Leveraging Intra-User and Inter-User Representation Learning for Automated Hate Speech Detection

Jing Qian, Mai ElSherief, Elizabeth M. Belding, William Yang Wang
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106 USA
{jing_qian, mayelsherif, ebelding, william}@cs.ucsb.edu

Abstract

Hate speech detection is a critical, yet challenging problem in Natural Language Processing (NLP). Despite the existence of numerous studies dedicated to the development of NLP hate speech detection approaches, the accuracy is still poor. The central problem is that social media posts are short and noisy, and most existing hate speech detection solutions take each post as an isolated input instance, which is likely to yield high false positive and negative rates. In this paper, we radically improve automated hate speech detection by presenting a novel model that leverages intra-user and inter-user representation learning for robust hate speech detection on Twitter. In addition to the target Tweet, we collect and analyze the user's historical posts to model intra-user Tweet representations. To suppress the noise in a single Tweet, we also model the similar Tweets posted by all other users with reinforced inter-user representation learning techniques. Experimentally, we show that leveraging these two representations can significantly improve the f-score of a strong bidirectional LSTM baseline model by 10.1%.

1 Introduction

The rapid rise in user-generated web content has not only yielded a vast increase in information accessibility, but has also given individuals an easy platform on which to share their beliefs and to publicly communicate with others. Unfortunately, this has also led to nefarious uses of online spaces, for instance for the propagation of hate speech.

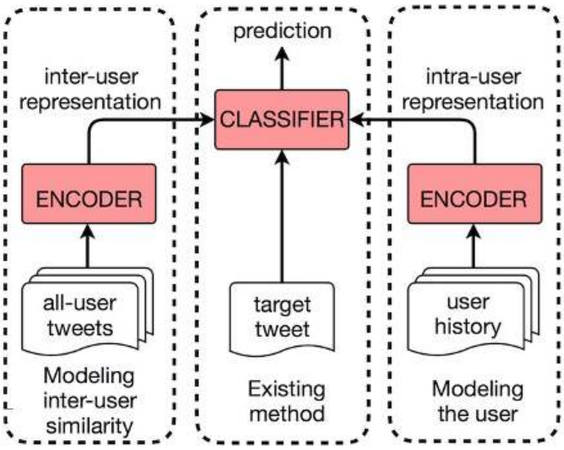

Figure 1: Our hate speech classifier. In contrast to existing methods that focus on a single target Tweet as input (center), we incorporate intra-user (right) and inter-user (left) representations to enhance performance.

An extensive body of work has focused on the development of automatic hate speech classifiers. A recent survey outlined eight categories of features used in hate speech detection (Schmidt and Wiegand, 2017): simple surface (Warner and Hirschberg, 2012; Waseem and Hovy, 2016), word generalization (Warner and Hirschberg, 2012; Zhong et al., 2016), sentiment analysis (Van Hee et al., 2015), lexical resources and linguistic features (Burnap and Williams, 2016), knowledge-based features (Dinakar et al., 2012), meta-information (Waseem and Hovy, 2016), and multi-modal information (Zhong et al., 2016). Closely related to our work is research that leverages user attributes in the classification process such as history of participation in hate speech and usage of profanity (Xiang et al., 2012; Dadvar et al., 2013). Both Xiang et al. (2012) and Dadvar et al. (2013) collect user history to enhance detection accuracy. The former requires the user history to be labeled instances. However, labeling user history requires significant human effort. The latter models the user with manually selected features. In contrast, our approach leverages unlabeled user history to automatically model the user.

In this paper, we focus on augmenting hate speech classification models by first performing

Proceedings of NAACL-HLT 2018, pages 118–123
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

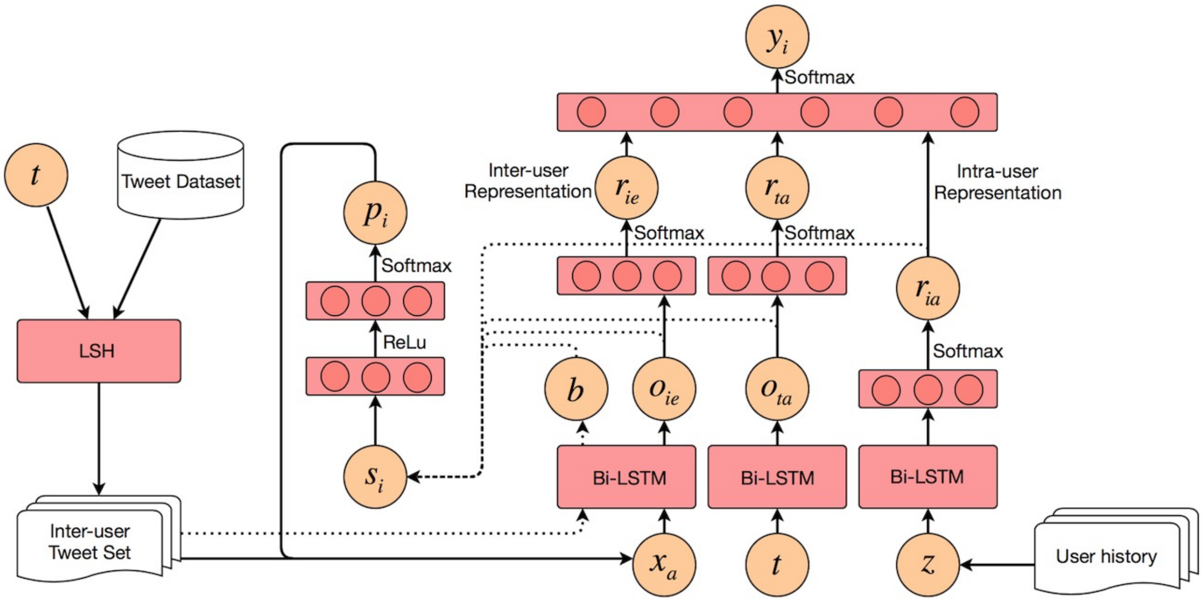

Figure 2: The overview of our proposed model. t is the input target Tweet, z denotes intra-user Tweets, and x_a is the selected inter-user Tweet. r_{ie} is the inter-user representation, r_{ia} is the intra-user representation, and r_{ta} is the representation of the target Tweet. These three branches respectively correspond to the three branches illustrated in Figure 1. y_i is the prediction at the time step i and s_i is the state input for the agent at the time step i. The computing process is detailed in Section 2.3

representation learning to model user history without supervision. The hypothesis is that, by analyzing a corpus of the user's past Tweets, our system will better understand the language and behavior of the user, leading to better hate speech detection accuracy. Another issue is that using a single Tweet as input is often noisy for any machine learning classifier. For example, the Tweet *"I'm not sexist but I can not stand women commentators"* is actually an instance of hate speech, even though the first half is misleading. To minimize noise, we also consider semantically similar Tweets posted by other users. To do so, we propose a reinforced bidirectional long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997) to interactively leverage the similar Tweets from a large Twitter dataset to enhance the performance of the hate speech classifier. An overview of our approach is shown in Figure 1. The main contributions of our work are:

- We provide a novel perspective on hate speech detection by modeling intra-user Tweet representations.
- To improve robustness, we leverage similar Tweets from a large unlabeled corpus with reinforced inter-user representations.
- We integrate target Tweets, intra-user and inter-user representations in a unified framework, outperforming strong baselines.

2 Approach

Figure 2 illustrates the architecture of our model. It includes three branches, whose details will be described in the following subsections.

2.1 Bidirectional LSTM

Given a target Tweet, the baseline approach is to feed the embeddings of the Tweet into a bidirectional LSTM network (Hochreiter and Schmidhuber, 1997; Zhou et al., 2016; Liu et al., 2016) to obtain the prediction. This is shown in the middle branch in Figure 1. However, this method is likely to fail when the target tweet is noisy or the critical words for making predictions are out of vocabulary.

2.2 Intra-User Representation

The baseline approach does not fully utilize available information, such as the user's historical Tweets. In our approach, we collect the user's historical posts through the Twitter API. For a target Tweet t, suppose we collect m Tweets posted by this user: $Z_t = \{z_1, z_2, ..., z_m\}$. These intra-user Tweets are fed into a pre-trained model to obtain an intra-user representation. The pre-trained model has the same structure as the baseline model. This is shown in the right branch in Figures 1 and 2. The intra-user representation is then combined with the baseline branch for the final prediction. The computation process is:

$$o_{ta}(t) = f_{ta}(t, \mathbf{0}) \tag{1}$$

$$r_{ta}(t) = l_{ta}(\sigma(o_{ta}(t))) \tag{2}$$

$$o_{ia}(z_j) = f_{ia}(z_j, \mathbf{0}) \tag{3}$$

$$r_{ia}(t) = \sigma(\sum_{j=1}^{m} l_{ia}(\sigma(o_{ia}(z_j)))) \tag{4}$$

where f_{ta} is the bi-LSTM of the baseline branch; o_{ta} is the output of the bi-LSTM; and l_{ta}, l_{ia} are linear functions. Similarly, f_{ia} is the bi-LSTM of the intra-user branch and o_{ia} is the output. r_{ta} is the output prediction of the baseline branch. r_{ia} is the intra-user representation, and σ is the non-linear activation function.

2.3 Inter-User Representation

In addition to the user history, the Tweets that are semantically similar to the target Tweet can also be utilized to suppress noise in the target Tweet. We collect similar Tweets from large unlabeled Tweet set U by Locality Sensitive Hashing (LSH) (Indyk and Motwani, 1998; Gionis et al., 1999). Since the space of all Tweets is enormous, we use LSH to efficiently reduce the search space. For each target Tweet t, we use LSH to collect n nearest neighbors of t in U: x_1, x_2, ..., x_n. These n Tweets form the inter-user Tweet set for t: $X_t = \{x_1, x_2, ..., x_n\}$.

Due to the size of this set, a policy gradient-based deep reinforcement learning agent is trained to interactively fetch inter-user Tweets from X_t. The policy network consists of two layers as shown in the middle part of Figure 2 and the policy network is trained by the REINFORCE algorithm (Williams, 1992). At each time step i, the action of the agent is to select one Tweet x_a from X_t. x_a is then fed into a bi-LSTM followed by a linear layer. The result is combined with the intra-user representation and the baseline prediction (the right and the middle branch in Figures 1 and 2) to get the prediction at time step i. At each time step, the bi-LSTM layer that encodes the selected inter-user is initialized with the output hidden state of the last time step. The number of time steps for each target Tweet is set to be a fixed number T so that the agent will terminate after T fetches. The final prediction occurs at the last time step. The computation is shown by the following equations.

Algorithm 1 Training Algorithm

```
for $t$ in training set do
    collect $X_t$ and $Z_t$;
    compute intra-user representation $r_{ia}(t)$;
end for
initialize parameters $\theta_p$ of the policy network;
initialize parameters $\theta_e$ of the other nets;
for $epoch = 1, E$ do
    for $t$ in training set do
        compute $o_{ta}(t)$, $r_{ta}(t)$;
        compute the raw prediction $y'(t)$;
        compute $b(X_t)$;
        $x_a = t$;
        compute $o_{ie}(t)$;
        initialize the state $s(t)_0$;
        for $i = 1, T$ do
            agent select action by $\epsilon - greedy$;
            update $x_a$;
            compute $o_{ie}(t)$, $r_{ie}(t)$;
            compute $y(t)_i$ and $s(t)_i$;
            compute the reward $v_i(t)$;
        end for
        apply REINFORCE to update $\theta_p$;
        update $\theta_e$ on the loss $\mathcal{L}(\theta_e) = e(y(t)_T, y^*)$;
    end for
end for
```

$$o_{ie}(x_a)_i, h_{ie}(x_a)_i = f_{ie}(x_a, h_{ie}(x_b)_{i-1}) \tag{5}$$

$$r_{ie}(x_a)_i = l_{ie}(\sigma(o_{ie}(x_a)_i)) \tag{6}$$

$$y'(t) = \sigma(l_c(r_{ta}(t) \oplus r_{ia}(t))) \tag{7}$$

$$y(t)_i = \sigma(l_c(r_{ie}(t) \oplus r_{ta}(t) \oplus r_{ia}(t))) \tag{8}$$

where x_b is the selected inter-user Tweet at time step $i - 1$. f_{ie} is the bi-LSTM of the inter-user branch. o_{ie} and h_{ie} are the output and the hidden state. l_c is a linear function. r_{ie} is the inter-user representation. y' is the prediction made without the inter-user branch and y is the prediction made with the inter-user branch. The symbol $\oplus$ means concatenation. The subscript i denotes time step i.

The state at each time step for the agent is the concatenation of encoded inter-user Tweets, the output of the Bi-LSTM in the inter-user branch and the baseline branch, together with the intra-user representation in the intra-user branch (the dotted arrows in Figure 2). Each inter-user Tweet x_j in X_t is encoded by the bi-LSTM of the inter-user branch (the dotted arrow through the Bi-LSTM of the inter-user branch).

$$b(x_j) = f_{ie}(x_j, \mathbf{0}) \tag{9}$$

$$s(t)_i[j] = o_{ie}(x_a)_i \oplus b(x_j) \oplus o_{ta}(t) \oplus r_{ia}(t) \tag{10}$$

b is the output of the bi-LSTM of the inter-user branch. In order to differentiate with o_{ie} mentioned above, we use b. $s(t)_i[j]$ is the jth row of the state at time step i.

By using reinforcement learning, the state for the agent is updated after each fetch of the inter-user Tweet. Thus, the agent can interactively make selections and update the inter-user representations step by step. The reward v_i for the agent at time step i is based on the original prediction without the agent and the prediction at the last time step with the agent. The computation is shown as:

$$q(t)_i = e(y'(t), y^*) - e(y(t)_T, y^*) \tag{11}$$

$$v(t)_i = \begin{cases} \alpha * q(t)_i & \text{if } y'(t)! = y^* \\ q(t)_i & \text{else if } y(t)_T! = y^* \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where e is the loss function; $q(t)$ is the basic reward; and $v(t)_i$ is the modified reward at time step i. α is a positive number used to amplify the reward when the original classification is incorrect. The intuition of this reward is to train the agent to be able to correct the misclassified Tweets. When the original prediction and the last prediction are both correct, the reward is set to 0 to make the agent focus on the misclassified instances.

The complete training process is shown in Algorithm 1. Before the training, the intra-user Tweets and inter-user Tweets are collected for each target Tweet. Then intra-user representations are computed, followed by the computation for initializing the environment and state for the agent. Next, the agent's actions, state updates, prediction, and reward are computed. Finally, the parameters are updated.

3 Experiments

3.1 Experimental Settings

Dataset: We use the dataset published by Waseem and Hovy (2016). This dataset contains 16,907 Tweets. The original dataset only contains the Tweet ID and the label for each Tweet. We expand the dataset with user ID and Tweet text. After deleting the Tweets that are no longer accessible, the dataset we use contains 15,781 Tweets from 1,808 users. The published dataset has three labels: racism, sexism and none. Since we consider a binary classification setting, we union the first two sets. In the final dataset, 67% are labeled as non-hate speech, and 33% are labeled as hate speech. 1000 Tweets are randomly selected for testing and the remaining 14,781 Tweets are for training.

Method	Prec.	Rec.	F1
SVM	**.793**	.656	.718
Logistic Regression	.782	.611	.686
Bi-LSTM + attention	.760	.665	.710
CNN-static	.701	.707	.703
CNN-non-static	.743	.699	.720
N-gram	.729	**.777**	.739
Bi-LSTM	.672	.737	.703
+ Intra. Rep.	.772	.749	.760
+ Intra.+ Randomized Inter. Rep.	.773	.764	.768
+ Intra.+ Reinforced Inter. Rep.	.775	.773	**.774**

Table 1: Experimental results. Prec.: precision. Rec.: recall. F1: F measure. Bi-LSTM: the baseline bidirectional LSTM model. Bi-LSTM + attention: an attentional bidirectional LSTM model. The experimental settings of the last three rows are illustrated in Section 3.1. + Intra. Rep.: the model consists of the target Tweet branch and the intra-user branch. + Intra. + Randomized Inter. Rep. incorporates randomly selected inter-user Tweets while + Intra. + Reinforced Inter. Rep. further incorporates the reinforced inter-user branch. The best results are in bold.

Baseline: The baseline model is a bi-LSTM model. The input for the model is the word embeddings of the target Tweet. The word embedding is of size 200. The hidden size of the LSTM is 64. The optimizer is Adam and we use mini-batches of size 25. The word embedding model is pre-trained on a Tweet corpus containing 3,433,513 Tweets.

Intra-user Representation Learning: Based on the target Tweet, we collect at most 400 Tweets posted by the same user, with the target Tweet removed. The baseline branch and the intra-user branch are combined via a linear layer.

Combining with Inter-user Representation: The inter-user Tweet set is collected from the dataset via Locality Sensitive Hashing (LSH). In our experiments, we use a set size of either 50, 100 or 200 Tweets. At each time step, one Tweet is selected from the inter-user Tweet set by the policy agent. We also experimented with a second setting, in which we replace the agent by random selection. At each time step, an inter-user Tweet is randomly selected from X and fed into the inter-user branch.

3.2 Results

We compare the above settings with six classification models: Supported Vector Machine (SVM) (Suykens and Vandewalle, 1999), Logistic Regression, attentional BI-LSTM, two CNN models by Kim (2014), and a N-gram model (Waseem and Hovy, 2016). We evaluate these models on three metrics: precision, recall and F1 score. The results are shown in Table 1. We report results for $|U| = 100$ in Table 1, as results with sizes 50 and 200 are similar. We find that leveraging the intra-user information helps reduce false positives. The performance is further improved when integrating our model with inter-user similarity learning. Our results show that selection by the policy gradient agent is slightly better than random selection, and we hypothesize the effect would be more salient when working with a larger unlabeled dataset. The McNemar's test shows that our model gives significantly better (at $p < 0.01$) predictions than the baseline bi-LSTM and attentional bi-LSTM.

3.3 Error Analysis

There are two types of hate speech that are misclassified. The first type contains rare words and abbreviations, e.g. *FK YOU KAT AND ANDRE! #mkr*. Such intentional misspellings or abbreviations are highly varied, making it difficult for the model to learn the correct meaning. The second type of hate speech is satire or metaphor, e.g. *Congratulations Kat. Reckon you may have the whole viewer population against you now #mkr*. Satire and metaphors are extremely difficult to recognize. In the above two cases, both the baseline branch and the inter-user branch can be unreliable.

4 Conclusion

In this work, we propose a novel method for hate speech detection. We use bi-LSTM as the baseline method. However, our framework can easily augment other baseline methods by incorporating intra-user and reinforced inter-user representations. In addition to detecting potential hate speech, our method can be applied to help detect suspicious social media accounts. Considering the relationship between online hate speech and real-life hate actions, our solution has the potential to help analyze real-life extremists and hate groups. Furthermore, intra-user and inter-user representation learning can be generalized to other text classification tasks, where either user history or a large collection of unlabeled data are available.

Acknowledgments

The first author is supported by the Chancellor's Fellowship. We gratefully acknowledge the support of NVIDIA Corporation with the donation of one Titan X Pascal GPU used for this research.

References

Pete Burnap and Matthew L Williams. 2016. Us and Them: Identifying Cyber Hate on Twitter across Multiple Protected Characteristics. *EPJ Data Science* 5(1):11.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *ECIR*. Springer, pages 693–696.

Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2(3):18.

Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *VLDB*. volume 99, pages 518–529.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*. ACM, pages 604–613.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090* .

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection using Natural Language Processing. In *SocialNLP'17: Proceedings of the 5th International Workshop on Natural Language Processing for Social Media*. pages 1–10.

Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9(3):293–300.

Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Véronique Hoste. 2015. Detection and Fine-grained Classification of Cyberbullying Events. In *RANLP'15: International Conference Recent Advances in Natural Language Processing*. pages 672–680.

William Warner and Julia Hirschberg. 2012. Detecting Hate Speech on the World Wide Web. In *ACL'12: Proceedings of the 2nd Workshop on Language in*

Social Media. Association for Computational Linguistics, pages 19–26.

Zeerak Waseem and Dirk Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *NAACL Student Research Workshop*. pages 88–93.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. 2012. Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus. In *CIKM'12: Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. ACM, pages 1980–1984.

Haoti Zhong, Hao Li, Anna Cinzia Squicciarini, Sarah Michele Rajtmajer, Christopher Griffin, David J Miller, and Cornelia Caragea. 2016. Content-Driven Detection of Cyberbullying on the Instagram Social Network. In *IJCAI'16: Proceedings of the 25th International Joint Conference on Artificial Intelligence*. pages 3952–3958.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Short Papers)*. volume 2, pages 207–212.