Early Detection of Online Hate Speech Spreaders with Learned User Representations

Darius Irani[1], Avyakta Wrat[2] and Silvio Amir[3]

[1]*Johns Hopkins University, Baltimore MD, USA*
[2]*Indian Institute of Technology, Bombay, India*
[3]*Northeastern University, Boston MA, USA*

Abstract

We developed and evaluated models for early detection of online hate speech spreaders. We addressed the problem as a social media author profiling task: given a small collection of tweets, the goal is to predict whether the author is likely to spread hate speech in the future (e.g. against women or immigrants). We investigated the impact of augmenting standard lexical representations with learned user-level representations from author-topic models and neural user embeddings. The evaluation was conducted on a dataset created for the social media author profiling shared task at PAN 2021. Our results show that: (i) learned user representations capture latent user aspects that correlate with the propensity to spread hate speech; and (ii) different user representations are complementary and can combined to improve hate speech detection.

Keywords

hate speech detection, social media analysis, user representation learning,

1. Introduction

The mass adoption of social media has been accompanied by a rise in abusive and toxic communications such as, cyberbullying and hate speech. Hate speech is commonly defined as any communication that disparages a person or a group on the basis of some characteristic such as race, colour, ethnicity, gender, sexual orientation, nationality or religion [1]. While most social platforms prohibit such behaviours, manually enforcing such policies is too laborious for the volume and velocity of user generated content. Despite being critical to ensure the safety and well-being of social media users, the problem of automatic hate speech detection has been relatively underappreciated. Most previous work has framed this as a document-level text classification task, i.e. predicting whether a *document* contains hate speech. However, clues from a single document are often insufficient to detect more nuanced forms of hate speech [2]. Moreover, operationalizing document-level hate speech detectors may still be too slow or burdensome for large social media platforms: offensive posts may still need to be manually flagged by the users; or every single post must be analyzed before publication.

In this paper, we address the problem of online hate speech detection as an author-profiling task, i.e. given a small set of social media posts, predict whether the *author* is likely to spread

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

✉ dr.irani21@gmail.com (D. Irani); avyaktawrat@gmail.com (A. Wrat); s.amir@northeastern.edu (S. Amir)

hate speech in the future (e.g. against women or immigrants). Approaching this problem from an author profiling perspective can help to overcome some of the limitations of document-level methods: first, a set of posts may provide a stronger signal than a single post; second, this opens the door for early detection systems to identify abusive users early on. In turn, this can expedite and reduce the burden of content moderation.

We investigate the impact of learned user representations on models for early detection of hate speech spreaders. Specifically, we develop classifiers with a combination of BOW features and user representations based on author-topic models estimated with Author-LDA [3] and user embeddings learned with User2Vec [4]. We conduct experiments on a dataset created for the 2021 edition of the PAN author profiling shared task: *Profiling hate speech Spreaders on Twitter*[1]. Our results show that: learned user representations capture relevant latent attributes for hate speech detection; (ii) different user-level representations capture complementary user aspects that can be combined to improve early detection of hate speech spreaders.

2. Social Media Author Profiling with User Representations

Social media content tends to be noisy and short, which poses challenges to traditional text classifiers. Standard bag-of-words models (BOW) yield very large and sparse vectors which are are often insufficient for inferences over nuanced content. Therefore, previous work on document-level online hate speech detection has sought to improve BOW models with additional features that capture high-level semantic properties of words and documents, such as latent topic models [5], word embeddings [6] and paragraph embeddings [7].

We address the problem of online hate speech detection as an author profiling task (note that we will use the terms *user* and *author* interchangeably). Similarly to document-level approaches, we seek to augment BOW models with additional features that capture high-level *personal aspects of the authors*. We hypothesize that augmenting local features from individual documents with learned user representations can improve online hate speech detection by: first, providing models with a *global* view of the author; and second, allowing models to discover latent personal aspects that correlate with the propensity to spread hate speech. We compare two approaches to learn user representations: author-topic models induced with Author LDA [3] and user embeddings induced with User2Vec [4]. User2Vec embeddings have been shown to capture meaningful latent attributes of social media users e.g., that correlate with political leanings [4] and mental-health status [8, 9]. Here, we assess whether these embeddings can also capture relevant attributes for hate speech detection.

Formally, let $\mathcal{C}_k = \{d_k^1, \ldots, d_k^M\}$ be a collection of documents authored by user u_k, where each post $d_k^j = \{w_1, \ldots, w_N\}$ is composed of words w_i from a vocabulary $\mathcal{V}$. We estimate the probability that user u_k is a hate speech spreader as

$$P(y = 1 | C_k, u_k) = f([\mathbf{c}_k, \mathbf{u}_k]) \tag{1}$$

where f is a binary classifier, $\mathbf{c_k} \in 0, 1^{|\mathcal{V}|}$ is a BOW representation of $\mathcal{C}_k$, and $\mathbf{u}_k \in \mathbb{R}^h$ is a learned representation of user u_k.

[1] https://pan.webis.de/clef21/pan21-web/author-profiling.html

2.1. Author-Topic Model

Latent Dirichlet Allocation (LDA) is a generative probabilistic model in which documents are represented as mixtures of latent topics, and topics are characterized as distributions over words drawn from a Dirichlet distribution [10]. LDA defines the following generative process for each document: (i) a distribution over topics Z is sampled from a Dirichlet distribution; (ii) for each word in the document, a single topic z is chosen according to this distribution; and (iii) each word is sampled from a multinomial distribution for topic z.

Author-topic models are an extension of LDA that includes authorship information by associating each author u_k with a distribution over topics [3]. The generative process is similar to LDA, however, the observed words are generated from a topic distribution sampled from an author-specific distribution over topics Z_{u_k}. By correlating authorship information with particular topics, author-topic models capture information about the topics that authors typically write about, and are able to represent documents in terms of these topics.

2.2. Neural User Embeddings

User2Vec aims to capture the relations between users and the content (i.e., the words) they generate, by optimizing the probability of sentences conditioned on their authors. Each user u_k is associated with a parameter vector $\mathbf{u}_k$, which is then optimized to accurately predict the words from previous posts written by said user

$$P(\mathcal{C}_k|u_k) \propto \sum_{d_k^j \in \mathcal{C}_k} \sum_{w_i \in d_k^j} \log P(w_i|\mathbf{u}_k) \tag{2}$$

Since the goal is to learn user representations, the term $P(w_i|\mathbf{u}_k)$ can be approximated with Negative Sampling [11] by minimizing the following Hinge-loss objective:

$$\mathcal{L}(\mathbf{w}_i, \mathbf{u}_k) = \sum_{\tilde{w}_j \in \mathcal{V}} \max(0, m - \mathbf{w}_i \cdot \mathbf{u}_k + \tilde{\mathbf{w}}_j \cdot \mathbf{u}_k) \tag{3}$$

where word $\tilde{w}_j$ (and associated embedding, $\tilde{\mathbf{w}}_j$) is a *negative sample*, i.e. a word not occurring in the post under consideration (authored by user u_k); and m is an hyperparameter that controls the loss margin. Note that both *words* and *users* are represented as d-dimensional vectors — pretrained word vectors $\mathbf{w}_i \in \mathbb{R}^d$ and user vectors $\mathbf{u}_k \in \mathbb{R}^d$ to be learned.

3. Experiments

We investigate the impact of learned **user-level** representations on the performance of author profiling models for online hate speech detection. As baselines, we consider **document-level** representations based on latent document-topics and word embeddings. Since the goal is to make predictions with respect to users, all document-level representations must be aggregated into a single user vector. We represent each user u_k by averaging all their associated document vectors $\mathbf{u}_k = \frac{1}{n} \sum_{d_k^i \in \mathcal{C}_k} \mathbf{d}_k^i$. We develop models that combine BOW features with the following learned representations.

Document Topics (LDA): Documents represented as mixtures of latent document-topics. We train a LDA topic model using the `lda` python package with the default hyperparameters. For each tweet, we use the model to estimate document-topic proportions and compute a feature vector $\mathbf{d} \in \mathbb{R}^h$, where $h = 50$ is the number of topics.

Word Embeddings (Avg-*): Documents represented as the average of pretrained embeddings associated with each word. For each tweet, we compute a feature vector $\mathbf{d} = \frac{1}{n} \sum [\mathbf{w}_1, \ldots, \mathbf{w}_n]$, where $\mathbf{w}_i \in \mathbb{R}^k$ is the embedding for word w_i and h is the embedding size. We leverage static embeddings trained with fastText [12] ($h = 400$) over a collection of 400 Million tweets[2]. We also experiment with contextualized word embeddings produced by deep pretrained language models: we extract ELMo [13] embeddings ($h = 256$) with a pretrained implementation available on the `AllenNLP` library[3]; and BERT [14] embeddings ($h = 768$) with a pretrained implementation available on the `Huggingface` library[4].

Author Topics (Author-LDA): Users represented as a mixture of latent author-topics. We train an Author-LDA model using the implementation available on the `Gensim` python library[5] with the default training hyperparameters. For each user, we estimate author-topic distributions and induce a feature vector $\mathbf{u} \in \mathbb{R}^h$ where $h = 50$ is the number of topics.

User Embeddings (User2Vec-*): Users represented as User2Vec user embeddings. For each user, we estimate an embedding $\mathbf{u} \in \mathbb{R}^h$, where h is the embedding size. We experimented with the same set of fastText, ELMo and BERT word representations. The embeddings were trained by minimizing Eq. 3 with ADAM [15] for 20 epochs with a fixed learning rate and early-stopping (using 20% of the tweets as a development set). We compare the performance of embeddings trained with different learning rates from the set $L = \{10, 1, 0.1, 0.01\}$ and the *margin* values from the set $M = \{1, 5, 10, 15\}$.

3.1. Evaluation

We evaluate our models on a dataset created for the social media author profiling shared task at PAN 2021: *Profiling Hate Speech Spreaders on Twitter.* The dataset contains 200 users annotated with binary labels indicating if they are hate speech spreaders; each user is also associated with a set of 200 tweets. We pre-process each tweet by lower-casing, removing white spaces and stop words and tokenizing the text with the Twokenize python package[6].

Given the small size of the training dataset and the lack of a validation dataset, we adopt a Leave-One-Out cross-validation methodology. We conduct preliminary experiments to select the best performing classifier for this task. We use `scikit-learn`[7] python library [16] to

[1] https://pypi.org/project/lda/

[2] https://github.com/FredericGodin/TwitterEmbeddings

[3] https://github.com/allenai/allennlp

[4] https://huggingface.co/

[5] https://radimrehurek.com/gensim/models/atmodel.html

[6] https://pypi.org/project/twokenize/

[7] https://scikit-learn.org

Table 1
Main results in terms of Accuracy. We compare the impact of augmenting BOW models (rows 1-2) with learned document-level representations (rows 3-6) and user-level representations (rows 7-10). The loss margin m and learning rate l used to train the User2Vec user embeddings are shown in parenthesis. The last two rows show the results of combining different user-level representations.

	Accuracy
BOW	0.595
+ Char-3	0.62
+ LDA	0.63
+ Avg-FastText	0.58
+ Avg-ELMo	0.63
+ Avg-BERT	0.635
+ Author-LDA	0.665
+ User2Vec-FastText ($m = 5; l = 0.1$)	0.65
+ User2Vec-ELMo ($m = 15; l = 0.1$)	0.685
+ User2Vec-BERT ($m = 5; l = 1$)	0.695
+ User2Vec-ELMo + User2Vec-BERT	0.705
+ Char-3 + User2Vec-ELMo + Author-LDA	**0.72**

implement the classifiers. We compare the performance of Support Vector Machines [17] and Random Forests [18] with different random seed initializations. We obtain the best results with Random Forests and thus adopt this classifier for the main experiments.

3.2. Results

Table 1 shows our main results in terms of Accuracy. As expected, BOW representations perform rather poorly by themselves. We can alleviate the problem of feature sparsity by including character n-grams, e.g. we obtain an improvement of 2% in Accuracy by adding character trigrams (Char-3). Regarding our main hypothesis, we observe that document-level representations based on document-topics and contextualized word embeddings provide only modest gains compared to a baseline of just lexical features (up to 1.5% in Accuracy) and static word embeddings actually hurt the models performance. Document-level representations are suboptimal for this task, in part because averaging features from multiple documents squashes all the information and thus the resulting vectors are less discriminative.

In contrast, all user-level representations yield noticeable gains. Indeed, we find that the user-level representations always outperform their document-level counterparts — i.e, replacing *document-topic* with *author-topic* representations improves performance by 3.5%, and replacing *word embeddings* with the corresponding *user embeddings* yields gains of up to 7.5%. Moreover, we see that User2Vec user embeddings induced with contextualized word embeddings (User2Vec-ELMo and User2Vec-BERT) outperform user embeddings from static word embeddings (User2Vec-FastText). While this is not surprising, it shows that User2Vec user embeddings can directly benefit from improvements on the underlying word representations.

Finally, we assess whether different user representations capture complementary user aspects that can be combined to improve the models. We find that ELMo user embeddings (User2Vec-ELMo) can be combined with BERT user embeddings (User2Vec-BERT) and with author-topic features (Author-LDA). Overall, these results confirm our hypothesis: learned user-level representations encode personal aspects that correlate with hate speech spreading behaviour. Our best performing model obtains an Accuracy of 72% with a combination of BOW, character tri-grams, ELMo user embeddings and author-topic features. The learned user representations improve the model's absolute performance by 10%.

4. Conclusions

We developed and evaluated social media author profiling models for the *Profiling Hate Speech Spreaders on Twitter* shared task at PAN 2021. Specifically, we investigated whether learned user-level representations capture latent aspects of users that correlate with hate speech spreading behaviour. We compared the impact of enriching BOW models with user-level representations based on latent author-topics and user embeddings, against document-level representations based on latent document-topics and word embeddings. While document-level representations provide some gains, we found that user-level representations yield much larger improvements (up to 10% in Accuracy). We believe that this is because user representations are able to complement local information from single documents with global information about the author.

Our results also show that User2Vec user embeddings induced with contextualized word embeddings perform better than static word embeddings; and that different embeddings capture complementary user aspects. However, it is not clear why these differences are observed. Moving forward we would like to probe into the learned representations to better understand what kinds of personal aspects are being captured by different models. In this work, we only considered contextualized word representations produced by the last layer of deep language models. In the future, we will investigate the impact of learning user representations using contextualized word representations from the inner layers of these models. This can open the door to more sophisticated user embedding methods, e.g. to induce *deep* or *contextualized* user embeddings.

This work presents an important first step towards early detection of online hate speech spreaders. However, the limited size of the dataset makes it difficult to draw definitive conclusions about the relative performance of each method. Moving this line of research forward may require larger annotated datasets.

References

[1] P. Fortuna, S. Nunes, A survey on automatic detection of hate speech in text, ACM Computing Surveys (CSUR) 51 (2018) 1–30.

[2] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: Proceedings of the fifth international workshop on natural language processing for social media, 2017, pp. 1–10.

[3] M. Rosen-Zvi, T. Griffiths, M. Steyvers, P. Smyth, The author-topic model for authors and documents, arXiv preprint arXiv:1207.4169 (2012).

[4] S. Amir, B. C. Wallace, H. Lyu, P. Carvalho, M. J. Silva, Modelling context with user embeddings for sarcasm detection in social media, in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 167–177.

[5] H. Zhong, H. Li, A. C. Squicciarini, S. M. Rajtmajer, C. Griffin, D. J. Miller, C. Caragea, Content-driven detection of cyberbullying on the instagram social network., in: IJCAI, volume 16, 2016, pp. 3952–3958.

[6] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, Y. Chang, Abusive language detection in online user content, in: Proceedings of the 25th international conference on world wide web, 2016, pp. 145–153.

[7] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, N. Bhamidipati, Hate speech detection with comment embeddings, in: Proceedings of the 24th international conference on world wide web, 2015, pp. 29–30.

[8] S. Amir, G. Coppersmith, P. Carvalho, M. J. Silva, B. C. Wallace, Quantifying mental health from social media with neural user embeddings, in: Machine Learning for Healthcare Conference, PMLR, 2017, pp. 306–321.

[9] S. Amir, M. Dredze, J. W. Ayers, Mental health surveillance over social media with digital cohorts, in: Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology, 2019, pp. 114–120.

[10] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, the Journal of machine Learning research 3 (2003) 993–1022.

[11] C. Dyer, Notes on noise contrastive estimation and negative sampling, arXiv preprint arXiv:1410.8251 (2014).

[12] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.

[13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proc. of NAACL, 2018.

[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[15] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.

[17] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.

[18] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.