Hate Speech Detection with Comment Embeddings

Nemanja Djuric, Jing Zhou, Robin Morris,
Mihajlo Grbovic, Vladan Radosavljevic, Narayan Bhamidipati
Yahoo Labs, 701 First Ave, Sunnyvale CA, USA
{nemanja, jingzh, rdm, mihajlo, vladan, narayanb}@yahoo-inc.com

ABSTRACT

We address the problem of hate speech detection in online user comments. Hate speech, defined as an "abusive speech targeting specific group characteristics, such as ethnicity, religion, or gender", is an important problem plaguing websites that allow users to leave feedback, having a negative impact on their online business and overall user experience. We propose to learn distributed low-dimensional representations of comments using recently proposed neural language models, that can then be fed as inputs to a classification algorithm. Our approach addresses issues of high-dimensionality and sparsity that impact the current state-of-the-art, resulting in highly efficient and effective hate speech detectors.

1. INTRODUCTION

In the age of ever-increasing volume and complexity of the internet, millions of users have unrestricted access to vast amounts of content that allows for privileges unimaginable several decades ago, such as access to knowledge bases or latest news within just a few clicks. However, due to internet's non-restrictive nature and, in certain countries, legal protection of free speech which also includes hate speech [4], some users misuse the medium to promote offensive and hateful language, which mars experience of regular users, affects business of online companies, and may even have severe real-life consequences [1]. To mitigate these detrimental effects, many companies (including Yahoo, Facebook, and YouTube) strictly prohibit hate speech on websites they own and operate, and implement algorithmic solutions to discern hateful content. However, scale and multifacetedness of the task renders it a difficult endeavour, and hate speech still remains a problem in online user comments.

Curiously, despite prevalence and large impact of online hate speech, to the best of our knowledge there exist only a few published works addressing this problem. In [1] (see also references therein) authors extract linguistic and bag-of-words (BOW) features and explore several classifiers to detect hateful tweets following the 2013 incident in Woolwich, UK. In [6] authors use BOW representation of user comments and train Support Vector Machine to filter anti-semitic content. Motivated by [6], authors of [2] use BOW and Naïve Bayes to flag racist comments. Interestingly, in all these works authors comment on limitations of BOW-based representation of text. This especially holds in the context of hate speech where offenders often use simple yet effective tricks to obfuscate their comments and make them more difficult for automatic detection (such as replacing or removing characters of offensive words), while still keeping the intent clear to a human reader. This results in high-dimensionality and large sparsity of the problem, making models susceptible to overfitting [6]. To address these issues, in this work we propose an approach that learns low-dimensional, distributed representations of user comments, allowing for efficient training of effective hate speech detectors.

We note that the task is different from, albeit related to, sentiment analysis [5] as there are no shades of hate speech and, unlike hate speech, even negative sentiment provides useful and actionable insights. Related work also includes attempts to remove offensive words without modifying the underlying meaning of comments [7]. This approach is however not applicable to hate speech detection as the conveyed message itself is considered harmful and should be removed.

2. PROPOSED APPROACH

We propose a two-step method for hate speech detection. First, we use paragraph2vec [3] for joint modeling of comments and words, where we learn their distributed representations in a joint space using the continuous BOW (CBOW) neural language model. This results in low-dimensional text embedding, where semantically similar comments and words reside in the same part of the space. Then, we use the embeddings to train a binary classifier to distinguish between hateful and clean comments. During inference, for newly observed comment, we infer representation by "folding in" using already learned word embeddings, as detailed in [3].

2.1 Neural language model

Neural language models take advantage of word order, and state the same assumption of n-gram language models that words that are close in a sentence are also statistically more dependent. In this work, we use the CBOW model as a component of paragraph2vec [3], which, based on the surrounding words, tries to predict the central word, as well as the user comment the words belong to.

More formally, let us assume we are given a set $\mathcal{D}$ of M documents, $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$, where each document d_m

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742760.

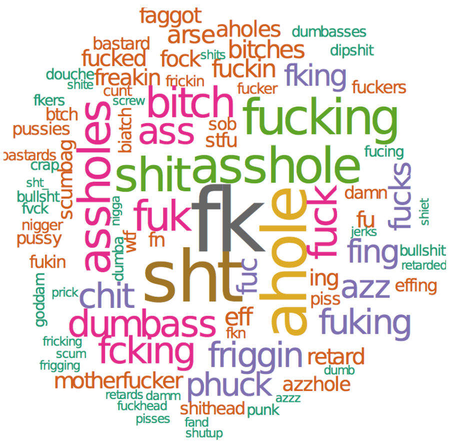

Figure 1: Nearest neighbors for swearword "fck"

is a sequence of T_m words, $d_m = (w_{m1}, w_{m2}, \dots, w_{m,T_m})$. We aim to simultaneously learn low-dimensional representations of documents and words in a common vector space, and represent each document and word as a continuous feature vector of dimensionality D. Then, the objective of paragraph2vec is to maximize the data log-likelihood,

$$\mathcal{L} = \sum_{d_m \in \mathcal{D}} \log \mathbb{P}(d_m | w_{m1}, w_{m2}, \dots, w_{m,T_m}) + \sum_{d_m \in \mathcal{D}} \sum_{w_{mt} \in d_m} \log \mathbb{P}(w_{mt} | w_{m,t-c}, \dots, w_{m,t+c}, d_m), \quad (1)$$

where c is the length of the context for word sequences. When modeling the probability of a document and the probability of a word, we define both models using a softmax function. Probability of the central word w_{mt} is defined as

$$\mathbb{P}(w_{mt} | w_{m,t-c}, \dots, w_{m,t+c}, d_m) = \frac{\exp(\bar{\mathbf{v}}^\top \mathbf{v}'_{w_{mt}})}{\sum_{w=1}^{V} \exp(\bar{\mathbf{v}}^\top \mathbf{v}'_{w})}, \quad (2)$$

where $\mathbf{v}'_{w_{mt}}$ is the output vector representation of w_{mt}, V is vocabulary size, and $\bar{\mathbf{v}}$ is an averaged vector representation of the context (including the containing comment d_m),

$$\bar{\mathbf{v}} = \frac{1}{2c+1}(\mathbf{v}_{d_m} + \sum_{-c \leq i \leq c, i \neq 0} \mathbf{v}_{w_{m,t+i}}). \quad (3)$$

We similarly define $\mathbb{P}(d_m | w_{m1}, \dots, w_{m,T_m})$, probability of a comment, by replacing appropriate variables in (2) and (3).

We use stochastic gradient ascent to maximize (1). However, compute time of $\nabla \log \mathbb{P}$ in (1) is proportional to vocabulary size, which may be expensive in practice. As an alternative we use hierarchical soft-max [3], which significantly reduces time complexity and allows for efficient training.

3. EMPIRICAL ANALYSIS

We evaluated our approach on a large-scale data set of user comments collected on Yahoo Finance website. The data set comprises 56,280 comments containing hate speech and 895,456 clean comments generated by 209,776 anonymized users, collected and editorially labeled over a 6-month period. We preprocessed the text by lowercasing and removing stopwords and special characters, resulting in a vocabulary size of V = 304,427. This makes the used data the largest hate speech data set considered thus far in the literature.

Table 1: AUC of various methods

Algorithm	AUC
BOW (*tf*)	0.7889
BOW (*tf-idf*)	0.6933
paragraph2vec	**0.8007**

We compared our method to the current state-of-the-art methods employing BOW representation, using *tf* and *tf-idf* encodings. We set $D = 200$ and $c = 5$ for paragraph2vec, while training on the entire data for 5 iterations. Once we learned vector representations we trained logistic regression classifier, and report the classification performance of competing methods after 5-fold cross-validation.

We first validated that the paragraph2vec representations are meaningful, and that semantically similar words are close to each other in the embedding space. This is illustrated in Figure 1, where we show a wordcloud of nearest neighbors in terms of cosine distance to obscured swearword "fck". We can see that using paragraph2vec resulted in this word, its variations, as well as semantically related swearwords having similar low-dimensional representations, grouping them in the same part of the vector space. Interestingly, the model even found some non-obvious swearwords, such as "chit".

Next, we validated utility of the learned vectors on the hate speech classification task. To this end, in Table 1 we report Area under the Curve (AUC), where we see that the proposed method outperformed the competing approaches. Interestingly, *tf* encoding achieved better performance than *tf-idf* and obtained very competitive AUC, which explains why many of the existing approaches use BOW representation. Nevertheless, paragraph2vec obtained higher AUC than either BOW model, while requiring less memory and training time to learn very effective hate speech detectors. The results clearly indicate the benefits of the proposed approach, and constitute a step towards solution of the problem of hate speech detection in online user comments.

4. REFERENCES

[1] P. Burnap and M. Williams. Hate speech, machine classification and statistical modelling of information flows on Twitter: Interpretation and communication for policy decision making. In *IPP*, 2014.

[2] I. Kwok and Y. Wang. Locate the hate: Detecting tweets against blacks. In *AAAI*, 2013.

[3] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv:1405.4053*, 2014.

[4] T. M. Massaro. Equality and freedom of expression: The hate speech dilemma. *Wm. & Mary L. Rev.*, 32:211, 1990.

[5] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[6] W. Warner and J. Hirschberg. Detecting hate speech on the World Wide Web. In *Workshop on Language in Social Media at ACL*, pages 19–26, 2012.

[7] Z. Xu and S. Zhu. Filtering offensive language in online communities using grammatical relations. In *Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, 2010.