Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pages 137–143
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Offensive Language Detection Explained

Julian Risch,[1] **Robin Ruff,**[2,3] **Ralf Krestel**[1,2]
[1]Hasso Plattner Institute, University of Potsdam, [2]University of Passau, [3]Karlsruhe Institute of Technology
[1]Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
[2]Innstraße 41, 94032 Passau, Germany
[3]Kaiserstraße 12, 76131 Karlsruhe, Germany
julian.risch@hpi.de, upnub@student.kit.edu, ralf.krestel@uni-passau.de

Abstract

Many online discussion platforms use a content moderation process, where human moderators check user comments for offensive language and other rule violations. It is the moderator's decision which comments to remove from the platform because of violations and which ones to keep. Research so far focused on automating this decision process in the form of supervised machine learning for a classification task. However, even with machine-learned models achieving better classification accuracy than human experts in some scenarios, there is still a reason why human moderators are preferred. In contrast to black-box models, such as neural networks, humans can give explanations for their decision to remove a comment. For example, they can point out which phrase in the comment is offensive or what subtype of offensiveness applies. In this paper, we analyze and compare four attribution-based explanation methods for different offensive language classifiers: an interpretable machine learning model (naive Bayes), a model-agnostic explanation method (LIME), a model-based explanation method (LRP), and a self-explanatory model (LSTM with an attention mechanism). We evaluate these approaches with regard to their explanatory power and their ability to point out which words are most relevant for a classifier's decision. We find that the more complex models achieve better classification accuracy while also providing better explanations than the simpler models.

Keywords: neural networks, offensive language detection, explanation methods

1. The Need for Explanations

Online news platforms (e.g., New York Times), question answering platforms (e.g., Stack Overflow), collaborative projects (e.g., Wikipedia), and social networks (e.g., Facebook): all these social media platforms have one thing in common. They provide a discussion space for users, where content moderators are employed to keep a respectful tone, and foster fruitful discussions. Moderators ensure that the platform's discussion rules are adhered to, including the ban of offensive language. They enforce these rules by partially or entirely removing a user comment.

Typically, a platform's rules are listed in the form of guidelines, and they overlap considerably with the "netiquette", the basic rules about communication over the Internet. However, that does not mean all users have these rules in mind when they post comments. Moderators on online discussion platforms, therefore, explain why they intervene.

For example, they replace a removed comment with the following text: "Removed. Please refrain from insults." or "Removed. Please refrain from insinuations and personal attacks.". In case they ultimately close a comment section, they post a final comment, for example, stating: "This comment section has been closed due to (racist) generalizations, baseless assumptions up to conspiracy theories and extreme polemics.". On the one hand, the idea behind these explanations is transparency. On the other hand, they aim to educate users to adhere to the discussion rules.

Research on comment classification focuses on supervised machine learning approaches and often uses black-box models. For example, there is research on detecting hate speech (Gao and Huang, 2017), racism/sexism (Waseem and Hovy, 2016) or offensive/aggressive/abusive language (Struß et al., 2019; Kumar et al., 2018). However, to support moderators, semi-automated comment moderation in the form of a pre-classification of comments (Risch and Krestel, 2018) is not enough. Black-box models lack the ability to give explanations for their automated decisions. Therefore, they cannot be properly applied to comment moderation. Users and moderators are skeptical about an incomprehensible automation. Explanations help to build trust and increase the acceptance of machine-learned classifiers. Only then can a fair and transparent moderation process be ensured.

There are two more reasons for explanations in general. First, there are legal reasons to utilize machine-learned classifiers only if they can give explanations for their decisions. For example, under certain circumstances, the General Data Protection Regulation (GDPR) in the EU grants users the right to "obtain an explanation of the decision reached" if they are significantly affected by automated decision-making, e.g., if a credit application is refused.[1] A second reason is that explanations help to reveal the strengths and weaknesses of a model. They could also benefit the task of identifying a potential bias in a model's decisions. Researchers can then work on improving the models based on these insights.

Contributions The main contribution of this paper is the evaluation and comparison of attribution-based explanation methods for offensive language detection. To this end, we use a word deletion task to compare an interpretable machine learning model (naive Bayes), a model-agnostic explanation method (LIME), a model-based explanation method (LRP), and a self-explanatory model (LSTM with an attention mechanism). In a second experiment, we use the explanatory power index (EPI) as a metric to evaluate the approaches. Further, we take into account the classifi-

[1] https://eur-lex.europa.eu/eli/reg/2016/679/oj

cation accuracy of each approach and discuss strengths and weaknesses in the application context of automated content moderation. Based on this discussion, we give directions for future work.

Outline In the following, we summarize related work on explanation methods in Section 2 and describe which of these methods and what classifiers we implement for offensive language detection in Section 3. Section 4 evaluates the methods with the help of a word deletion task and the explanatory power index (EPI), while Section 5 discusses the results. We conclude with a summary of the contributions and an outlook on future work in Section 6.

2. Related Work

There is plenty of research on offensive language detection, and the classification accuracy for this task drastically increased in recent years — not least due to deep learning approaches for natural language processing. However, one aspect of this classification task has gone mostly unnoticed: the need for explaining classification results.

More precisely, research on explanation methods distinguishes explainability from interpretability. The former refers to locally comprehending individual decisions, while the latter refers to globally comprehending the decision function (Došilović et al., 2018; Monroe, 2018; Montavon et al., 2017). Unfortunately, there is no universal definition of these two terms. The definition used in this paper is:

- A decision function f is called explainable if the decision $f(x)$ for each single input $x \in X$ (in domain X) can be explained in understandable terms to humans.
- A decision function f is called interpretable if the whole function f (for the whole domain X) can be explained in understandable terms to humans.

In the field of image classification, CNN-based explanation methods are prominent. For example, DeConvNet (Zeiler and Fergus, 2014) inverts the convolutional operations to gain explanations and an approach by Simonyan et al. (2014) applies sensitivity analysis to achieve similar results. There have been several follow-up papers that compare these two approaches and propose combinations (Kindermans et al., 2018; Springenberg et al., 2015).

Explanation methods for text classification are rarely studied. For example, Nguyen (2018) compares human evaluation and automatic evaluation for explanation methods. The comparison uses the twenty newsgroups dataset and a dataset of movie reviews. To the best of our knowledge, the only publication on explanation methods in the field of offensive language detection is by Carton et al. (2018). The authors use an attention mechanism to generate explanations for the detection of personal attacks.

An empirical study by Chakrabarty et al. (2019) shows the importance of contextual or self-attention for abusive language detection. Whether attention weights can also be used as explanations is under discussion (Wiegreffe and Pinter, 2019; Jain and Wallace, 2019). In this paper, we consider a long short-term memory (LSTM) neural network (Hochreiter and Schmidhuber, 1997; Gers et al., 1999) with an attention mechanism (Yang et al., 2016) as an example of a self-explanatory model. The inherent attention weights provide attribution-based explanations. Further, we consider a naive Bayes classifier, which is an example of an interpretable model. A classification result (and the entire model) can be understood with the help of the discrete conditional probabilities in the classifier. The relevance of a word w is the probability that the class c is predicted given w:

$$P(c|w) = \frac{P(c) \cdot P(w|c)}{P(w)}$$

The attention-based LSTM and the naive Bayes classifier are two a priori explainable models. We also consider two post-hoc explanation methods in our paper: layer-wise relevance propagation (LRP) and local interpretable model-agnostic explanations (LIME). We describe these two methods in the following. The idea behind LRP (Bach et al., 2015) is to backpropagate the relevance scores from the output layer to the input layer of a neural network. To this end, the relevance of each input value (feature) is derived from the neuron activations in the output layer. This procedure makes LRP a *model-based* explanation method.

The idea behind LIME (Ribeiro et al., 2016) is to use a local approximation of the classifier f at a point x and its neighborhood. This local approximation needs to be an interpretable classifier and a good approximation of f in the local neighborhood of point x. The authors evaluate their *model-agnostic* explanation method with text and image classification tasks.

3. Explanation Methods

For our comparative study, we implement a variety of classifiers for offensive language detection and suitable explanation methods. To train the classifiers, we use a dataset of toxic comments published by Google Jigsaw in the context of a Kaggle challenge.[2] The Python code for all classifiers, a web application to visualize the explanations, and the training and evaluation procedures are published online.[3]

3.1. Classifiers

There are four different classifiers that we implement and pair with different attribution-based explanation methods. First, there is a multinomial naive Bayes classifier, which serves as a baseline. It is interpretable by default and provides explanations in the form of conditional probabilities. Further, we implement a support vector machine (SVM) and a long short-term memory (LSTM) neural network. The input to the SVM is a TF-IDF vector representation of the unigrams in the comment text. GloVe word embeddings (Pennington et al., 2014) serve as the input to the neural network.

Both the SVM and the LSTM network are paired with the two explanation methods LRP and LIME. To this end, we adapt the LRP implementation by Arras et al.[4] and the

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

[3] https://hpi.de/naumann/projects/repeatability/text-mining.html

[4] https://github.com/ArrasL/LRP_for_LSTM/

Table 1: Absolute and relative frequency of the six class labels in the training dataset and test dataset. The class distribution is highly imbalanced.

Class	Training Set		Test Set	
Toxic	19,235	9.56%	2,149	9.61%
Severe Toxic	1,757	0.87%	205	0.92%
Obscene	10,922	5.43%	1,218	5.45%
Threat	617	0.31%	72	0.32%
Insult	10,178	5.06%	1,126	5.04%
Identity Hate	1,906	0.95%	211	0.94%

LIME implementation by Ribeiro et al.[5]. To generate explanations for SVM and LSTM with the model-agnostic method LIME, we first sample perturbations of the input text by randomly deleting words. For each sample, we calculate the class probabilities with the SVM and the LSTM by applying a softmax function as the final calculation step. The default ridge regression algorithm is used to train an interpretable linear model. This model learns the word relevance scores bases on the classified samples.
Last but not least, we implement an LSTM network with an attention mechanism, which is an example of a self-explanatory model. It uses attention weights on the word level (not on the sentence level) and implements the architecture by Yang et al. (2016).

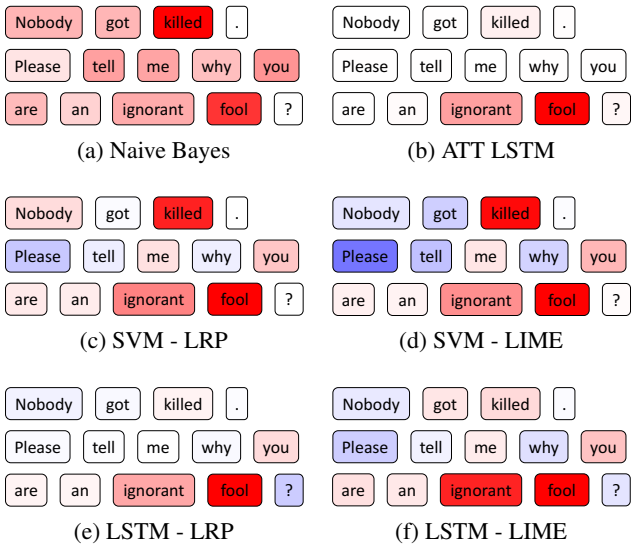
3.2. Dataset

The *toxic comments* dataset contains about 220,000 comments, each labeled with regard to six non-exclusive classes: *toxic*, *severe toxic*, *insult*, *threat*, *obscene*, and *identity hate*. Table 1 shows the class distribution in the training set and test set. Note that a comment is always labeled as toxic if one of the other labels applies. Even if none of the other labels apply, it can still be labeled as toxic.

3.3. Training Procedure

The GloVe word embeddings are trained from scratch on the training and test set. We restrict the input length of the basic LSTM network and the LSTM network with an attention mechanism to a maximum of 250 words. Further, we use 50 LSTM units, which means the output of this layer is 50-dimensional. The training of the networks runs for five, respectively, three epochs with the Adam optimizer until the validation loss increases.
The task on our dataset is a multi-label classification task. Our network architecture addresses this multi-label task by sharing the same LSTM layer across all class labels. However, for each label, an independent fully-connected layer follows after the output of the last LSTM unit. The attention mechanism is also trained for each label individually and fits in between the LSTM output and the following fully-connected layer.
SVM and naive Bayes use stemming to reduce the vocabulary size. They are trained according to a one-against-all scheme to conform to the multi-label classification task. The trained models therefore can be seen as six independent

[5] https://github.com/marcotcr/lime



(a) Naive Bayes (b) ATT LSTM (c) SVM - LRP (d) SVM - LIME (e) LSTM - LRP (f) LSTM - LIME

Figure 1: This heatmap visualizes positive (red) and negative (blue) word relevance scores generated by combinations of different classifiers and explanation methods.

binary naive Bayes classifiers, respectively, six independent binary SVMs. The SVM uses a linear kernel. There is only one hyperparameter to choose, which is the regularization term c. We set $C = 0.6$ and thereby relax the penalty for misclassifications.

3.4. Heatmap Visualization

To give an example of the explanations, Figure 1 and Figure 2 visualize the word relevance scores generated by the different explanation methods for two toxic comments. The conditional probabilities of the naive Bayes approach and the attention weights of the attention-based LSTM define positive word relevance scores between 0 and 1. In contrast to that, LIME and LRP define unbound relevance scores, which can also be negative. A negative word relevance score means that the respective word indicates the absence of a particular class rather than its presence. Because the attention weights are class-independent, these weights can only explain the predicted class. All other methods can also be used to explain a class that was not predicted by the classifier. This property can be used to analyze which words speak in favor of a not predicted class.
In Figure 1, the naive Bayes classifier marks the words *killed* and *fool* as most relevant for the decision to classify this comment as toxic. Similarly, the SVM classifier with LRP and LIME mark these two words. In contrast to that, the word *killed* is less relevant for the LSTM classifiers (with and without attention). Only the naive Bayes and the SVM classifiers use stemming but not the LSTM classifiers. The stemming collapses *killed* to *kill*. Therefore, our naive Bayes and SVM classifiers cannot distinguish the active form of the verb from other words with the same stem. In this particular context, the non-stemmed word is not toxic. The stemming misleads the classifiers to wrongly explain the toxicity of the comment with this word.
The attention mechanism highlights the words *ignorant* and *fool*. The word *killed* is marked as slightly relevant and all

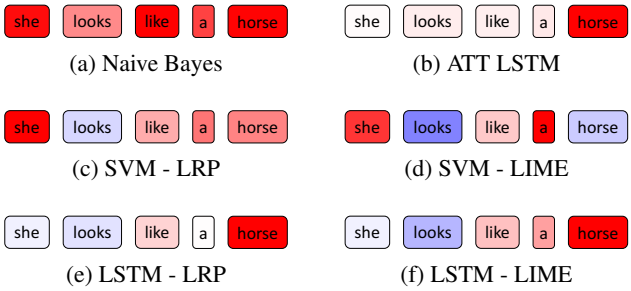


Figure 2: This heatmap visualizes positive (red) and negative (blue) word relevance scores generated by combinations of different classifiers and explanation methods.

other words as irrelevant. This explanation aligns with an explanation a human would give. In general, we find that the attention mechanism gives meaningful explanations for toxic comments. For non-toxic comments, however, its explanations can be misleading. The attention mechanism distributes a relevance score of one among the words — even if there is nothing toxic in the comment. To our surprise, the attention mechanism often marks punctuation as relevant in non-toxic comments.

The basic LSTM approach marks only a few words as relevant, and most words have relevance close to zero. These sparse explanations are suitable for our dataset, as there is typically a small set of toxic words, which explains the toxicity of the entire comment. In Figure 1c to 1f, LIME and LRP assign negative relevance scores to the word *Please*. This negative relevance score means that this word speaks against the toxicity of the comment.

The heatmaps in Figure 2 visualize the word relevance scores of another comment. Only the basic LSTM classifies this short comment correctly. It contains no swear words, but it is still offensive. The negatively connoted association of a person with an animal falls into the category of dehumanizing language. Without the full context, none of the single words explains the toxicity of the comment. Therefore, it is difficult to provide an attribution-based explanation.

4. Evaluation

The following evaluation is three-fold. First, we compare the different classification approaches (naive Bayes, SVM, LSTM, and LSTM with attention mechanism) with regard to their classification performance on the toxic comments dataset. Second, we pair the approaches with attribution-based explanation methods and evaluate the generated explanations based on a word deletion task. The third part of the evaluation uses the explanatory power index (EPI) by Arras et al. (2017).

4.1. Classification Performance

To evaluate the classification performance of the different classifiers, we use a multi-label classification task on the toxic comments dataset. Due to the imbalanced class distribution of this dataset, we refrain from using accuracy as the evaluation metric and instead use precision, recall, and F1-score. Table 2 lists the results on the test set and shows that the naive Bayes baseline is weakest, followed by the SVM approach. The basic LSTM network and the LSTM network with attention mechanism overall achieve similar F1-score with larger differences in the less populated classes *severe_toxic*, *threat*, and *identity_hate*. For the following evaluation of explanation methods, we consider a binary classification task based on the *toxic* class label only. All classifiers achieve their best performance for this most frequent label.

Table 2: Precision (P), Recall (R) and F1-score of the classifiers on the toxic comments dataset (in percent). Bold font indicates best F1-score per class.

Class	Metric	NB	SVM	LSTM	ATT
Toxic	P	69.87	83.22	81.66	84.54
	R	63.89	65.98	68.36	69.74
	F1	66.75	73.60	74.42	**76.43**
Severe Toxic	P	14.45	52.11	56.96	58.33
	R	92.20	18.05	21.95	07.69
	F1	24.98	26.81	**31.69**	13.59
Obscene	P	51.89	85.64	81.09	86.15
	R	75.70	67.57	71.84	67.13
	F1	61.57	75.54	**76.19**	75.46
Threat	P	03.95	72.41	31.43	89.29
	R	59.72	29.17	15.28	35.21
	F1	07.41	41.58	20.56	**50.51**
Insult	P	48.41	78.43	72.67	77.64
	R	75.75	57.82	69.18	59.56
	F1	59.07	66.56	**70.88**	67.40
Identity Hate	P	11.72	64.47	55.36	65.77
	R	73.46	23.22	29.38	49.75
	F1	20.21	34.15	38.39	**56.64**

4.2. Word Deletion Task

We consider a word deletion task to evaluate whether explanation methods correctly identify which input words are most relevant for the classifier's output. It is based on an idea by Arras et al. (2017). The task evaluates whether the words that the explanation points out to be relevant for the classification indeed have a strong influence on it. Each explanation method, therefore, needs to calculate a relevance score for each input word. The word with the highest relevance is deleted, and it is checked whether the model's classification result changes with the perturbed input.

Given the set of true positives (toxic comments that are correctly identified as toxic), we use each explanation method to calculate word relevance scores for each comment. For each method, we then delete the most relevant words from each comment. If the word is indeed relevant for the classifier's decision, the classification most likely changes for the perturbed comment. Step-by-step, we delete more and more words with decreasing relevance scores. An explanation method is considered to provide good relevance scores if the classification changes for a large number of comments after deleting only a few words.

Figure 3 shows how the accuracy quickly drops as more and

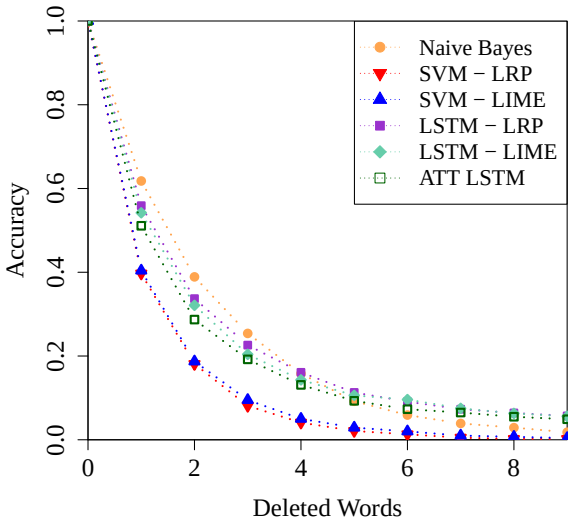


Figure 3: Correct classifications into the *toxic* class change to *non-toxic* if the most relevant input words are deleted. This result shows that the word relevance scores successfully mirror a word's influence on the classification result.

more words are deleted. By deleting four words, more than 80% of the comments that were previously correctly classified as toxic (true positives) are classified as non-toxic. This result confirms that the classifiers detect those words that often constitute the toxicity of a comment (e.g., swear words).

Further, Figure 3 suggests that SVMs provide better explanations than LSTMs. This suggestion is misleading and reveals one limitation of the experiment. Each method starts with its own set of true positives. Therefore each line in the plot corresponds not only to a different explanation method but also to a slightly different dataset. While the overlap of the sets is relatively large, the LSTM network's set of true positives is slightly larger (almost a superset). It also contains some of the more difficult samples of toxic comments, which are correctly classified by the LSTM but misclassified by the naive Bayes approach. One idea to get rid of this problem is to use the intersection of all sets of true positives. The resulting comments are unanimously correctly classified. However, when we further explored this idea, we found that this set is rather small and, more importantly, it contains only the most simple comments — the comments that *all* classifiers detect correctly as toxic.

Still, for those comments that it classifies correctly, the SVM classifier definitely provides the best explanations according to the word deletion experiment. However, the true positives of the LSTM approach also contain comments whose toxicity can only be detected with context. A comment that contains a single swear word is easier to perturb to be classified as non-toxic than a comment that is toxic in its entirety.

4.3. Explanatory Power Index

Arras et al. (2017) propose a three-step approach to quantify the explanatory power of a text classifier with their explanatory power index (EPI). We follow this approach and first calculate one document summary vector per comment in the test set based on each combination of a classifier and an explanation method. The document summary vector is either calculated as a weighted average of the comment's GloVe word embeddings or as the comment's weighted TF-IDF vector representation. We compare a variety of approaches for weighting the words based on word relevance scores.

In the second step, we perform a k-nearest neighbor (kNN) classification on these document summary vectors based on each classifier's predictions. This step is repeated ten times on different random splits of the data and with different values of k. The classification accuracy of the KNN classifier is averaged for each k over the ten runs. The EPI is defined as the maximum achieved classification accuracy. We limit the dataset to all toxic comments and a random sample of non-toxic comments of the same size. This downsampling reduces the data to a balanced set of $4,300$ comments and allows to properly use accuracy as the evaluation metric. Intuitively speaking, the EPI mirrors how good the document summary vectors capture the semantic similarity of documents of the same class by clustering them closer to each other in the high-dimensional vector space.

Table 3 lists the EPI for the different classifiers paired with the respective explanation methods. The results show that weighting a document's bag-of-words vector representation with conditional probabilities from the naive Bayes baseline has the weakest explanatory power. Its performance is followed by the other two baselines: the SVM approach with TF-IDF weights and the basic LSTM approach with averaged GloVe vectors to obtain document summary vectors. The explanatory power of the basic LSTM classifier combined either with LIME or LRP is superior to all other methods. Although the LSTM with attention mechanism achieves slightly better classification results (F1-score of 76.4% vs. 74.4%), the attention weights are not as suited for explanations as word relevance scores generated with LIME or LRP for the basic LSTM network.

Table 3: Explanatory Power Index (EPI) for classifiers and explanation methods. Hyperparameter k denotes the number of nearest neighbors that maximizes the EPI.

Classifier	Explanation Method	EPI	k
Naive Bayes	Conditional Probability	82.29	3
SVM	TF-IDF	87.59	25
	LRP	93.38	19
	LIME	93.14	19
LSTM	GloVe	84.74	15
	LRP	**99.67**	3
	LIME	99.48	9
ATT LSTM	Attention Mechanism	92.04	11

5. Discussion

LIME and LRP achieve similar results in our experiments. However, they strongly differ in their computational costs. The runtime to generate explanations with LIME is about 40 times higher than with LRP. This difference is because

LRP needs only one backpropagation run to propagate the relevance scores from the output layer to the input (word) layer. In contrast to that, LIME requires perturbing a large set of samples. These samples need to come from the local neighborhood of the comment to be explained. Fore example, they need to have many words in common. The more samples are used, the more stable are the explanations.

In the word deletion experiment, LIME has an unfair advantage over the other explainability methods due to the way it is trained. The perturbation in its training process is similar to the perturbation in the word deletion task. Therefore, LIME is tailored to this task.

A downside of the attention mechanism is that it cannot provide class-specific word relevance scores. Strictly speaking, the attention weights — and thus also the derived relevance scores — do not refer to the word level. The weights instead refer to the hidden states in the sequence of LSTM units. The attention mechanism explains which states are most relevant for the network's final output. The activation of a hidden state is the result of processing a subsequence of the input word sequence — regardless of the actual classification output (toxic/non-toxic). The heatmap visualizations in Figure 1b and Figure 2b show that the attention mechanism distributes the relevance only among a few words, more precisely, hidden states. One reason for that is that a single hidden state actually captures information gained from a sequence of input words.

A limitation of attribution-based explanations for offensive language detection seems to be a focus on words that are toxic regardless of the context. This limitation might render them inappropriate for the detection of implicit offensive language. The latter defines offensiveness that is not directly expressed but only arises from the context, uses irony or sarcasm, or can be inferred from metaphors, comparisons, or ascribed properties (Struß et al., 2019).

In the application scenario of content moderation on an online platform, a classifier that achieves slightly worse accuracy might be preferable if it provides explanations. The reason for this trade-off is not only the importance of transparency of the moderation process and acceptance by the user community. Explanations also facilitate the maintenance of a trained classification model. As the topics of online news articles and the corresponding user discussions change daily, adaptation is necessary — also adaptation of machine-learned models.

For example, on one day, an offensive comment might be removed from the platform. However, on the next day, the same comment might be the legitimate center of the discussion because it is a quotation by a well-known politician. In industry applications in general, explanations can support software developers and maintainers to understand machine-learned models and the associated software better.

6. Conclusions

Besides the need for automated offensive language detection, there is also a need for understanding these automated decisions. To this end, we studied explanation methods and compared four different approaches to make offensive language detection explainable: an interpretable machine learning algorithm (naive Bayes), a model-agnostic explanation method (LIME), a model-based explanation method (LRP), and a self-explanatory model (LSTM network with an attention mechanism).

In future work, we plan to generate explanations for users on online discussion platforms. The goal there is to make content moderation more comprehensible by using a fine-grained classifier (insult, threat, profanity, etc.) together with highlighting the most relevant input words as explanations. We also envision either selecting pre-defined text blocks or generating text as explanations and plan to compare these approaches to the explanations that a human moderator would provide. Last but not least, we are working on a journal article as an extended version of this paper (Risch et al., 2020).

7. Bibliographical References

Arras, L., Horn, F., Montavon, G., Müller, K.-R., and Samek, W. (2017). What is relevant in a text document?: An interpretable machine learning approach. *PLOS ONE*, 12(8):1–23.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46.

Carton, S., Mei, Q., and Resnick, P. (2018). Extractive adversarial networks: High-recall explanations for identifying personal attacks in social media posts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3497–3507. ACL.

Chakrabarty, T., Gupta, K., and Muresan, S. (2019). Pay "attention" to your context when classifying abusive language. In *Proceedings of the Workshop on Abusive Language Online (ALW@ACL)*, pages 70–79. ACL.

Došilović, F. K., Brčić, M., and Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215. IEEE.

Gao, L. and Huang, R. (2017). Detecting online hate speech using context aware models. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 260–266. INCOMA Ltd.

Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12:2451–2471.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.

Jain, S. and Wallace, B. C. (2019). Attention is not Explanation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3543–3556. ACL.

Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., Erhan, D., Kim, B., and Dähne, S. (2018). Learning how to explain neural networks: PatternNet and PatternAttribution. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–16.

Kumar, R., Reganti, A. N., Bhatia, A., and Maheshwari, T. (2018). Aggression-annotated Corpus of Hindi-

English Code-mixed Data. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. ELRA.

Monroe, D. (2018). AI, explain yourself. *Communications of the ACM*, 61(11):11–13.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222.

Nguyen, D. (2018). Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1069–1078. ACL.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144.

Risch, J. and Krestel, R. (2018). Delete or not delete? semi-automatic comment moderation for the newsroom. In *Proceedings of the Workshop on Trolling, Aggression and Cyberbullying (TRAC@COLING)*, pages 166–176.

Risch, J., Ruff, R., and Krestel, R. (2020). Explaining offensive language detection. *Journal for Language Technology and Computational Linguistics (JLCL)*, 34(1):1–19.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Workshop Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–8.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. (2015). Striving for simplicity: The all convolutional net. In *Workshop Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–14.

Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., and Klenner, M. (2019). Overview of germeval task 2, 2019 shared task on the identification of offensive language. In *Proceedings of the Conference on Natural Language Processing (KONVENS)*.

Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93.

Wiegreffe, S. and Pinter, Y. (2019). Attention is not not explanation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20. ACL.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1480–1489.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833. Springer.