Learning Representations for Detecting Abusive Language

Magnus Sahlgren
RISE AI and FOI
Box 1263, 164 29 Kista
Sweden
magnus.sahlgren@ri.se

Tim Isbister
FOI
164 90 Stockholm
Sweden
tim.isbister@foi.se

Fredrik Olsson
FOI
164 90 Stockholm
Sweden
fredrik.olsson@foi.se

Abstract

This paper discusses the question whether it is possible to learn a generic representation that is useful for detecting various types of abusive language. The approach is inspired by recent advances in transfer learning and word embeddings, and we learn representations from two different datasets containing various degrees of abusive language. We compare the learned representation with two standard approaches; one based on lexica, and one based on data-specific n-grams. Our experiments show that learned representations *do* contain useful information that can be used to improve detection performance when training data is limited.

1 Introduction

Abusive language is prevalent on the Internet of today. Many users of social media can attest to the frequent occurrence of negative slurs, racist and sexist comments, hate speech, cyberbullying, and outright threats. Commentary fields of news outlets, discussion forums, blogs, and normal websites are overflooded by abusive language, forcing administrators to restrict the possibility to comment on content, and in many cases removing this possibility altogether. As unfortunate as this development may be, it is hardly surprising. Our current information landscape has been designed to maximize the *effectiveness* of human communication, and factors such as transparency, trust and credibility have remained of peripheral concern for service providers. The combination of accessibility and anonymity of many online services provides the perfect conditions for "dark triad" behavior (Paulhus and Williams, 2002) to flourish. Even traditional news media, which have been seen as the last bastion for credibility and trust, are nowadays driven by the need for fast updates, sensationalism, and the hunt for clicks. It should serve as a cautionary observation that even fringe phenomena such as *trolling* fit comfortably in the current media landscape on the Internet (Phillips, 2015).

Our research focus in this paper is the question whether an inherently abusive environment such as a discussion forum or a white supremacist website can be used to learn a generic representation of abusive language, and whether such a representation can be used in supervised methods for detecting abusive language. Our work is inspired on the one hand by recent advances in transfer learning and pre-training of deep neural networks (Pan and Yang, 2010; Erhan et al., 2010; Peters et al., 2018), and on the other hand the use of embeddings as representation layer in text classification (Sahlgren and Cöster, 2004; Jin et al., 2016). We use two different data sources for learning representations (Stormfront and Reddit, further detailed in Section 4.1) and three different representation learning mechanisms (character-enhanced word embeddings, document-enhanced word embeddings, and a character-level language model, all further detailed in Section 3.3). We compare the proposed approaches with standard lexicon-based classification as well as supervised classification using Bag-of-Words n-gram representations.

2 Previous Work

The widespread occurrence of abusive language and behavior in online environments makes it necessary to devise detection methods to identify and mitigate such phenomena. Where the occurrence of abusive language is an increasing nuisance that may have economic consequences for service providers, it can be a matter of life and death for individuals and organizations who are targeted with more extreme forms of abusive language, such as explicit death threats. There has

Proceedings of the Second Workshop on Abusive Language Online (ALW2), pages 115–123
Brussels, Belgium, October 31, 2018.

been a fair amount of previous work on detecting various forms of abusive language. In particular the general concept of hate speech, which may include anything from negative remarks and racist comments to threats, has enjoyed a considerable amount of previous research (see e.g. Warner and Hirschberg (2012); Wester et al. (2016); Ross et al. (2016); Waseem and Hovy (2016); Davidson et al. (2017); Isbister et al. (2018)), demonstrating both the complexity of such a general problem as manifested by low inter-annotator agreement scores, but also the viability of using machine learning for detecting specific instances of hate speech in online conversations. Several researchers have focused on more specific types of abusive language, such as cyberbullying (see e.g. Reynolds et al. (2011); Nandhini and Sheeba (2015); Murnion et al. (2018)) and threats (e.g. Hammer (2014); Wester et al. (2016)), demonstrating the applicability of machine learning for detection purposes. The somewhat related, but more general, tasks of sentiment analysis and stance detection have a long history in Natural Language Processing (NLP), with a large body of literature on both theoretical and practical issues related to detection and monitoring (see e.g. Turney (2002); Pang and Lee (2008); Liu (2012); Pozzi et al. (2016); Kucher et al. (2017)).

3 Text Representations

Our primary interest in this study is the effect of text representations for the task of detecting abusive language. We consider three inherently different approaches: predefined sets of keywords (i.e. lexica), data-specific n-grams (i.e. Bag-of-Words), and pretrained embeddings. The following sections provide more details regarding the different approaches.

3.1 Lexica

The arguably most simplistic representation for detecting abusive language is to use a set of keywords (i.e. a lexicon) of abusive terms, possibly augmented by weights that quantify the relative importance of lexicon items. As an example, a lexicon with negative slurs may list terms such as "idiot", "redneck" and "white trash", and weights may be assigned that give higher importance to occurrences of "idiot" and "white trash" than to occurrences of "redneck". It is obvious that the coverage of the lexicon will be dependent on the inventiveness of the lexicographer. Coming up with an exhaustive list of all possible negative slurs is a daunting task, and (the almost certain) failure to do so will affect the coverage of the method. One way to alleviate this *synonymy* problem is to use unsupervised (or semi-supervised) machine learning to augment the compiled lexicon (Gyllensten and Sahlgren, 2018). Another obvious problem with keyword matching is *polysemy*, or the fact that words can have several different meanings. As an example, consider a statement such as "you should use the white trash can", which contains the abusive keyword "white trash", but does not signal negativity. Accounting for such context-sensitivity is a challenging problem (referred to as *word-sense disambiguation* in NLP) that affects the precision of the classification.

Despite these apparent drawbacks, lexicon-based classification is common in both sentiment analysis (see e.g. Taboada et al. (2011); Jurek et al. (2015)), and in hate speech detection (e.g. Njagi et al. (2015); Schmidt and Wiegand (2017); Isbister et al. (2018)). The two main reasons for this is simplicity and transparency. Simply defining a set of keywords and counting their occurrences in text is a quick and easy way to get an impression of the prevalence of some phenomenon. Furthermore, being able to precisely identify the matching keywords in text is a transparent and simple way to explain a classification decision to a user. Such explainability can be a very important consideration in practical applications.

In the following experiments, we use four previously published lexica relating to abusive language. We use the `baseLexicon` containing 537 tokens, and the `expandedLexicon` containing 2,680 tokens from (Wiegand et al.)[1], and the `hatebaseLexicon` containing 1,018 tokens, and the `refined_ngramLexicon` containing 178 tokens from (Davidson et al., 2017)[2]. We refer to these lexica simply as Lexicon #1, Lexicon #2, Lexicon #3, and Lexicon #4 in the remainder of this article. The original lexica are modified such that only terms that are certain to convey abusiveness, according to the lexicon creators, are retained and word class information is discarded. That is, in Lexicon #1, all terms labeled FALSE by the annotators are removed, and in Lexicon #2 all terms with a positive score (in-

[1] https://github.com/uds-lsv/lexicon-of-abusive-words

[2] https://github.com/t-davidson/hate-speech-and-offensive-language

dicating abusiveness) are kept. Lexicon #3 and #4 are used as-is. For the classification tasks using the lexica, a text is considered as abusive if it contains at least one term from a lexicon. Examples of abusive lexical entries include: "linthead", "alligator bait", and "you fuck wit".

3.2 Bag-of-Words

In contrast to a priori defining a set of representative terms, a *Bag-of-Words* (BoW) representation identifies informative terms directly from the training data. A BoW representation is an unordered collection of terms, in which each text is represented as an n-dimensional vector, where n is the size of the vocabulary, and each dimension encodes the weight (or *informativeness*) of a specific word in the current text. The standard term weight is some variant of TF-IDF (Sparck Jones, 1988), which combines a measure of the representativeness of a word for a text (often simply the frequency of the word in the text, TF) with a measure of how discriminative words are (often the inverse document frequency of a word, IDF). This "simple and proven" (Robertson and Jones, 1994) method for representing text is often employed in practical document processing applications, such as text categorization, document clustering, and information retrieval.

The main benefit of using BoW representations is that does not require any a priori knowledge, and that it operates completely on the given data. As such, a BoW representation may learn to use features that are not obvious for a lexicographer, and it may learn to use certain features in combination (e.g. that the occurrence of "can" in conjunction with "white trash" signals absence rather than presence of abusiveness). On the other hand, a BoW representation will be sensitive to out-of-vocabulary items, and it may learn to use features that do not make sense for a human analyst, which obviously decreases the explainability of the method.

In the following experiments, we augment standard BoW unigram features with character n-grams. This is normally done in order to account for morphological variation; if a training example contains the word "abuse" but a test example contains the word "abusive", a standard BoW representation will allocate different vector dimensions for these two words, and consequently there will be no similarity between the BoW representations for these texts. Using character n-grams, we would instead represent these words by their component character n-grams (up to some size of n), which would allocate the same vector dimensions to shared n-grams such as "abus", thus inducing similarity between their representations.

We use up to 4-grams for the character sequences, and we also allow for word bigrams in the representations. Word bigrams can be helpful to distinguish the use of collocations from use of the component terms. As an example, we would assign very different abusive scores to the two statements "you can use the trash can that is white" and "you are white trash"; it is not the individual occurrences of the words "white" and "trash" that is significant here, but the collocation "white trash". We weight the dimensions of the resulting representations using standard TF-IDF. For the classification tasks, the weighted n-gram representations are fed into a Logistic Regression classifier with L2 penalization.

3.3 Embeddings

While a lexicon relies completely on prior knowledge about the task and the domain, a BoW representation relies exclusively on task-dependent features. The idea of using pre-trained embeddings is a way to combine these two perspectives; we use bag-of representations to encode task-specific features, but take prior knowledge about the domain into account by learning a representation from representative data, which (in the best case) encodes latent variables that may be useful for relevant classification tasks.

This approach is inspired by recent advances in transfer learning and pre-training of deep neural networks (Pan and Yang, 2010; Erhan et al., 2010; Peters et al., 2018), in which a model that has been learned on some data is used as the foundation for learning a new model on new data. By doing so, the new model can take advantage of the previous knowledge already encoded in the representations of the pre-trained model. This is conceptually the same idea as using pre-trained word embeddings as representation layer in text classification (Sahlgren and Cöster, 2004; Jin et al., 2016), where the hope is that the embeddings can provide useful generalizations in comparison with only using standard BoW.

We investigate three flavors of this idea. The first flavor relies on character-enhanced word em-

beddings trained using the FastText model (Bojanowski et al., 2017), which uses the character n-grams of a set of context words to predict a target word. For those who are familiar with the word2vec model (Mikolov et al., 2013), this is essentially the same idea as the CBOW architecture, but using character n-grams instead of only word tokens. The resulting embeddings are used to produce text representations by simply averaging the TF-IDF-weighted vectors for all words in a text. For the embeddings, we use 300-dimensional vectors using the CBOW architecture with character n-grams, where n ranges between 3 to 6 characters. We use a window size of 5 tokens, and discard tokens that occur less than 100 times in the training data. These parameter settings are standard in the literature on embeddings, and are based on experience and trial and error.

The second flavor uses document-enhanced word embeddings trained using the Doc2Vec model (Le and Mikolov, 2014), which also relies on the architectures from word2vec, but adds a document id as input signal in addition to the word tokens. In this case, we use the distributed bag-of-words architecture, which predicts a set of word vectors based on a document vector (i.e. a document-based version of the SkipGram architecture). We use 300 dimensions for the embeddings, and the distributed bag-of-words architecture with a window size of 5, including only tokens that occur more than 100 times in the training data.

The third flavor uses a character-level language model that is trained to predict the next character given an input sequence of characters. We use a simple architecture consisting of one recurrent layer with Gated Recurrent Units (GRU) (Cho et al., 2014) using recurrent dropout, followed by a dense output layer using softmax activation. For training the network, we use adam optimization with learning rate decay, and a context size of 32 characters. For producing input vectors for training examples in the supervised classification experiments, we split the examples into consecutive chunks of 32 characters, and average the activations of the GRU layer over all chunks.

Despite being conceptually similar, there is an important difference between these three approaches that concerns the compositionality of the text representations. In the case of character-enhanced and document-enhanced word embeddings, we are essentially using bag-of representations that disregard the sequential nature of the data. That is, the average embedding for the sequences "Bob hates Mary" and "Mary hates Bob" will be exactly the same. This is not the case for the language model, which operates on the character sequences, and therefore will produce different compositional representations for these two sequences. It is an empirical question whether this difference is of practical importance when using the resulting representations as input to a supervised classifier.

4 Experiments

In order to compare the viability for detecting abusive language of the representations described in the previous sections, we use two different datasets for building embeddings, and four different datasets for validating classifiers based on the representations. Since our main focus in this paper is to study the effect of the representations rather than pursuing state of the art results, we use the same supervised classifier in all cases; a Logistic Regression classifier with L2 penalization. Before turning to the results, we describe the various datasets used in the experiments.

4.1 Data for Pre-Training

The three types of embeddings are trained on two different data sets; a collection of roughly 5 million posts from the white supremacist website Stormfront, and a random sample of approximately 54 million comments from the discussion forum Reddit. Both datasets were crawled specifically for these experiments.

The reason for selecting these data is that they can be expected to contain various levels of abusive language. In the case of Stormfront, which has been classified as a hate site (Levin, 2002), we can expect to find a wide diversity of abusive language, ranging from negative slurs of various sorts (racial, sexual, religious, political) to explicit threats and targeted hate. Reddit, on the other hand, is mainly a discussion forum where registered users discuss anything from general life style topics such as movies, gaming, and music, to very specialized topics such as machine learning and survivalism. Due to its diverse and inherently conversational nature, Reddit can also be expected to contain a fair amount of controversial topics and abusive language. However, it seems reasonable to assume that Stormfront contains a wider spectrum

Dataset	Token ratio	Doc. ratio
Stormfront	0.043	6.237
Reddit	0.041	1.497

Table 1: Ratios of occurrences of lexicon items in Stormfront vs. Reddit.

of abusive language, due to the fact that subjects who are active in white-supremacist environments tend to exhibit not only racial prejudice but *generalized prejudice*, which means we can also expect to find a substantial amount of sexism, homophobia, islamophobia, and so on.

As a simple demonstration of this, Table 1 shows the proportion of occurrences of terms from the various lexica introduced in Section 3.1 in Stormfront vs. Reddit. We count ratios of the occurrence of lexical items over both tokens and documents to account for differences in document length (comments on Stormfront are on average 143 words long, but only 36 words long on Reddit). The ratios in Table 1 demonstrate a slightly higher concentration of abusive terminology (especially when considering document ratio) in Stormfront compared to Reddit.

4.2 Data for Classification

Table 2 shows the four different datasets used to evaluate the viability of the different representations for detecting various forms of abusive language. The columns **Pos. used** and **Neg. used** specify the number of examples included in our experiments; we delimit all datasets to 10,000 data points (by random sampling) for efficiency reasons, and in order to use an equal amount of data in all experiments. Note that all datasets are highly imbalanced between the classes. This means that one could achieve high accuracy by simply guessing the majority class (i.e. the negative examples). To address this, we use weighted F1 score, which calculates F1 for each label and then their average is weighted by support – i.e. the number of true instances for each label. Note that the use of weighted F1 can result in an F-score that does not lie in between precision and recall. We also provide a baseline method similar to random guessing by using a stratified dummy classifier that generates predictions by respecting the training set's class distribution.

4.3 Results

The results for the various representations on the various datasets are shown in Table 3 (next side). It is obvious that all representations beat the baseline, which demonstrates that they all provide useful information. Starting with the lexica, it is interesting to note that Lexicon #1 (the `baseLexicon` containing 537 tokens) outperforms the other lexica in all Wikipedia datasets, despite Lexicon #2 and #3 being significantly bigger. Lexicon #4, which is the `refined_ngramLexicon` containing merely 178 tokens, performs only slightly worse than the other lexica. The biggest lexicon, the `expandedLexicon` with 2,680 tokens, performs best on the Twitter Hatespeech data. These differences demonstrate that lexica in general do not generalize well.

The best-performing lexicon for each dataset outperforms the character-level language model representations. However, none of the lexica beat the GRU representations on all datasets, which indicates that pretrained representations have better generalization capabilities than simple keyword matching approaches. This is further corroborated by the fact that the FastText representations consistently outperform the lexica (and the language model). The best performing embeddings are the Doc2Vec representations, which produce competitive results in particular for the Wikipedia datasets (aggression, attack, and toxicity).

However, the standard BoW n-gram representations outperform all other representations on all datasets. We include three variants of the BoW vectors. The row labeled "n-grams" contain results from data-specific BoW representa-

Dataset	Reference	Pos. available	Pos. used	Neg. available	Neg. used
Twitter Hatespeech	Waseem and Hovy (2016)	2,989*	2,989	8,270*	7,011
Wikipedia Aggression	Wulczyn et al. (2017)	21,496	1,647	94,368	8,353
Wikipedia Attack	Wulczyn et al. (2017)	19,627	1,492	96,237	8,508
Wikipedia Toxicity	Wulczyn et al. (2017)	23,023	1,205	136,663	8,795
*Note that since the publishing of Waseem and Hovy (2016), more than 5,000 tweets in the original corpus have become unavailable, and are thus not included in our experiment.					

Table 2: Datasets used for evaluating the representations in supervised classification of abusive language.

Representation	Hatespeech	Aggression	Attack	Toxicity
Baseline	0.610	0.700	0.720	0.753
Lexicon #1	0.664	0.795	0.806	0.831
Lexicon #2	0.729	0.751	0.758	0.783
Lexicon #3	0.647	0.777	0.798	0.824
Lexicon #4	0.626	0.749	0.771	0.804
n-grams	**0.871**	0.831	0.848	0.870
n-grams-Reddit	0.854	0.833	0.850	0.870
n-grams-Stormfront	0.851	0.826	0.844	0.869
n-grams-Doc2Vec-Reddit	0.857	**0.841**	**0.857**	**0.878**
FastText-Stormfront	0.749	0.796	0.814	0.843
FastText-Reddit	0.738	0.804	0.824	0.848
Doc2Vec-Stormfront	0.820	0.817	0.840	0.862
Doc2Vec-Reddit	0.816	0.824	0.846	0.869
GRU-Stormfront	0.719	0.780	0.798	0.828
GRU-Reddit	0.711	0.774	0.791	0.824

Table 3: Results for the various representations on the datasets used in these experiments. The baseline is based on random guessing in proportion to the class distributions, and the lexica use simple Boolean matching (i.e. presence or absence of lexicon terms). All other results are produced by feeding the representations to a Logistic Regression classifer with L2 penalization.

tions (i.e. n-grams collected from the training data), while the "n-grams-Reddit" and the "n-grams-Stormfront" use vocabulary collected from the Reddit and Stormfront data, respectively. The point of including all three variants is to study the effect of data-specific vocabulary, which only seems to have a clear positive effect on the Twitter Hatespeech data. This is hardly surprising, since the Twitter data features a lot of domain-specific terminology such as hashtags and @-mentions.

In order to investigate whether the best-performing embeddings (Doc2Vec) contribute additional information in comparison with the data-specific BoW n-grams, we also include results with a model that concatenates the data-specific BoW n-grams with the Doc2Vec-Reddit model. These augmented representations produce the best results on all the Wikipedia datasets, but lowers the score for the Twitter Hatespeech data. This demonstrates that the document-based embeddings *do* contribute useful information in addition to the BoW n-grams, but that domain-specific vocabulary is important to include. It could be interesting in future research to investigate whether Doc2Vec representations that have been trained on Twitter data would improve the results for the Twitter Hatespeech dataset.

Note that the FastText and Doc2Vec embeddings trained on Stormfront produce slightly better results for the Twitter Hatespeech data than the ones trained on Reddit, but the opposite is true for the Wikipedia data. One interpretation of this is that the Wikipedia data is more similar in nature to Reddit than to Stormfront; both Wikipedia discussions and Reddit are essentially conversational in nature, and typically do not contain explicit hatespeech to the extent present in the Twitter Hatespeech data and in the white supremacist website Stormfront. This analysis does not hold for the GRU language model, however, where the model trained on Stormfront produces slightly better results on all datasets. Although the differences in the results produced with the representations learned from Stormfront and Reddit are very small, they indicate that the choice of background data can have an influence on the suitability of the learned representations for specific classification tasks.

In order to further investigate the effect of using pretrained representations, we compute learning curves for the Logistic Regression classifier using the various representations, shown in Figures 1 to 4 (next side). The score for each set of training examples in these Figures is the average of 10 cross-validations. Note that the number of training examples on the x axis is log scale, since we want to investigate how the representations perform on limited amounts of training data. Note also the

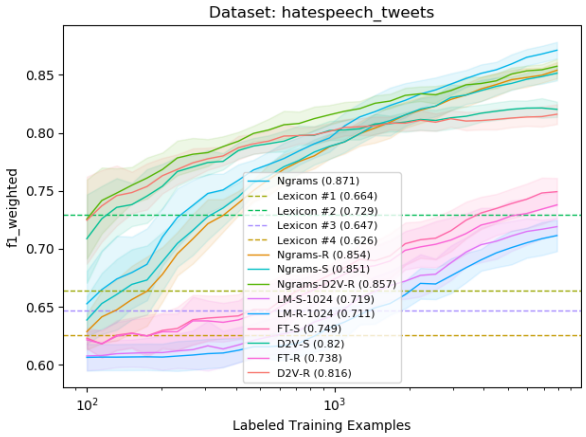

Figure 1: Learning curves for the Twitter Hatespeech data using different representations with a Logistic Regression classifier.

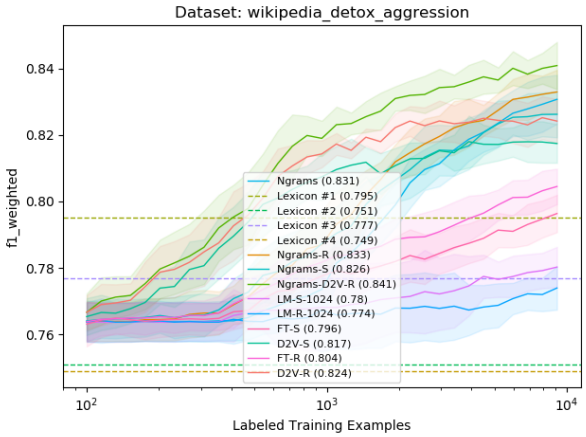

Figure 2: Learning curves for the Wikipedia Aggression data using different representations with a Logistic Regression classifier.

straight lines for the Baseline and Lexicon representations, which do not rely on training data.

The arguably most interesting aspect of the learning curves is the fact that the Doc2Vec representations lead to the best performance for all datasets when the amount of training data is limited. Up to a couple of hundred training examples, the Doc2Vec representation outperforms the BoW n-grams by a large margin. This does not apply for the other types of embeddings, however, which seem to require substantial amounts of training data in order to produce useful results. This suggests that when there is limited training data available, it is beneficial to utilize pretrained document-based embeddings as representation layer.

Note also that very few labelled examples are needed to beat the best-performing lexicon when using Doc2Vec embeddings. In the case of the Twitter Hatespeech data, only a hundred labelled examples are needed to beat the best lexicon. For the Wikipedia datasets, a couple of hundred examples are needed. This observation seems inconsistent with claims that lexica are more efficient to compile than collecting training data. Compiling a suitable lexicon with a couple of hundred relevant terms is hardly an easier, or more efficient, task than collecting a couple of hundred data samples. It could be an interesting future study to quantify the relative efforts involved in lexicon construction vs. data annotation.

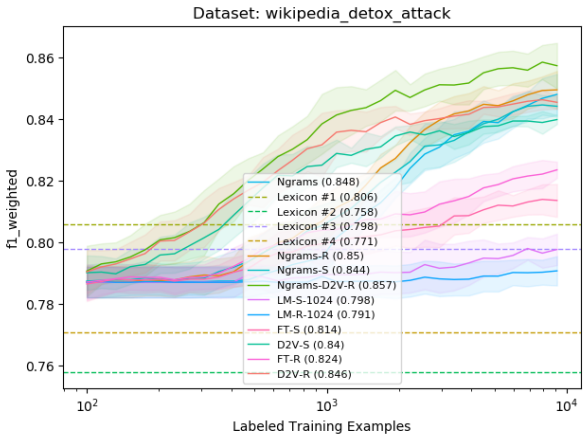

Figure 3: Learning curves for the Wikipedia Attack data using different representations with a Logistic Regression classifier.

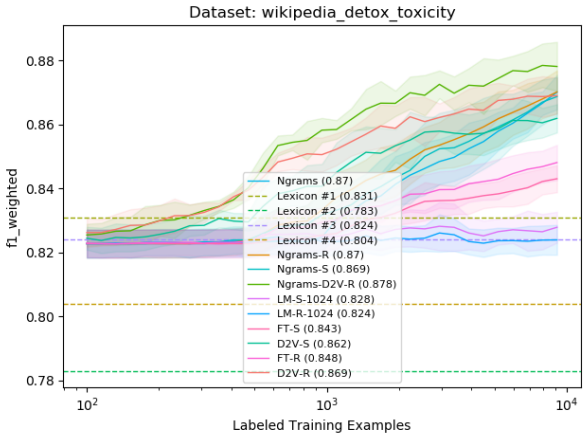

Figure 4: Learning curves for the Wikipedia Toxicity data using different representations with a Logistic Regression classifier.

5 Conclusions

This paper has investigated the question whether an inherently abusive environment such as a dis-

cussion forum or a white supremacist website can be used to learn a generic representation of abusive language, and whether such a representation can be used in supervised methods for detecting abusive language. The answer seems to be yes.

Our main result is the fact that pretrained embeddings, in particular Doc2Vec representations, trained on some relevant background data produce better results than standard BoW n-grams when training data is limited. We hypothesize that the results produced with the Doc2Vec representations will be very difficult to beat even when using state of the art methods if the classifier can only use a couple of hundred training examples. The fact that Doc2Vec representations produce more useful results than the other embeddings suggests that it could be interesting to investigate the use of more traditional document-based embedding techniques such as Latent Semantic Analysis (LSA) or Latent Dirichlet Allocation (LDA). We leave this as a suggestion for future research.

We acknowledge the fact that other machine learning methods may be more suitable to use for the input representations included in these experiments. We use Logistic Regression mainly for its simplicity and its well-known effectiveness. There have been many successful results using (deep) neural networks with pretrained embeddings, but these models learn complex internal representations that are difficult to interpret, which means that such models are less suitable to use when studying the effect of the input representations on the classification performance. Even so, it could be interesting to investigate whether the Doc2Vec embeddings produce the best results also when using other (deep) machine learning models.

Our results also show that lexica do not generalize well across tasks, and that only a couple of hundred training examples are needed for a supervised classifier based on pretrained document-based embeddings to beat the best-performing lexicon.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP 2014*, pages 1724–1734. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media*, pages 512–515.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660.

Amaru Cuba Gyllensten and Magnus Sahlgren. 2018. Distributional term set expansion. In *Proceedings of LREC 2018*.

H. L. Hammer. 2014. Detecting threats of violence in online discussions using bigrams of important words. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 319–319.

Tim Isbister, Magnus Sahlgren, Lisa Kaati, Milan Obaidi, and Nazar Akrami. 2018. Monitoring targeted hate in online environments. In *Second Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS 2018)*.

Peng Jin, Yue Zhang, Xingyuan Chen, and Yunqing Xia. 2016. Bag-of-embeddings for text classification. In *Proceedings of IJCAI 2016*, pages 2824–2830. AAAI Press.

Anna Jurek, Maurice D. Mulvenna, and Yaxin Bi. 2015. Improved lexicon-based sentiment analysis for social media analytics. *Security Informatics*, 4(1):9.

Kostiantyn Kucher, Carita Paradis, Magnus Sahlgren, and Andreas Kerren. 2017. Active learning and visual analytics for stance classification with alva. *ACM Transactions on Interactive Intelligent Systems*, 7(3):14:1–14:31.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of ICML 2014*, ICML'14, pages II–1188–II–1196. JMLR.org.

Brian Levin. 2002. Cyberhate: A legal and historical analysis of extremists' use of computer networks in america. *American Behavioral Scientist*, 45(6):958–988.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Shane Murnion, William J. Buchanan, Adrian Smales, and Gordon Russell. 2018. Machine learning and semantic analysis of in-game chat for cyberbullying. *Computers & Security*, 76:197 – 213.

B. Sri Nandhini and J.I. Sheeba. 2015. Online social network bullying detection using intelligence techniques. *Procedia Computer Science*, 45:485 – 492. International Conference on Advanced Computing Technologies and Applications (ICACTA).

Dennis Njagi, Z Zuping, Damien Hanyurwimfura, and Jun Long. 2015. A lexicon-based approach for hate speech detection. 10:215–230.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Delroy L Paulhus and Kevin M Williams. 2002. The dark triad of personality: Narcissism, machiavellianism, and psychopathy. *Journal of Research in Personality*, 36(6):556 – 563.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL: HLT*, pages 2227–2237. Association for Computational Linguistics.

Whitney Phillips. 2015. *This Is Why We Can'T Have Nice Things: Mapping the Relationship Between Online Trolling and Mainstream Culture*. The MIT Press.

Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Bing Liu. 2016. *Sentiment Analysis in Social Networks*, 1st edition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

K. Reynolds, A. Kontostathis, and L. Edwards. 2011. Using machine learning to detect cyberbullying. In *2011 10th International Conference on Machine Learning and Applications and Workshops*, volume 2, pages 241–244.

Stephen Robertson and K. Sparck Jones. 1994. Simple, proven approaches to text retrieval. Technical report.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2016. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. In *Proceedings of NLP4CMC III*. Bochumer Linguistische Arbeitsberichte.

Magnus Sahlgren and Rickard Cöster. 2004. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proceedings of COLING 2004*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10. Association for Computational Linguistics.

Karen Sparck Jones. 1988. Document retrieval systems. chapter A Statistical Interpretation of Term Specificity and Its Application in Retrieval, pages 132–142. Taylor Graham Publishing, London, UK, UK.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Peter D. Turney. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL 2002*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, LSM '12, pages 19–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016*, pages 88–93.

Aksel Wester, Lilja Øvrelid, Erik Velldal, and Hugo Lewi Hammer. 2016. Threat detection in online discussions. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 66–71, San Diego, USA.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. Inducing a lexicon of abusive words – a feature-based approach. In *Proceedings of NAACL: HTL*, pages 1046–1056. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Wikipedia Talk Labels: Personal Attacks.