Challenges for Toxic Comment Classification: An In-Depth Error Analysis

Betty van Aken[1], Julian Risch[2], Ralf Krestel[2], and Alexander Löser[1]

[1]Beuth University of Applied Sciences, Germany
{bvanaken, aloeser}@beuth-hochschule.de
[2]Hasso Plattner Institute, University of Potsdam, Germany
firstname.lastname@hpi.de

Abstract

Toxic comment classification has become an active research field with many recently proposed approaches. However, while these approaches address some of the task's challenges others still remain unsolved and directions for further research are needed. To this end, we compare different deep learning and shallow approaches on a new, large comment dataset and propose an ensemble that outperforms all individual models. Further, we validate our findings on a second dataset. The results of the ensemble enable us to perform an extensive error analysis, which reveals open challenges for state-of-the-art methods and directions towards pending future research. These challenges include missing paradigmatic context and inconsistent dataset labels.

1 Introduction

Keeping online conversations constructive and inclusive is a crucial task for platform providers. Automatic classification of toxic comments, such as hate speech, threats, and insults, can help in keeping discussions fruitful. In addition, new regulations in certain European countries have been established enforcing to delete illegal content in less than 72 hours.[1]

Active research on the topic deals with common challenges of natural language processing, such as long-range dependencies or misspelled and idiosyncratic words. Proposed solutions include bidirectional recurrent neural networks with attention (Pavlopoulos et al., 2017) and the use of pretrained word embeddings (Badjatiya et al., 2017). However, many classifiers suffer from insufficient variance in methods and training data and therefore often tend to fail on the long tail of real world data (Zhang and Luo, 2018). For future research, it is essential to know which challenges are already addressed by state-of-the-art classifiers and for which challenges current solutions are still error-prone.

We take two datasets into account to investigate these errors: comments on Wikipedia talk pages presented by Google Jigsaw during Kaggle's Toxic Comment Classification Challenge[2] and a Twitter Dataset by Davidson et al. (2017). These sets include common difficulties in datasets for the task: They are labeled based on different definitions; they include diverse language from user comments and Tweets; and they present a multi-class and a multi-label classification task respectively.

On these datasets we propose an ensemble of state-of-the-art classifiers. By analysing false negatives and false positives of the ensemble we get insights about open challenges that all of the approaches share. Therefore, our main contributions are:

1) We are the first to apply and compare a range of strong classifiers to a new public multi-label dataset of more than 200,000 user comments. Each classifier, such as Logistic Regression, bidirectional RNN and CNN, is meant to tackle specific challenges for text classification. We apply the same classifiers to a dataset of Tweets to validate our results on a different domain.

2) We apply two different pretrained word embeddings for the domain of user comments and Tweets to compensate errors such as idiosyncratic and misspelled words.

3) We compare the classifiers' predictions and show that they make different errors as measured by Pearson correlation coefficients and F1-measures. Based on this, we create an ensemble that improves macro-averaged F1-measure especially on sparse classes and data with high variance.

[1]https://www.bbc.com/news/technology-42510868

[2]https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

4) We perform a detailed error analysis on results of the ensemble. The analysis points to common errors of all current approaches. We propose directions for future work based on these unsolved challenges.

2 Related Work

Task definitions. Toxic comment classification is not clearly distinguishable from its related tasks. Besides looking at toxicity of online comments (Wulczyn et al., 2017; Georgakopoulos et al., 2018), related research includes the investigation of hate speech (Badjatiya et al., 2017; Burnap and Williams, 2016; Davidson et al., 2017; Gambäck and Sikdar, 2017; Njagi et al., 2015; Schmidt and Wiegand, 2017; Vigna et al., 2017; Warner and Hirschberg, 2012), online harassment (Yin and Davison, 2009; Golbeck et al., 2017), abusive language (Mehdad and Tetreault, 2016; Park and Fung, 2017), cyberbullying (Dadvar et al., 2013; Dinakar et al., 2012; Hee et al., 2015; Zhong et al., 2016) and offensive language (Chen et al., 2012; Xiang et al., 2012). Each field uses different definitions for their classification, still similar methods can often be applied to different tasks. In our work we focus on toxic comment detection and show that the same method can effectively be applied to a hate speech detection task.

Multi-class approaches. Besides traditional binary classification tasks, related work considers different aspects of toxic language, such as "racism" (Greevy and Smeaton, 2004; Waseem, 2016; Kwok and Wang, 2013) and "sexism" (Waseem and Hovy, 2016; Jha and Mamidi, 2017), or the severity of toxicity (Davidson et al., 2017; Sharma et al., 2018). These tasks are framed as multi-class problems, where each sample is labeled with exactly one class out of a set of multiple classes. The great majority of related research considers only multi-class problems. This is remarkable, considering that in real-world scenarios toxic comment classification can often be seen as a multi-label problem, with user comments fulfilling different predefined criteria at the same time. We therefore investigate both a multi-label dataset containing six different forms of toxic language and a multi-class dataset containing three mutually exclusive classes of toxic language.

Shallow classification and neural networks. Toxic comment identification is a supervised classification task and approached by either methods including manual feature engineering (Burnap and Williams, 2015; Mehdad and Tetreault, 2016; Waseem, 2016; Davidson et al., 2017; Nobata et al., 2016; Kennedy et al., 2017; Samghabadi et al., 2017; Robinson et al., 2018) or the use of (deep) neural networks (Ptaszynski et al., 2017; Pavlopoulos et al., 2017; Badjatiya et al., 2017; Vigna et al., 2017; Park and Fung, 2017; Gambäck and Sikdar, 2017). While in the first case manually selected features are combined into input vectors and directly used for classification, neural network approaches are supposed to automatically learn abstract features above these input features. Neural network approaches appear to be more effective for learning (Zhang and Luo, 2018), while feature-based approaches preserve some sort of explainability. We focus in this paper on baselines using deep neural networks (e.g. CNN and Bi-LSTM) and shallow learners, such as Logistic Regression approaches on word n-grams and character n-grams.

Ensemble learning. Burnap and Williams (2015) studied advantages of ensembles of different classifiers. They combined results from three feature-based classifiers. Further the combination of results from Logistic Regression and a Neural Network has been studied (Gao and Huang, 2017; Risch and Krestel, 2018). Zimmerman et al. (2018) investigated ensembling models with different hyper-parameters. To our knowledge, the approach presented in this paper, combining both various model architectures and different word embeddings for toxic comment classification, has not been investigated so far.

3 Datasets and Tasks

The task of toxic comment classification lacks a consistently labeled standard dataset for comparative evaluation (Schmidt and Wiegand, 2017). While there are a number of annotated public datasets in adjacent fields, such as hate speech (Ross et al., 2016; Gao and Huang, 2017), racism/sexism (Waseem, 2016; Waseem and Hovy, 2016) or harassment (Golbeck et al., 2017) detection, most of them follow different definitions for labeling and therefore often constitute different problems.

Class	# of occurrences
Clean	201,081
Toxic	21,384
Obscene	12,140
Insult	11,304
Identity Hate	2,117
Severe Toxic	1,962
Threat	689

Table 1: Class distribution of Wikipedia dataset. The distribution shows a strong class imbalance.

Class	# of occurrences
Offensive	19,190
Clean	4,163
Hate	1,430

Table 2: Class distribution of Twitter dataset. The majority class of the dataset consists of offensive Tweets.

3.1 Wikipedia Talk Pages dataset

We analyse a dataset published by Google Jigsaw in December 2017 over the course of the 'Toxic Comment Classification Challenge' on Kaggle. It includes 223,549 annotated user comments collected from Wikipedia talk pages and is the largest publicly available for the task. These comments were annotated by human raters with the six labels 'toxic', 'severe toxic, 'insult', 'threat', 'obscene' and 'identity hate'. Comments can be associated with multiple classes at once, which frames the task as a multi-label classification problem. Jigsaw has not published official definitions for the six classes. But they do state that they defined a toxic comment as "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion".[3]

The dataset features an unbalanced class distribution, shown in Table 1. 201,081 samples fall under the majority 'clear' class matching none of the six categories, whereas 22,468 samples belong to at least one of the other classes. While the 'toxic' class includes 9.6% of the samples, only 0.3% are labeled as 'threat', marking the smallest class.

Comments were collected from the English Wikipedia and are mostly written in English with some outliers, e.g., in Arabic, Chinese or German language. The domain covered is not strictly locatable, due to various article topics being discussed. Still it is possible to apply a simple categorization of comments as follows:[4]
1) 'community-related':

> Example: *"If you continue to vandalize Wikipedia, you will be blocked from editing."*

2) 'article-related':

> Example: *"Dark Jedi Miraluka from the Mid-Rim world of Katarr, Visas Marr is the lone surviving member of her species."*

3) 'off-topic':

> Example: *"== I hate how my life goes today == Just kill me now."*

3.2 Twitter dataset

Additionally we investigate a dataset introduced by Davidson et al. (2017). It contains 24,783 Tweets fetched using the Twitter API and annotated by CrowdFlower workers with the labels 'hate speech', 'offensive but not hate speech' and 'neither offensive nor hate speech'. Table 2 shows the class distribution. We observe a strong bias towards the offensive class making up 77.4% of the comments caused by sampling tweets by seed keywords from Hatebase.org. We choose this dataset to show that our method is also applicable to multi-class problems and works with Tweets, which usually have a different structure than other online user comments due to character limitation.

3.3 Common Challenges

We observe three common challenges for Natural Language Processing in both datasets:

Out-of-vocabulary words. A common problem for the task is the occurrence of words that are not present in the training data. These words include slang or misspellings, but also intentionally obfuscated content.

Long-Range Dependencies. The toxicity of a comment often depends on expressions made in early parts of the comment. This is especially problematic for longer comments (>50 words) where the influence of earlier parts on the result can vanish.

[3] http://www.perspectiveapi.com/

[4] Disclaimer: This paper contains examples that may be considered profane, vulgar, or offensive. These contents do not reflect the views of the authors and exclusively serve to explain linguistic research challenges.

Multi-word phrases. We see many occurrences of multi-word phrases in both datasets. Our algorithms can detect their toxicity only if they can recognize multiple words as a single (typical) hateful phrase.

4 Methods and Ensemble

In this section we study baseline methods for the above mentioned common challenges. Further, we propose our ensemble learning architecture. Its goal is to minimize errors by detecting optimal methods for a given comment.

4.1 Logistic Regression

The Logistic Regression (LR) algorithm is widely used for binary classification tasks. Unlike deep learning models, it requires manual feature engineering. Contrary to Deep Learning methods, LR permits obtaining insights about the model, such as observed coefficients. Research from Waseem and Hovy (2016) shows that word and character n-grams belong to one of the most indicative features for the task of hate speech detection. For this reason we investigate the use of word and character n-grams for LR models.

4.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) interpret a document as a sequence of words or character n-grams. We use four different RNN approaches: An LSTM (Long-Short-Term-Memory Network), a bidirectional LSTM, a bidirectional GRU (Gated Recurrent Unit) architecture and a bidirectional GRU with an additional attention layer.

LSTM. Our LSTM model takes a sequence of words as input. An embedding layer transforms one-hot-encoded words to dense vector representations and a spatial dropout, which randomly masks 10% of the input words, makes the network more robust. To process the sequence of word embeddings, we use an LSTM layer with 128 units, followed by a dropout of 10%. Finally, a dense layer with a sigmoid activation makes the prediction for the multi-label classification and a dense layer with softmax activation makes the prediction for the multi-class classification.

Bidirectional LSTM and GRU. Bidirectional RNNs can compensate certain errors on long range dependencies. In contrast to the standard LSTM model, the bidirectional LSTM model uses two LSTM layers that process the input sequence in opposite directions. Thereby, the input sequence is processed with correct and reverse order of words. The outputs of these two layers are averaged. Similarly, we use a bidirectional GRU model, which consists of two stacked GRU layers. We use layers with 64 units. All other parts of the neural network are inherited from our standard LSTM model. As a result, this network can recognize signals on longer sentences where neurons representing words further apart from each other in the LSTM sequence will 'fire' more likely together.

Bidirectional GRU with Attention Layer. Gao and Huang (2017) phrase that "attention mechanisms are suitable for identifying specific small regions indicating hatefulness in long comments". In order to detect these small regions in our comments, we add an attention layer to our bidirectional GRU-based network following the work of Yang et al. (2016).

4.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are recently becoming more popular for text classification tasks. By intuition they can detect specific combinations of features, while RNNs can extract orderly information (Zhang and Luo, 2018). On character level, CNNs can deal with obfuscation of words. For our model we choose an architecture comparable to the approach of Kim (2014).

4.4 (Sub-)Word Embeddings

Using word embeddings trained on very large corpora can be helpful in order to capture information that is missing from the training data (Zhang and Luo, 2018). Therefore we apply Glove word embeddings trained on a large Twitter corpus by Pennington et al. (2014). In addition, we use subword embeddings as introduced by Bojanowski et al. (2017) within the FastText tool. The approach considers substrings of a word to infer its embedding. This is important for learning representations for misspelled, obfuscated or abbreviated words which are often present in online comments. We train FastText embeddings on 95 million comments on Wikipedia user talk pages and article talk pages.[5] We apply the skip-gram method with a context window size of 5 and train for 5 epochs.

[5] https://figshare.com/articles/Wikipedia_Talk_Corpus/4264973

Model	Wikipedia				Twitter			
	P	R	F1	AUC	P	R	F1	AUC
CNN (FastText)	.73	.86	.776	.981	.73	.83	.775	.948
CNN (Glove)	.70	.85	.748	.979	.72	.82	.769	.945
LSTM (FastText)	.71	.85	.752	.978	.73	.83	.778	.955
LSTM (Glove)	**.74**	.84	.777	.980	.74	.82	.781	.953
Bidirectional LSTM (FastText)	.71	.86	.755	.979	.72	.84	.775	.954
Bidirectional LSTM (Glove)	**.74**	.84	.777	.981	.73	**.85**	.783	.953
Bidirectional GRU (FastText)	.72	.86	.765	.981	.72	.83	.773	.955
Bidirectional GRU (Glove)	.73	.85	.772	.981	.76	.81	.784	.955
Bidirectional GRU Attention (FastText)	**.74**	**.87**	**.783**	**.983**	.74	.83	**.791**	**.958**
Bidirectional GRU Attention (Glove)	.73	**.87**	.779	**.983**	**.77**	.82	**.790**	.952
Logistic Regression (char-ngrams)	**.74**	.84	.776	.975	.73	.81	.764	.937
Logistic Regression (word-ngrams)	.70	.83	.747	.962	.71	.80	.746	.933
Ensemble	**.74**	**.88**	**.791**	**.983**	.76	.83	**.793**	.953

Table 3: Comparison of precision, recall, F1-measure, and ROC AUC on two datasets. The results show that the ensemble outperforms the individual classifiers in F1-measure. The strongest individual classifier on both datasets is a bidirectional GRU network with attention layer.

4.5 Ensemble Learning

Each classification method varies in its predictive power and may conduct specific errors. For example, GRUs or LSTMs may miss long range dependencies for very long sentences with 50 or more words but are powerful in capturing phrases and complex context information. Bi-LSTMs and attention based networks can compensate these errors to a certain extent. Subword Embeddings can model even misspelled or obfuscated words.

Therefore, we propose an ensemble deciding which of the single classifiers is most powerful on a specific kind of comment. The ensemble observes features in comments, weights and learns an optimal classifier selection for a given feature combination. For achieving this functionality, we observe the set of out-of-fold predictions from the various approaches and train an ensemble with gradient boosting decision trees. We perform 5-fold cross-validation and average final predictions on the test set across the five trained models.

5 Experimental Study

Our hypothesis is that the ensemble learns to choose an optimal combination of classifiers based on a set of comment features. Because the classifiers have different strengths and weaknesses, we expect the ensemble to outperform each individual classifier. Based on results from previous experiments mentioned in Section 2 we expect that the state-of-the-art models have a comparable performance and none outperforms the others significantly. This is important because otherwise the ensemble learner constantly prioritizes the outperforming classifier. We expect our ensemble to perform well on both online comments and Tweets despite their differing language characteristics such as comment length and use of slang words.

5.1 Setup

To evaluate our hypotheses, we use the following setup: We compare six methods from Section 4. For the neural network approaches we apply two different word embeddings each and for LR we use character and word n-grams as features.

We need binary predictions to calculate precision, recall and the resulting F1-measure. To translate the continuous sigmoid output for the multi-label task (Wikipedia dataset) into binary labels we estimate appropriate threshold values per class. For this purpose we perform a parameter search for the threshold to optimize the F1-measure using the whole training set as validation. In case of the multi-class task (Twitter dataset) the softmax layer makes the parameter search needless, because we can simply take the label with the highest value as the predicted one.

We choose the macro-average F1 measure since it is more indicative than the micro-average F1

for strongly unbalanced datasets (Zhang and Luo, 2018). For the multi-label classification we measure macro-precision and -recall for each class separately and average their results to get the F1-measure per classifier. The Area under the Receiver Operating Curve (ROC AUC) gives us a measurement of classifier performance without the need for a specific threshold. We add it to provide additional comparability of the results.

5.2 Correlation Analysis

Total accuracy of the ensemble can only improve when models with comparable accuracy produce uncorrelated predictions. We therefore measure the correlation of the predictions of different classifiers. We look at a set of combinations, such as shallow learner combined with a neural net, and inspect their potential for improving the overall prediction. For measuring the disparity of two models we use the Pearson correlation coefficient. The results are shown in Table 4.

5.3 Experimental Results

As shown in Table 3 our ensemble outperforms the strongest individual method on the Wikipedia dataset by approximately one percent F1-measure. We see that the difference in F1-measure between the best individual classifier and the ensemble is higher on the Wikipedia dataset as on the Twitter dataset. This finding is accompanied by the results in Table 4 which show that most classifier combinations present a high correlation on the Twitter dataset and are therefore less effective on the ensemble. An explanation for this effect is that the text sequences within the Twitter set show less variance than the ones in the Wikipedia dataset. This can be reasoned from 1) their sampling strategy based on a list of terms, 2) the smaller size of the dataset and 3) less disparity within the three defined classes than in the six from the Wikipedia dataset. With less variant data one selected classifier for a type of text can be sufficient.

As the results in Table 4 show, ensembling is especially effective on the sparse classes "threat" (Wikipedia) and "hate" (Twitter). The predictions for these two classes have the weakest correlation. This can be exploited when dealing with strongly imbalanced datasets, as often the case in toxic comment classification and related tasks. Table 4 gives us indicators for useful combinations of classifiers. Combining our shallow learner approach with Neural Networks is highly effective. Contrary to that we see that the different word embeddings used do not lead to strongly differing predictions. Another finding is that word and character n-grams learned by our Logistic Regression classifier produce strongly uncorrelated predictions that can be combined for increasing accuracy.

Class	F1		Pearson
	Different word embeddings		
	GRU+G	GRU+FT	
W avg.	.78	.78	.95
W threat	.70	.69	.92
T avg.	.79	.79	.96
T hate	.53	.54	.94
	CNN+G	CNN+FT	
W avg.	.75	.78	.91
W threat	**.67**	**.73**	**.82**
T avg.	.77	.78	.94
T hate	.49	.53	.90
	Different NN architectures		
	CNN	BiGRU Att	
W avg.	**.78**	**.78**	**.85**
W threat	**.73**	**.71**	**.65**
T avg.	.78	.79	.96
T hate	.50	.49	.93
	Shallow learner and NN		
	CNN	LR char	
W avg.	**.78**	**.78**	**.86**
W threat	**.73**	**.74**	**.78**
T avg.	.78	.76	.92
T hate	**.50**	**.51**	**.86**
	BiGRU Att	LR char	
W avg.	**.78**	**.78**	**.84**
W threat	**.71**	**.74**	**.67**
T avg.	.79	.76	.92
T hate	.49	.51	.88
	Character- and word-ngrams		
	LR word	LR char	
W avg.	**.75**	**.78**	**.83**
W threat	**.70**	**.74**	**.69**
T avg.	.75	.77	.94
T hate	.50	.51	.91

Table 4: F1-measures and Pearson correlations of different combinations of classifiers. When the pearson score is low and F1 is similar, an ensemble performs best. We see that this appears mostly on the Wikipedia dataset and on the respective minority classes 'threat' and 'hate'. 'W': Wikipedia dataset; 'T': Twitter dataset; 'G': Glove embeddings; 'FT': FastText embeddings; 'avg.': Averaged

6 Detailed Error Analysis

The ensemble of state-of-the-art classifiers still fails to reach F1-measures higher than 0.8. To find out the remaining problems we perform an extensive error analysis on the result of the ensemble.

We analyse common error classes of our ensemble based on research from Zhang and Luo (2018); Zhang et al. (2018); Qian et al. (2018); Davidson et al. (2017); Schmidt and Wiegand (2017); Nobata et al. (2016). Moreover, we add additional error classes we encountered during our manual analysis. To address deficits in both precision and recall we inspect false negative and false positive classifications. We focus on error classes with the highest frequency in the observed samples. The occurrence of an error class within a comment is taken to be binary (occurs in comment or not).

We present the results on class 'toxic' of the Wikipedia dataset and class 'hate' of the Twitter dataset. Both classes are of high significance for the task of user comment moderation. Our ensemble results in 1794 false negatives and 1581 false positives for the Wikipedia dataset. We choose 200 random samples out of each set as representatives. For the smaller Twitter dataset we get 55 false negatives and 58 false positives, we perform our analysis on all of these samples.

6.1 Error Classes of False Negatives

Doubtful labels. We observe a high number of comments for which we question the original label when taking the respective class definition into account. A common occurrence is actual toxic or hateful content that is cited by the comment's author. Another pattern is the use of potentially toxic words within an explanation or self reproach.

Example: *"No matter how upset you may be there is never a reason to refer to another editor as 'an idiot' "*

We find that 23% of sampled comments in the false negatives of the Wikipedia dataset do not fulfill the toxic definition in our view. Taking the hate speech definition of the authors into account, we question 9% of the Twitter dataset samples. For the remaining error classes we only include the comments with undoubtful labels.

Toxicity without swear words. Davidson et al. (2017) phrase the problem that hate speech may not contain hate or swear words at all.

Example: *"she looks like a horse"*

50% of Wikipedia dataset samples have no common hate or swear word in them. This makes it the largest error class for the Wikipedia dataset and shows that our classifiers often fail when there are no obvious hateful words present. We observe this in 18% of hate speech comments from the Twitter dataset. It is important to notice that the frequency of swear words is naturally higher within this dataset, because of its sampling method with hateful words as seeds. In many cases the problem is a lack of paradigmatic context. Hence, an important research topic for future work is investigating improved semantic embeddings, which can better distinguish different paradigmatic contexts.

Rhetorical questions. It is common practice to wrap toxic statements online within rhetorical or suggestive questions as pointed out by Schmidt and Wiegand (2017).

Example: *"have you no brain?!?!"*

21% of Wikipedia dataset samples and 10% of Twitter dataset samples contain a rhetorical or suggestive question. Again paradigmatic context can help to identify this kind of comments. An additional signal is the existence of question words and question marks.

Metaphors and comparisons. Subtle metaphors and comparisons often require understanding of implications of language or additional world knowledge. Zhang and Luo (2018) and Schmidt and Wiegand (2017) report on this common error class.

Example: *"Who are you a sockpuppet for?"*

We only see this problem in the Wikipedia dataset samples with 16% of false negatives impacted.

Idiosyncratic and rare words. Errors caused by rare or unknown words are reported by Nobata et al. (2016); Zhang and Luo (2018); Qian et al. (2018). From our observation they include misspellings, neologisms, obfuscations, abbreviations and slang words. Even though some of these words appear in the embedding, their frequency may be too low to correctly detect their meaning on our word embeddings.

Example: *"fucc nicca yu pose to be pullin up"*

We find rare or unknown words in 30% of examined false negatives from the Wikipedia dataset and in 43% of Twitter dataset samples. This also reflects the common language on Twitter

with many slang words, abbreviations and misspellings. One option to circumvent this problem is to train word embeddings on larger corpora with even more variant language.

Sarcasm and irony. Nobata et al. (2016) and Qian et al. (2018) report the problem of sarcasm for hate speech detection. As sarcasm and irony detection is a hard task itself, it also increases difficulty of toxic comment classification, because the texts usually state the opposite of what is really meant.

Example: *"hope you're proud of yourself. Another milestone in idiocy."*

Sarcasm or irony appears in 11% of Wikipedia dataset samples, but in none of the Twitter dataset samples.

6.2 Error Classes of False Positives

Doubtful labels. We find that 53% of false positive samples from the Wikipedia dataset actually fall under the definition of toxic in our view, even though they are labeled as non-toxic. Most of them contain strong hateful expressions or spam. We identify 10% of the Twitter dataset samples to have questionable labels.

Example: *"IF YOU LOOK THIS UP UR A DUMB RUSSIAN"*

The analysis show that doubtful labels belong to one of the main reasons for a false classification on the Wikipedia dataset, especially for the false positives. The results emphasize the importance of taking labeler agreement into account when building up a dataset to train machine learning models. It also shows the need for clear definitions especially for classes with high variance like toxicity. Besides that, a deficient selection of annotators can amplify such problems as Waseem et al. (2018) point out.

Usage of swear words in false positives. Classifiers often learn that swear words are strong indicators for toxicity in comments. This can be problematic when non-toxic comments contain such terms. Zhang and Luo (2018) describe this problem as dealing with non distinctive features.

Example: *"Oh, I feel like such an asshole now. Sorry, bud."*

60% of false positive Wikipedia dataset samples and 77% of Twitter dataset samples contain swear words. In this case, the paradigmatic context is not correctly distinguished by the embedding. Hence, the classifier considered signals for the trigger word (a swear word) stronger, than other signals from the context, here a first person statement addressing the author himself.

Quotations or references. We add this error class because we observe many cases of references to toxic or hateful language in actual non-hateful comments.

Example: *"I deleted the Jews are dumb comment."*

In the Wikipedia dataset samples this appears in 17% and in the Twitter dataset in 8% of comments. Again the classifier could not recognize the additional paradigmatic context referring to typical actions in a forum, here explicitly expressed with words 'I deleted the…' and '…comment'.

Idiosyncratic and rare words. Such words (as described in Section 6) in non-toxic or non-hateful comments cause problems when the classifier misinterprets their meaning or when they are slang that is often used in toxic language.

Example: *"WTF man. Dan Whyte is Scottish"*

8% of Wikipedia dataset samples include rare words. In the Twitter dataset sample the frequency is higher with 17%, but also influenced by common Twitter language.

7 Conclusion

In this work we presented multiple approaches for toxic comment classification. We showed that the approaches make different errors and can be combined into an ensemble with improved F1-measure. The ensemble especially outperforms when there is high variance within the data and on classes with few examples. Some combinations such as shallow learners with deep neural networks are especially effective. Our error analysis on results of the ensemble identified difficult subtasks of toxic comment classification. We find that a large source of errors is the lack of consistent quality of labels. Additionally most of the unsolved challenges occur due to missing training data with highly idiosyncratic or rare vocabulary. Finally, we suggest further research in representing world knowledge with embeddings to improve distinction between paradigmatic contexts.

Acknowledgement

Our work is funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No. 732328 (FashionBrain) and by the German Federal Ministry of Education and Research (BMBF) under grant agreement No. 01UG1735BX (NOHATE).

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *WWW*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Pete Burnap and Matthew L. Williams. 2015. Cyber hate speech on twitter : An application of machine classification and statistical modeling for policy and decision making. volume 7, pages 223–242.

Pete Burnap and Matthew L. Williams. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Science*, 5:1–15.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. *SOCIALCOMPASSAT*, pages 71–80.

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

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *ICWSM*.

Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind W. Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *TiiS*, 2:18:1–18:30.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *ALW1@ACL*.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. In *RANLP*.

Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, and Vassilis P. Plagianakos. 2018. Convolutional neural networks for toxic comment classification. In *SETN*.

Jennifer Golbeck, Zahra Ashktorab, Rashad O. Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A. Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, Kelly M. Hoffman, Jenny Hottle, Vichita Jienjitlert, Shivika Khare, Ryan Lau, Marianna J. Martindale, Shalmali Naik, Heather L. Nixon, Piyush Ramachandran, Kristine M. Rogers, Lisa Rogers, Meghna Sardana Sarin, Gaurav Shahane, Jayanee Thanki, Priyanka Vengataraman, Zijian Wan, and Derek Michael Wu. 2017. A large labeled corpus for online harassment research. In *WebSci*.

Edel Greevy and Alan F. Smeaton. 2004. Classifying racist texts using a support vector machine. In *SIGIR*.

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*.

Akshita Jha and Radhika Mamidi. 2017. When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In *NLP+CSS@ACL*.

George W. Kennedy, Andrew W. McCollough, Edward Dixon, A. M. Parra Bastidas, J. Mark Ryan, and Chris Loo. 2017. Hack harassment : Technology solutions to combat online harassment. In *ALW1@ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *AAAI*.

Yashar Mehdad and Joel R. Tetreault. 2016. Do characters abuse more than words? In *SIGDIAL*.

Dennis Njagi, Z Zuping, Damien Hanyurwimfura, and Jun Long. 2015. A lexicon-based approach for hate speech detection. In *International Journal of Multimedia and Ubiquitous Engineering*, volume 10, pages 215–230.

Chikashi Nobata, Joel R. Tetreault, Achint Oommen Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *WWW*.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. *CoRR*, abs/1706.01206.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Michal Ptaszynski, Juuso Kalevi Kristian Eronen, and Fumito Masui. 2017. Learning deep on cyberbullying is always better than brute force. In *IJCAI*.

Jing Qian, Mai ElSherief, Elizabeth M. Belding-Royer, and William Yang Wang. 2018. Leveraging intra-user and inter-user representation learning for automated hate speech detection. In *NAACL-HLT*.

Julian Risch and Ralf Krestel. 2018. Aggression identification using deep learning and data augmentation. In *TRAC-1@COLING*, pages 150–158.

David Robinson, Ziqi Zhang, and Jonathan Tepper. 2018. Hate speech detection on twitter: Feature engineering v.s. feature selection. In *ESWC*.

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. *CoRR*, abs/1701.08118.

Niloofar Safi Samghabadi, Suraj Maharjan, Alan Sprague, Raquel Diaz-Sprague, and Thamar Solorio. 2017. Detecting nastiness in social media. In *ALW1@ACL*.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *SocialNLP@EACL*.

Sanjana Sharma, Saksham Agrawal, and Manish Shrivastava. 2018. Degree based classification of harmful speech using twitter data. *CoRR*, abs/1806.04197.

Fabio Del Vigna, Andrea Cimino, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook. In *ITASEC*.

W. Lloyd Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *LSM@ACL*.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *NLP+CSS@EMNLP*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *SRW@NAACL-HLT*.

Zeerak Waseem, James Thorne, and Joachim Bingel. 2018. Bridging the gaps: Multi task learning for domain transfer of hate speech detection. *Online Harassment*, pages 29–55.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *WWW*.

Guang Xiang, Bin Fan, Ling Wang, Jason I. Hong, and Carolyn Penstein Rosé. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *CIKM*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*.

Dawei Yin and Brian D. Davison. 2009. Detection of harassment on web 2 . 0. In *CAW2.0@WWW*.

Ziqi Zhang and Lei Luo. 2018. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *CoRR*, abs/1803.03662.

Ziqi Zhang, David Robinson, and Jonathan A. Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *ESWC*.

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*.

Steven Zimmerman, Udo Kruschwitz, and Chris Fox. 2018. Improving hate speech detection with deep learning ensembles. In *LREC*.