words "black" and "queer" were in our negative seed words, so this reveals a need to be more careful about which words are included in the weak supervision. One hypothesis why the word "boy" is among the highest-scored words is that this word co-occurs with many offensive words in the Twitter data. Generally, the score of all keywords returned by rnn_node2vec is lower than rnn_none, which suggests that the learning algorithm that co-trains using social structure helps to reduce the bias and sensitivity of the RNN message detector. The score of "woman" was not higher than the score of "man" for both methods, and the score of "girl" was much lower than the score of "boy," while in a model biased against women, we expect to observe the reverse behavior.

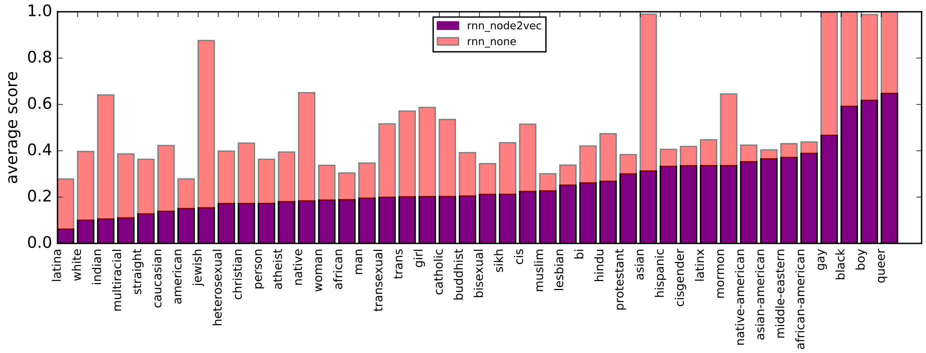

Figure 2: Average score of statements containing each keyword by rnn_node2vec and rnn_none.

5 Conclusion

We present a method for detecting online harassment using weak supervision. Harassment detection requires managing the time-varying nature of language, the difficulty of labeling the data, and complexity of understanding the social structure behind these behaviors. We developed a weakly supervised framework in which two learners train each other to form a consensus whether the social interaction is bullying by incorporating nonlinear embedding models. Our preliminary experiments show that co-training can help improve precision as well as produce more equitable models.

References

[1] American Academy of Child Adolescent Psychiatry. Facts for families guide. the American Academy of Child Adolescent Psychiatry. *http://www.aacap.org/AACAP/*, 2016.

[2] Z. Ashktorab and J. Vitak. Designing cyberbullying mitigation and prevention solutions through participatory design with teenagers. In *Proc. of the CHI Conf. on Human Factors in Computing Systems*, pages 3895–3905, 2016.

[3] Y. Chen, Y. Zhou, S. Zhu, and H. Xu. Detecting offensive language in social media to protect adolescent online safety. *Intl. Conf. on Social Computing*, pages 71–80, 2012.

[4] M. Dadvar, F. de Jong, R. Ordelman, and D. Trieschnigg. Improved cyberbullying detection using gender information. *Dutch-Belgian Information Retrieval Workshop*, pages 23–25, February 2012.

[5] K. Dinakar, R. Reichart, and H. Lieberman. Modeling the detection of textual cyberbullying. *ICWSM Workshop on Social Mobile Web*, 2011.

[6] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. *CoRR*, abs/1607.00653, 2016.

[7] H. Hosseinmardi, A. Ghasemianlangroodi, R. Han, Q. Lv, and S. Mishra. Towards understanding cyberbullying behavior in a semi-anonymous social network. *IEEE/ACM International Conf. on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 244–252, August 2014.

[8] H. Hosseinmardi, S. Li, Z. Yang, Q. Lv, R. I. Rafiq, R. Han, and S. Mishra. A comparison of common users across Instagram and Ask.fm to better understand cyberbullying. *IEEE Intl. Conf. on Big Data and Cloud Computing*, 2014.

[9] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra. Analyzing labeled cyberbullying incidents on the Instagram social network. In *Intl. Conf. on Social Informatics*, pages 49–66, 2015.

[10] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra. Detection of cyberbullying incidents on the Instagram social network. *Association for the Advancement of Artificial Intelligence*, 2015.

[11] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[12] Q. Huang and V. K. Singh. Cyber bullying detection using social and textual analysis. *Proceedings of the International Workshop on Socially-Aware Multimedia*, pages 3–6, 2014.

[13] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

[14] H. Margono, X. Yi, and G. K. Raikundalia. Mining Indonesian cyber bullying patterns in social networks. *Proc. of the Australasian Computer Science Conference*, 147, January 2014.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, 2013.

[17] V. Nahar, X. Li, and C. Pang. An effective approach for cyberbullying detection. *Communications in Information Science and Management Engineering*, 3(5):238–247, May 2013.

[18] noswearing.com. List of swear words & curse words. *http://www.noswearing.com/dictionary*, 2016.

[19] M. Ptaszynski, P. Dybala, T. Matsuba, F. Masui, R. Rzepka, and K. Araki. Machine learning and affect analysis against cyber-bullying. In *Linguistic and Cognitive Approaches to Dialog Agents Symposium*, pages 7–16, 2010.

[20] E. Raisi and B. Huang. Cyberbullying detection with weakly supervised machine learning. In *Proceedings of the IEEE/ACM International Conference on Social Networks Analysis and Mining*, 2017.

[21] K. Reynolds, A. Kontostathis, and L. Edwards. Using machine learning to detect cyberbullying. *Intl. Conf. on Machine Learning and Applications and Workshops (ICMLA)*, 2:241–244, 2011.

[22] C. Sinders. Toxicity and tone are not the same thing: analyzing the new Google API on toxicity, PerspectiveAPI. https://medium.com/@carolinesinders/toxicity-and-tone-are-not-the-same-thing-analyzing-the-new-google-api-on-toxicity-perspectiveapi-14abe4e728b3.

[23] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proc. of the Intl. Conf. on Machine Learning*, pages 1113–1120, 2009.

[24] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards. Detection of harassment on Web 2.0. *Content Analysis in the WEB 2.0*, 2009.