Julian Risch, Robin Ruff, Ralf Krestel

Explaining Offensive Language Detection

Abstract

Machine learning approaches have proven to be on or even above human-level accuracy for the task of offensive language detection. In contrast to human experts, however, they often lack the capability of giving explanations for their decisions. This article compares four different approaches to make offensive language detection explainable: an interpretable machine learning model (naive Bayes), a model-agnostic explainability method (LIME), a model-based explainability method (LRP), and a self-explanatory model (LSTM with an attention mechanism). Three different classification methods: SVM, naive Bayes, and LSTM are paired with appropriate explanation methods. To this end, we investigate the trade-off between classification performance and explainability of the respective classifiers. We conclude that, with the appropriate explanation methods, the superior classification performance of more complex models is worth the initial lack of explainability.

1 Explainability and Interpretability

Automatic classification of text happens in many different application scenarios. One area where explanations are particularly important is in the context of online discussion moderation since the users who participate in a discussion usually want to know why a certain post was not published or deleted. On the one hand, comment platforms need to consider automatic methods due to the large volume of comments they process every day. On the other hand, these platforms do not want to lose comment readers and writers by seemingly censoring opinions. If humans moderate online discussions, it is desirable to get an explanation of why they classify a user comment as offensive and decide to remove it from the platform. Thereby, to some extent, moderators can be held accountable for their decisions. They cannot randomly remove comments but need to give reasons — otherwise, users would not comprehend the platform's rules and could not act by them.

Machine learning approaches have proven to be on or even above human-level accuracy for the task of offensive language detection (Wulczyn et al., 2017). A variety of shared tasks fosters further improvements of this classification accuracy, e.g., with focuses on hate speech against immigrants and women (Basile et al., 2019), offensive language (Zampieri et al., 2019; Struß et al., 2019), and aggression (Bhattacharya et al., 2020). As automated text classification applications find their way into our society and their decisions affect our lives (Risch & Krestel, 2018), it also becomes crucial that we can trust those systems in the same way that we can trust other humans.

JLCL 2020 – Band 34 (1) – 29–47

Machine-learned models, such as models that detect offensive language, should therefore be comprehensible. The field of Explainable AI (XAI) emerged to address this problem by making models interpretable and/or explainable.

Explainable AI is a young and multidisciplinary research area, ranging from machine learning, data visualization, and human-computer interaction to psychology. Researchers distinguish between interpretability and explainability (Lipton, 2018). Interpretability means to convey a mental model of the algorithm to humans. In other words, if a model is interpretable, humans can grasp how its internals work. In contrast, explainability means to explain individual *predictions* of a model, rather than the full model itself. With an explainable model, humans can comprehend the calculation steps that lead from a particular input to a particular output. On the other hand, interpretability enables developers to understand a model's weaknesses and to improve on them.

Some machine learning algorithms, such as decision trees, logistic regression, and naive Bayes, are interpretable by default. However, with an increasing number of features and sophisticated preprocessing, even these simple models lose their interpretability. More complex, non-linear models, such as neural networks and support vector machines (SVMs) with kernels, achieve better accuracy in some tasks but are not interpretable by default. So it might seem that there is a trade-off between accuracy and interpretability.

Explainability is easier to achieve as it is sufficient to explain only single predictions of a model rather than the model itself. There are explainability methods that are specific to machine learning algorithms (model-based) and methods that can be applied to any model (model-agnostic, post-hoc). With many explained predictions of a black-box model, a human's mental model of the algorithm improves. Thereby, explainability can lead to interpretability.

Recently, there is research that is contrary to post-hoc explanation methods. For example, Rudin (2019) states that the focus should be on creating inherently interpretable models rather than retrospectively explaining black-box models. With the General Data Protection Regulation (GDPR)[1] specifying the right to explanations, developing explainable AI systems is inevitable, and we expect the field of Explainable AI to grow in the future. Especially the highly complex neural networks with millions of parameters raise the bar for many natural language processing tasks significantly. At the same time, these models are non-interpretable black boxes. We see a need to make especially these most complex models explainable to ensure trust in them by humans.

In our work, we train a naive Bayes classifier, an SVM, and recurrent neural network models on a dataset of toxic comments. We examine the explanation methods Layer-wise Relevance Propagation (LRP) and Local Interpretable Model-agnostic Explanations (LIME), but also attention layers. Thereby, our study covers a model-based method, a model-agnostic method, and a self-explanatory model. The naive Bayes classifier serves as a baseline. For the evaluation of the explanation methods, we use a word deletion task, the explanatory power index, and t-SNE projections of document vector representations. We discuss the results and find that the explainability methods LRP

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

and LIME provide explainability beyond the limits of interpretable machine learning algorithms, such as naive Bayes.

Contributions In summary, with the present article, we make the following contributions: First, we provide an overview of explainability methods that can be applied to offensive language detection. Second, we implement a variety of such methods and compare them in different experiments. Finally, we interpret the results, discuss the strengths and weaknesses of the methods, and summarize implications for future work.

Article Outline The remainder of this article is structured as follows: In Section 2, we describe related work on explainability methods and set our work into its context. Section 3 focuses on those methods that we implement for this study and how we train the underlying models for offensive language detection. We evaluate the different methods and discuss the results in Section 4 and 5, before we conclude in Section 6.

2 Related Work

There are two principal ways to achieve explainability: either by using interpretable classifiers or by extending non-interpretable classifiers with explainability methods. The terms explainability and interpretability have no standard definitions in the context of machine learning. When they are not used interchangeably, the distinction is that explainability refers to comprehending individual predictions, whereas interpretability refers to comprehending the decision function (Došilović et al., 2018; Monroe, 2018; Montavon et al., 2017). The terms *local explainability/interpretability* and *global explainability/interpretability* are used to describe this difference (Mohseni et al., 2018; Ribeiro et al., 2016). For the lack of consensus in terminology, we define the terms for this article:

- 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.

For example, in the special case of a text classifier, an attribution-based explanation method might output one score per input feature, e.g., input word. The scores denote how much each input feature contributes to the classifier's decision. Note that interpretability comprises explainability. To this end, interpretability can be derived from explainability by agglomerating explanations. Ribeiro et al. (2016) propose an algorithm to select inputs so that the explanations of the decisions to those inputs give an interpretation of the model. Depending on the domain context of a model, other explanation forms are possible. For example, there are hierarchical explanations, which explain sentiment analysis decisions by considering word interactions (Singh et al., 2019; Tsang et al., 2018; Murdoch et al., 2018).

2.1 Interpretable Classifiers

Simple models are interpretable without any special methods and abstraction because they align with human intuitions. The most popular interpretable models are decision trees since they can easily be visualized and consist of a set of structured decision rules. Explaining a decision tree's prediction is as simple as following the branches that correspond to the input. The most relevant features are closer to the root of the tree. Thereby, the degree of abstraction for the interpretation can be increased simply by pruning the tree.

Another class of interpretable models is based on discrete probabilities. The naive Bayes classifier is interpretable because it uses simple calculations with discrete conditional probabilities. These probabilities can be interpreted as a contribution to the decision made by the classifier. We use this approach as a baseline in our evaluation.

2.2 Sensitivity Analysis and Shapley Values

Sensitivity analysis and Shapley values are two mathematical concepts behind most explainability methods. Sensitivity analysis figures out how sensitive the output $f(x)$ is to a change in the input x. For an infinitesimal change in x, this can be expressed as the gradient ∇f of the decision function f evaluated for the input x. Baehrens et al. (2010) define $-\nabla f(x)$ as the explanation vector. Simonyan et al. (2014) apply sensitivity analysis to explain image classifications made by convolutional neural networks (CNNs) by using the backpropagation algorithm to obtain the gradient. A simple variant of sensitivity analysis that leads to more specific explanations for image classification is gradients multiplied by input (Shrikumar et al., 2017). Explanations by sensitivity analysis cannot be interpreted as: "What input makes the prediction turn out positive?", but rather as: "How to change the input to make the prediction more positive?".

Shapley values have their origin in coalition game theory. They were proposed to assign each player of a coalition game the contribution he or she makes to the overall outcome of the game (Shapley, 1953). The axioms for Shapley values are also desirable properties in the context of explaining a classifier's decision:

1. **Efficiency** The explanation reflects the outcome of the classifier $f(x)$.
2. **Symmetry** Two features that add the same value to the decision $f(x)$ should be equally relevant.
3. **Additivity** If there are multiple decision functions in an ensemble, the final relevance score should match the sum of the scores of the individual functions.
4. **Dummy Player** A feature that does not change the outcome of the classifier should have no relevance.

Shapley values are not used in practice because of their computational costs. Even if feature interactions are neglected, it is infeasible to do the necessary calculations, especially with high-dimensional data, such as word embeddings. Despite not being used often in its pure form, the concept of Shapley values is still relevant. Lundberg

and Lee (2017) propose the SHAP framework inspired by Shapley values and show that other explainability methods are approximations of SHAP.

2.3 LRP and LIME

With layer-wise relevance propagation (LRP), Bach et al. (2015) bring the idea of the efficiency axiom of the Shapley values to deep neural networks. However, propagating the output $f(x)$ directly to the input features is complicated for complex decision functions that contain feature interactions and non-linearities, such as those modeled with neural networks. The LRP method makes use of the layered structure of neural networks to break this problem down by distributing the relevance stepwise for each layer in the network. The layer-wise relevance propagation concept defines the constraint that the summed-up relevance scores for each layer are conserved throughout the propagation. This constraint is called *relevance conservation property*.

Ribeiro et al. (2016) propose Local Interpretable Model-agnostic Explanations (LIME). To explain a decision $f(x)$, LIME approximates the local neighborhood of $f(x)$ with an interpretable classifier $g : \{0,1\}^d \rightarrow \mathbb{R}$ that serves as an explanation. Remark that g and f do not have the same domain. The domain of g is a binary space with the same dimension as the feature space. Therefore the input to g does only capture the absence or presence of a feature. LIME considers two aspects to choose the best explanation g for $f(x)$. First, g needs to be a good local approximation of f in the local neighborhood of x. Second, the complexity of g should be low to ensure that g is interpretable. To this end, the best explanation for a decision $f(x)$ is the model g that minimizes the unfaithfulness and the complexity of g.

2.4 Other Explainability Methods

Related work on explainability typically discusses image classification. CNNs are very prominent in this domain. Therefore, many CNN-based explainability methods have been developed. One of the first explainability methods for CNNs is DeConvNet (Zeiler & Fergus, 2014). This approach tries to explain decisions by inverting convolution, ReLU operations, and pooling. Applying sensitivity analysis to CNNs by using back-propagation to obtain the gradient leads to similar explanations (Simonyan et al., 2014). Springenberg et al. (2015) describe the differences between DeConvNet and Sensitivity analysis in the aspect of ReLU operations and propose a combination of the approaches called *guided backpropagation*. Kindermans et al. (2018) argue that splitting the input into a signal and a distractor part can lead to clearer explanations. They compare their methods to Sensitivity analysis, DeConvNet, Guided Backpropagation, and LRP.

Similar to LRP is DeepLIFT (Shrikumar et al., 2017). It also backpropagates relevance through neural network layers and complies with the relevance conservation property. Instead of starting with a relevance score that equals the output of the last layer neuron, DeepLift uses the difference to a reference point as an initial relevance score. The explainability method CAM (Class Activation Mapping), proposed by Zhou et al. (2016), uses a special CNN architecture to learn what parts of an image are

important for the decision, by considering the outputs of the last convolutional layer. GradCAM extends CAM by combining it with Sensitivity analysis and thereby avoids to retrain the network for explanations, as it is the case with CAM (Selvaraju et al., 2017). The concept of Taylor-type Decomposition was proposed alongside LRP (Bach et al., 2015) and later refined (Montavon et al., 2017). Instead of using relevance messages to propagate the relevance through the layers, first-order Taylor expansions are used to distribute the relevance scores to the next layer. Sundararajan et al. (2016) introduce *integrated gradients*, a way to fulfill the efficiency axiom of Shapley values by integrating over the gradients with respect to modified (counterfactual) inputs.

Murdoch and Szlam (2017) analyze the hidden cell states of an LSTM to construct interpretable rule-based classifiers. This method, called Cell Decomposition, is an explanation method specific to LSTMs. The same authors propose Contextual Decomposition, which does not only explain decisions with relevance scores to single words but also explains phrases and word interactions (Murdoch et al., 2018).

Related work rarely focuses on explanations for text classification. One publication compares human and automatic evaluation of explanation methods for text classification (Nguyen, 2018) and another one describes the application of an attention-based explanation method to a dataset of personal attacks (Carton et al., 2018). Earlier results of our research on explanations for offensive language classification are published in a short paper (Risch, Ruff, & Krestel, 2020).

2.5 Taxonomy of Explainability Methods

We focus on explainability methods that use feature relevance explanations. The first aspect in which explainability methods can differ is whether they use information about the model's structure or not. Layer-wise relevance propagation, for example, is designed to explain decisions made by neural networks and SVMs, as it uses the layered structure and the activation values of hidden layer neurons. Hence LRP is a *model-based* explainability method. Opposed to that, LIME operates on black-box models and does only use the models' input-output pairs to explain decisions. Thus LIME is a *model-agnostic* (or post-hoc) explainability method. Model-agnostic explainability methods often use sampling to approximate the model with another interpretable surrogate model.

Model-based methods can further be distinguished by the approach they are taking to assign relevance scores. Many methods rely on gradients to explain a decision. The simplest of those methods is sensitivity analysis. Guided Backpropagation, integrated gradients, and gradients $\times$ input extend this concept. Layer-wise relevance propagation and DeepLIFT have in common that they make use of the efficiency axiom of Shapley values and redistribute a fixed relevance score onto the features. Other methods, like DeConvNet and Cell Decomposition, are very specific to the machine learning algorithms. There are also so-called self-explanatory machine learning algorithms which inherently provide explanations as a side effect of the decision-making process. An example of such a self-explanatory machine learning algorithm is LSTM with an attention mechanism.

3 Explaining Offensive Language Detection

In order to be of practical relevance, automatic offensive language detection tools need to be trusted by users. Trust can only be established if the automatic decisions can be convincingly explained if needed. We implemented several different algorithms for offensive language detection and combined them with different explanation methods. We published our python code for all classifiers, a web application to visualize the explanations, and the training and evaluation procedures on GitHub.[2]

3.1 Classifiers and Explainability Methods

As a baseline, we implement a multinomial naive Bayes text classifier and add explainability based on the inherent conditional probabilities. This classifier is an example of an interpretable machine learning model. Second, we implement an explainable SVM classifier.[3] For multi-class classifications, we use the one-against-all scheme. We generate explanations for SVM decisions with the model-based explainability method LRP and the model-agnostic explainability method LIME. Last but not least, we implement an LSTM with an attention mechanism, which is an example of a self-explanatory model.

3.2 Dataset

There is a variety of datasets annotated for the detection of hate speech (Gao & Huang, 2017), racism/sexism (Waseem & Hovy, 2016) or offensive/aggressive/abusive language (Struß et al., 2019; Kumar et al., 2018). However, most of them are comparably small because of the immense labeling effort. In this article, we use one of the largest annotated datasets in this field, which contains more than 220,000 comments. Google Jigsaw released this dataset as part of a Kaggle challenge on toxic comment classification.[4] It comprises user discussions from talk pages of the English Wikipedia, where each comment can be labeled as *toxic*, *severe toxic*, *insult*, *threat*, *obscene* or *identity hate* (non-exclusive labels). Table 1 shows that the class distribution is strongly imbalanced.

3.3 Training Procedure

The toxic comments dataset represents a multi-label classification problem. Since there are six labels in the dataset, we can think of the naive Bayes and SVM classifiers as six independent binary naive Bayes classifiers, respectively, six independent binary SVMs. The LSTMs have a slightly different architecture for multi-label problems. All labels share the same LSTM layer, but each label has its own independent fully-connected layer

[2] https://hpi.de/naumann/projects/repeatability/text-mining.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[4] https://www.kaggle.com/c/jigsawtoxic-comment-classification-challenge

Table 1: The class distribution of the dataset is strongly imbalanced. The rarest label *threat* is assigned to only 0.3% of the samples.

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

at the last LSTM output. For the Attention LSTM, each label has its own independent attention layer with the following fully-connected layer.

We train the explainable LSTM with TensorFlow[5] and use the *LRP for LSTM*[6] implementation to explain decisions with LRP. We use an Attention LSTM by Yang et al. (2016) with the difference that we only use an attention layer on the word level and no additional sentence level. We choose the regularization term $C = 0.6$ for the SVM. LSTM and Attention LSTM both have a maximum input length of 250, use a 50-dimensional hidden layer for LSTM cells, and are trained with the Adam optimizer for five respectively three epochs. We train custom GloVe word vectors on the corpus of the training set and the unlabeled comments included in the dataset.

4 Evaluation

First, we compare the classification performance of different approaches for offensive language detection. We then describe the experimental setup for the evaluation of their respective explanations. Finally, we discuss the results.

4.1 Classification Performance

Table 2 presents precision, recall, and F1-score of the trained models on the test set. Both LSTM architectures outperform SVM, which in turn outperforms the naive Bayes baseline. We are unable to get good results for the labels *severe_toxic*, *threat*, and *identity_hate* because each of them makes up less than 1% of the dataset. For the evaluation of explanations, we only focus on the *toxic* label as the classifiers perform best on this label.

[5] `https://www.tensorflow.org/`
[6] `https://github.com/ArrasL/LRP_for_LSTM/`

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

Class Label	Metric	Naive Bayes	SVM	LSTM	Att. LSTM
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 Examples of Heatmap Visualization

Explanations by naive Bayes and Attention LSTMs only assign positive relevance scores between 0 and 1. Relevance scores of naive Bayes explanations are probabilities and relevance scores of Attention LSTM explanations are results of a normalizing softmax function. In contrast, explanations by LIME and LRP contain relevance scores that are unbounded and can also be negative. Attention LSTM explanations are the only explanations that are class-independent. Other explainability methods can explain any class, even if the classifier did not predict that class.

The comment in Figure 1 is correctly classified as toxic by both LSTM architectures. The naive Bayes classifier and the SVM classify it as non-toxic.

Figure 1a shows that the naive Bayes classifier explains the toxicity of the comment by marking the word *fool*. The word *killed* is stemmed to *kill* and, therefore, arguably marked also as an explanation, although it is not toxic in this context. Rarely occurring words, such as *Sarsa*, *sirhind*, and *wazir*, are also marked as toxic. The effect of relatively high relevance scores for words that are equally distributed among all classes is amplified in the binary classification case. For a word w that appears with equal

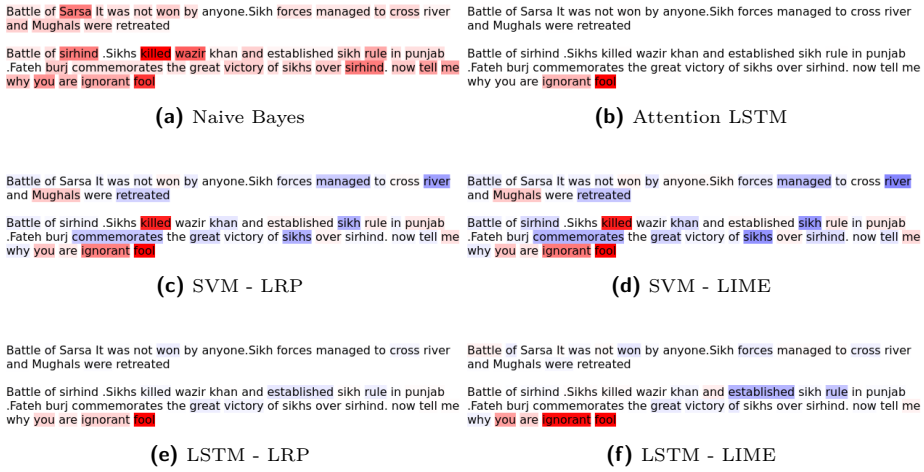


Figure 1: Heatmap visualization of the explanations by the different classifiers and explainability methods. For LRP and LIME, red indicates positive and blue indicates negative relevance.

frequency in both classes c, the relevance of a word is $P(c|w) \approx 0.5$. Together with the unbalanced dataset, this leads to problems for rarely occurring words because the used Laplace smoothing becomes more significant. This smoothing causes high relevance scores for the words *Sarsa*, *sirhind*, and *wazir*. Note that this example comment is labeled as not toxic by the naive Bayes classifier despite the high relevance scores of many words.

Figure 1b shows the explanation generated by the Attention LSTM. The words *fool* and *ignorant* are marked as relevant, and all other words as irrelevant. This explanation aligns with an explanation a human would give. The explanation does not mark *killed* as toxic (in contrast to the naive Bayes classifier). There are two reasons for that. Attention LSTMs do not use stemming (*killed* is considered less toxic than *kill*), and they take into account surrounding words (context awareness).

For toxic comments, we generally observe meaningful explanations by the Attention LSTM. However, for non-toxic comments, Attention LSTMs give misleading explanations. Note that the importance weights that we use as word relevance scores are the result of a softmax function. As a consequence, the Attention LSTM necessarily distributes a relevance score of one among the words — even if there are no toxic elements in the comment. We find that Attention LSTMs often mark punctuation as relevant for non-toxic comments.

Figure 1c and Figure 1d show that LRP and LIME generate almost identical explanations. The toxic words *ignorant* and *fool* are detected by the SVM classifier. The

(a) LSTM - LRP **(b)** LSTM - LIME

Figure 2: Heatmap visualization of the explanations made by LRP and LIME for a contextually toxic comment classified by an LSTM neural network.

word *killed* is also marked as toxic because of stemming. Explanations for non-toxic comments are also very similar for the SVM classifier. The maximum relevance scores for non-toxic comments are much smaller than we would expect.

Explanations for LSTM have an unbalanced relevance distribution among the words. Few words have high absolute relevance and most words have relevance close to zero. These sparse explanations are desirable in the context of this dataset, as there is typically a small set of words that explain the toxicity of a comment.

The LRP explanation is similar to the Attention LSTM explanation, but also includes the word *you*. LIME rates the term *ignorant* as much more toxic than LRP does. We find that LIME often assigns larger negative relevance scores. Explanations for non-toxic comments do not suffer from the problem with the Attention LSTM, as they have much smaller relevance scores overall.

In line with van Aken et al. (2018), we find the labeling of the dataset to be inconsistent. For many comments that are misclassified as toxic, the explanations indeed mark toxic words. Note that the naive Bayes classifier and the SVM label the example comment in Figure 1 as non-toxic, but the explanations highlight the toxic words. The labeling quality of the dataset makes it hard for the classifiers to learn the correct toxicity threshold.

Figure 2 shows a short toxic comment that contains no swear words. The LSTM without attention mechanism is the only classifier that correctly labels this comment as toxic. Without context, none of the words in the comment would be considered toxic on its own. It is therefore difficult to explain this example with attribution-based explanations.

Besides this qualitative evaluation by visualizing the explanations, quantitative, objective methods would be desirable. Unfortunately, evaluating the quality of an explanation is a hard task. Even for human assessors, deciding which explanation is good or bad is very hard and often undecidable. The fact, that the explanations (should) depend on the classification result (which might be wrong), makes the evaluation even more complicated. Nevertheless, we deploy two methods proposed in the literature to automatically and objectively evaluate the generated explanations: word deletion and explanatory power index.

4.3 Word Deletion Task

A good explanation for a text classification characterizes that the words with the highest relevance scores have the most impact on the classification. Therefore deleting relevant

words from a text should lead to a significant difference in the classification outcome. The word deletion evaluation measure builds on this premise (Arras et al., 2017). To evaluate a classifier and its explanations with word deletion, we generate explanations for all comments that are correctly classified as toxic (true positives). Consequently, the accuracy on this subset of comments is 100%. We then successively delete words with the highest relevance scores from each text and measure the accuracies of the modified texts at each step. For good explanations, the accuracies should decrease rapidly within the first few word deletions.

And indeed, the accuracy quickly drops in our experiments because only a few words often constitute the toxicity of a comment (e.g., swear words). For all classifiers, more than 80% of the toxic comments could be modified to be not toxic, by deleting only four words. This large number indicates that all classifiers pick up swear words, as those are the explanations for most of the toxic comments. For toxic comments without swear words, the word context is often important, which is the reason for the good performance of LSTMs.

Figure 3 suggests that SVMs give the best explanations according to the word deletion evaluation. This suggestion is misleading because we start for each classifier with its individual subset of true positives: the comments that were correctly labeled as toxic by that classifier. For LSTMs, this subset also contains comments that can only be detected as toxic with word context. It is harder to modify those comments with a few word deletions to be classified as not toxic than it is for comments with a single swear word.

There is no good alternative to using the individual sets of true positives for the evaluation of the explanations for each classifier. In our scenario, the different sets of true positives have a large overlap, which reduces the problem. It is not the case that each classifier is evaluated on entirely different data but rather on slightly different data. We explored the idea of using the intersection of all sets of true positives. However, this approach drastically reduces the size of the dataset for evaluation, and it implicates that the remaining set contains the most simple comments — the ones that *all* classifiers detected correctly as toxic.

4.4 Explanatory Power Index

Arras et al. (2017) propose the Explanatory Power Index (EPI) to evaluate explanations for text categorizations. The method uses an explanation and the corresponding input representation (TFIDF, GloVe word vectors) of a text and combines them to a document summary vector. To obtain this vector, each word in the input representation is scaled by the assigned relevance. Relevant words are emphasized, and irrelevant words are weakened. In the vector space of all input representations, the document summary vectors form clusters of semantically similar texts.

Better explanations lead to better document summary vectors and, therefore, to clearer clusters. The cluster formation can be quantified by the accuracy of a k-nearest neighbor (kNN) classifier that is trained and evaluated on multiple random data-splits

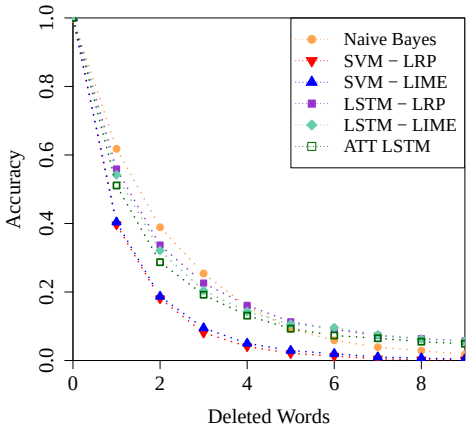


Figure 3: Word deletion experiment for the toxic comments dataset.

of the document summary vectors. The EPI is defined as the mean evaluation accuracy by the kNN classifiers on random data-splits. We use ten splits in our experiments. EPI is decoupled from the predictive power of a classifier since the kNN algorithm is trained and evaluated on the predicted classes for each classifier and not the true classes.

Remark that each entry in a TFIDF vector represents a word. Simply by multiplying each word's vector with the relevance assigned to that word by the explanation, we obtain the document summary vector. In the case of GloVe word vectors, a matrix represents a document, where each row represents a word. Each row vector gets multiplied by the relevance of the corresponding word. In a second step, all row vectors get summed up to obtain the document summary vector. Note that the document summary vector has the same dimension as the word vectors.

EPI uses the accuracy of the kNN classification. To properly use accuracy as a metric, we balance the dataset by downsampling the majority class. We use all toxic comments and randomly sample the same number of non-toxic comments. For each approach, the hyperparameter k is set so that the accuracy (and therefore the EPI) is maximized. Using the baseline representations TFIDF and GloVe word vectors, the kNN algorithm can already distinguish toxic comments from non-toxic comments with high accuracy. The EPI for naive Bayes explanations is worse than for TFIDF. Explanations by naive Bayes often assign high relevance scores to rarely occurring words, which results in the low EPI score. Figure 4c shows that these explanations lead to cluster formations of document summary vectors, but the resulting clusters are not homogeneous.

The explanation methods LIME and LRP have similar EPI scores for the SVM and LSTM classifiers. Figure 4f and Figure 4g confirm these high EPI scores by showing a clear separation of toxic and non-toxic comments into two large clusters. In general, the t-SNE projections of the document summary vectors in Figure 4 suggest that

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

Classifiers	Explanation	EPI	k
Naive Bayes	Probabilistic	82.29	3
SVM	TFIDF	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	92.04	11

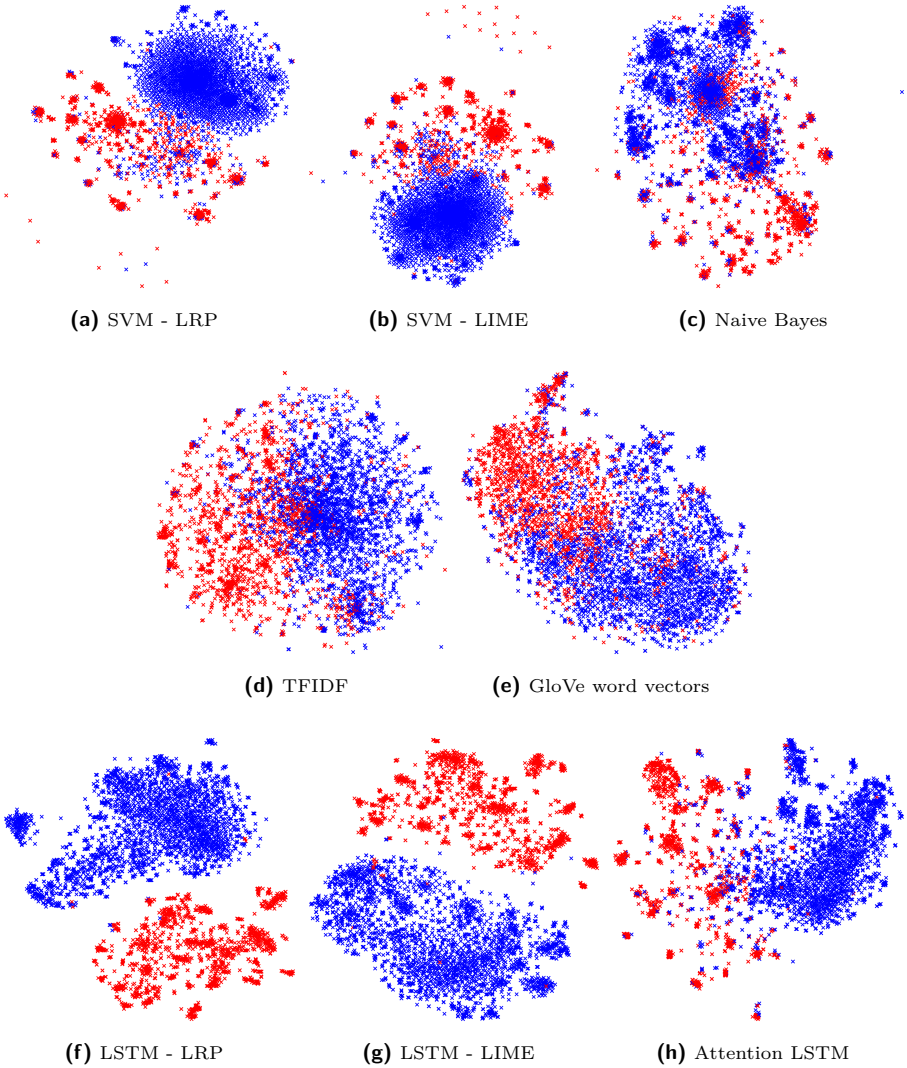
there are multiple clusters of toxic comments. Therefore, document summary vectors could be used to classify and analyze more fine-grained subclasses of toxic comments. The clusters of toxic document summary vectors of the Attention LSTM are denser. However, the separation between the two classes is not as clear as the vectors of the LSTM without an attention mechanism.

5 Discussion

Word deletion and EPI both define quantitative measures to rate explanations, but it is hard to measure the quality of explanations. We defined explainability as the ability to explain a decision of a model in understandable human terms. However, an explanation that aligns with human intuition does not necessarily need to mirror what the model actually is doing. So explainability can only be measured qualitatively within an application context and a target user group. Because our evaluation of explanations is detached from application context and has no target user group, it is hard to rate explanations and explainability methods qualitatively.

The model-agnostic property of LIME comes at the cost of a large number of computations. To achieve stable explanation results with LIME, many perturbed samples need to be classified first. Opposed to that, LRP does a single relevance backpropagation for each explanation. In our experiments, LIME takes up to 40 times longer for explanations than LRP.

The idea to occlude parts of the input and to measure the difference of the output can be generalized beyond text classification and is also used by LIME to generate explanations. Note that LIME is therefore tailored to the word deletion task and might have an unfair advantage in comparison to other explainability methods. For the linear SVM model, LRP and LIME achieve similar results. For more complex decision



(a) SVM - LRP **(b)** SVM - LIME **(c)** Naive Bayes

(d) TFIDF **(e)** GloVe word vectors

(f) LSTM - LRP **(g)** LSTM - LIME **(h)** Attention LSTM

Figure 4: t-SNE projections of the document summary vectors for each explanation method. For reference, Figures 4d and 4e show t-SNE projections of the TFIDF vectors and GloVe word vectors. Red and blue color mark toxic, respectively, non-toxic comments.

functions, such as those of non-linear LSTMs, the explanations by LRP and LIME differ considerably. All methods by far outperform the interpretable naive Bayes classifier.

The explanations of the self-explanatory LSTM with an attention mechanism have some undesirable characteristics. First, the attention mechanism only explains which words are relevant for a prediction in general (similar to sorting out stop words). However, the relevance scores of the words do not depend on the predicted class. Second, we find that the attention mechanism typically marks only a small set of words as relevant, while all other words are assigned a relevance score close to zero. The attention mechanism was not designed to achieve explainability. We suppose that slight modifications could eliminate the undesirable characteristics. For example, we imagine a hybrid explainability method that uses LRP for the fully-connected layer and the relevance scores of the attention mechanism.

6 Conclusions and Future Work

In this article, we compared four different approaches to make offensive language detection explainable: an interpretable machine learning algorithm (naive Bayes), a model-agnostic explainability method (LIME), a model-based explainability method (LRP), and a self-explanatory model (LSTM with an attention mechanism). We found that LRP and LIME achieve explainability beyond the limits of interpretable algorithms without giving up their superior predictive power.

The model-agnostic method LIME and the model-based method LRP differ mostly in the way they handle negative relevance scores for simple linear models. The attention mechanism of the LSTM cannot provide competitive explanations, which is not surprising, since it was not designed for this task in the first place. However, we assume that the explanatory power of the attention mechanism could be improved by tailoring it to the task of giving explanations.

Last but not least, we find that it is difficult to explain the toxicity of a comment if none of the single words is considered toxic without context. In this case, which includes implicit offensive language, attribution-based explanations fail. Therefore, we see other types of explainability as a promising direction for future work.

References

Arras, L., Horn, F., Montavon, G., Müller, K.-R., & 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., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, *10*(7), 1-46.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & Müller, K. (2010). How to explain individual classification decisions. *Journal of Machine Learning Research*, *11*, 1803–1831.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F. M. R., ... Sanguinetti, M. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval@NAACL)* (pp. 54–63).

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., ... Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression. *arXiv e-prints*, arXiv:2003.07428.

Carton, S., Mei, Q., & 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)* (pp. 3497–3507).

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

Gao, L., & 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)* (pp. 260–266).

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

Kumar, R., Reganti, A. N., Bhatia, A., & Maheshwari, T. (2018). Aggression-annotated Corpus of Hindi-English Code-mixed Data. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Lipton, Z. C. (2018). The mythos of model interpretability. *Communications of the ACM*, *61*(10), 36–43.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 4768–4777). USA.

Mohseni, S., Zarei, N., & Ragan, E. D. (2018). A Survey of Evaluation Methods and Measures for Interpretable Machine Learning. *arXiv e-prints*, arXiv:1811.11839.

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

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

Murdoch, W. J., Liu, P. J., & Yu, B. (2018). Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–15).

Murdoch, W. J., & Szlam, A. (2017). Automatic Rule Extraction from Long Short Term Memory Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–12).

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) (pp. 1069–1078).

Ribeiro, M. T., Singh, S., & 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)* (pp. 1135–1144).

Risch, J., & 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)* (p. 166-176).

Risch, J., Ruff, R., & Krestel, R. (2020). Offensive language detection explained. In *Proceedings of the Workshop on Trolling, Aggression and Cyberbullying (TRAC@LREC)* (p. 137-143).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*, 206–215.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the International Conference on Computer Vision (ICCV)* (p. 618-626).

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, *2*(28), 307–317.

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 3145–3153).

Simonyan, K., Vedaldi, A., & 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)* (pp. 1–8).

Singh, C., Murdoch, W. J., & Yu, B. (2019). Hierarchical interpretations for neural network predictions. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–11).

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

Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., & 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)* (pp. 354–365).

Sundararajan, M., Taly, A., & Yan, Q. (2016). Gradients of Counterfactuals. *arXiv e-prints*, arXiv:1611.02639.

Tsang, M., Sun, Y., Ren, D., & Liu, Y. (2018). Can I trust you more? Model-Agnostic Hierarchical Explanations. *arXiv e-prints*, arXiv:1812.04801.

van Aken, B., Risch, J., Krestel, R., & Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the Workshop on Abusive Language Online (ALW)* (pp. 33–42).

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

Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the International Conference on World Wide Web (WWW)* (pp. 1391–1399).

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & 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)* (pp. 1480–1489).

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the International Workshop on Semantic Evaluation (SemEval@NAACL)* (pp. 75–86).

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

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2921–2929).