Machine-annotated Rationales: Faithfully Explaining Text Classification

Elize Herrewijnen[1,2], **Dong Nguyen**[1], **Jelte Mense**[1,2], **Floris Bex**[1]
[1] Utrecht University, Utrecht, The Netherlands
[2] Dutch National Police, The Netherlands
e.herrewijnen@uu.nl, d.p.nguyen@uu.nl, j.p.mense@uu.nl, F.J.Bex@uu.nl

Abstract

We propose an approach to faithfully explaining text classification models, using a specifically designed neural network to find explanations in the form of machine-annotated rationales during the prediction process. This results in faithful explanations that are similar to human-annotated rationales, while not requiring human explanation examples during training. The quality of found explanations is measured on faithfulness, quantitative similarity to human explanations, and through a user evaluation.

1 Introduction

Explainable artificial intelligence (XAI) is concerned with explaining the behavior of AI systems (e.g. machine learning (ML) models). Explaining and understanding black box models, like deep neural networks, is a difficult task (Miller, Howe, and Sonenberg 2017; Kindermans et al. 2019). Often, post-hoc explanations are constructed after the prediction has been made based on just the input and output of a model. Such post-hoc explanations might be feasible or plausible, but they are not *faithful explanations* in that they correctly reflect the exact inner workings of a model (Jacovi and Goldberg 2020; Rudin 2019). Particularly in sensitive work fields like the legal and medical domain (Berk et al. 2018; Tjoa and Guan 2020), we need faithful explanations to give us meaningful agency – the explanations should not just allow us to passively understand a model or decision, but also to actively challenge, debug and change critical ML models (Jacovi and Goldberg 2020; Lipton 2018).

We focus on explaining text classification models. A text can contain words or (sub)sentences that form human-understandable natural language explanations for a classification, called *rationales* (Ehsan et al. 2019). In deciding if documents are positive or negative for instance, parts of the text can be used as rationales for the prediction of the document class. The sentence "I simply hated it" in a review would support a prediction that the review is negative, and this sentence consequently can be used as a rationale supporting this prediction.

When a human classifies a text and annotates parts of text to support the classification, we talk about *annotator rationales* (Zaidan, Eisner, and Piatko 2007). The use of annotator rationales in AI has proved to be useful in both explaining and improving classification accuracy of ML models, (Ehsan et al. 2019; Zaidan, Eisner, and Piatko 2007; Zaidan and Eisner 2008; Zhang, Marshall, and Wallace 2016; Bao et al. 2018). Gathering annotator rationales however requires humans to make and annotate (large numbers of) classifications, and these resources are not always available.

ML models can construct machine-generated rationales from annotator rationales (Zhang, Marshall, and Wallace 2016; Yessenalina, Choi, and Cardie 2010; Lei, Barzilay, and Jaakkola 2016; Robnik-Šikonja and Kononenko 2008). Usually, these machine-generated rationales are generated independently from the classification task, and do not explain a model's reasoning for a classification. Furthermore, human example rationales are often required when training the model. In this work, we propose a novel method for constructing machine-generated rationales that *faithfully* explain text classification tasks without requiring human examples. We extract relevant parts of the input *while classifying* text, that form an explanation for that specific classification. We call these explanations machine-annotated rationales (MaRs), as they are annotated (sub)sentences by the model that explain a classification. Our model does not post-hoc explain using human examples, but finds faithful rationales without learning from annotator rationales, during the prediction process. These machine-annotated rationales can be used 1) as explanations for the model's predictions, 2) to gain insight in the model's inner workings. We use the term 'inner workings' to describe the complete algorithmic process that was executed by the ML model to transform input to output.

We evaluate our machine-annotated rationales on three different aspects: Faithfulness is measured using the comprehensiveness and sufficiency metrics (DeYoung et al. 2020). Quantitative similarity to annotator rationales is measured using set theory. A user evaluation is used to find out whether machine-annotated rationales are useful for human users.

This work is outlined as follows: Related work is discussed in Section 2. This is followed by a description of the dataset and prepossessing steps in Section 3 and Section 4. Then, in Section 5, we describe two implementations of the model. The evaluations metrics and results are discussed in

Section 6. We conclude our work in Section 7.

2 Related work

Different approaches to generating explanations for text classification model have been proposed in literature.

Input eSrasure Determining which parts of the input are relevant for a classification can be done by removing parts and measuring the effect on the classification. This method is called erasure (Li, Monroe, and Jurafsky 2016). For this approach, it is assumed that different parts in the input are independent of each other and influence a prediction. This assumption is referred to as the Linearity Assumption in literature (Jacovi and Goldberg 2020). The Linearity Assumption can be applied to find or validate ML model explanations: Robnik-Šikonja and Kononenko (2008) find explanations for predictions by removing words from the input and measuring the effect on the prediction. DeYoung et al. (2020) evaluate the faithfulness of found explanations by removing explanations from the input and measuring their influence on the prediction.

Saliency methods and the attention mechanism Saliency methods can be used to find explanations, but may be unfaithful. Kindermans et al. (2019) show that saliency methods, where insight is gathered in what features in the input played an important role in a model's prediction, are not always consistent in faithfulness. Such methods can nevertheless help gain intuitions about the workings of neural networks (Kindermans et al. 2019). The attention mechanism, introduced by Bahdanau, Cho, and Bengio (2015), is based on the idea that often only part of the input is relevant for a prediction. The attention mechanism assigns weights to parts of the input to compute a representation, and the parts receiving high weights are sometimes used to explain predictions. However, whether these parts of the input are (faithful) explanation for the prediction, remains undetermined (Jain and Wallace 2019; Wiegreffe and Pinter 2019).

Annotator rationales Zaidan, Eisner, and Piatko (2007) propose explanations in the form of (sub)sentences annotated by humans, called annotator rationales. These annotator rationales are then used to improve classification accuracy (Zaidan, Eisner, and Piatko 2007; Zaidan and Eisner 2008). Teaching a CNN to first recognize rationale sentences using human examples, and then exploit these sentences, increases accuracy and provides explanations (Zhang, Marshall, and Wallace 2016). Bao et al. (2018) use annotator rationales to guide a high-quality attention mechanism, without having to train the model on a large dataset. Annotator rationales can be seen as human attention and compared to machine attention mechanisms, which show significant similarity (Sen et al. 2020). ML models can generate rationales for polarity classification by determining the polarity of sentences in a text, as shown by Yessenalina, Choi, and Cardie (2010). Ehsan et al. (2019) train a model on annotator rationales to generate rationales for predictions, and evaluate through a user study. All previously mentioned rationales are not faithful explanations however, since they are generated post-hoc or separately from the classification task.

Recent work by Jain et al. Jain et al. (2020) propose a method to faithfully identify rationales. Compared to our work, their approach uses an encoder-decoder, where the encoder requires annotator rationale examples to recognize potential rationales.

3 Dataset

The dataset used for training and finding rationales is the IMDB[1] movie review dataset enriched with annotator rationales by Zaidan, Eisner, and Piatko (2007)[2]. This dataset is chosen because it is often used in research on explanation through rationales (DeYoung et al. 2020) and contains straightforward rationales, which allows for user evaluation on a large target audience (everyone that can read English texts). The dataset consists of 1000 positive and 1000 negative textual reviews on movies from the polarity dataset (v2.0) from the Movie Review Dataset by Pang and Lee (2004). The enriched dataset contains annotator rationales annotated by human annotators. These annotators were asked to highlight words and phrases that justified a given positive or negative classification. Only rationales for the requested classification were required. The number of rationales selected depended on the annotator, who was requested to mark enough rationales to provide convincing support for the class of interest.

The average number of rationales annotated per document is 8.55. In this study, the whole sentence around the rationale is used as rationale, to reduce computational cost. The average number of rationales is therefore reduced to 8 per document, which can be explained by the occurrence of multiple rationales in the same sentence.

We split the dataset into multiple balanced sets for training (1200), tuning (200), testing (400), and user evaluation (200). The user evaluation set does not contain annotator rationales and is therefore used in the user evaluation.

4 Preprocessing

We preprocess the dataset as follows: Documents are split into sentences using the NLTK English punkt tokenizer[3](Loper and Bird 2002). The annotation tags (<POS></POS> and <NEG></NEG>) are removed from the text. All other words and punctuation except for repeating dots (...) are left in the documents.

Sentences are embedded using the Sentence-BERT (Reimers and Gurevych 2019) embedding model. This embedding model encodes specifically on sentence-level and therefore overcomes limitations of regular Bidirectional Encoder Representations from Transformers (BERT) models (Devlin et al. 2019). Sentence-BERT uses the BERT model to encode text and applies pooling to the output to derive semantically meaningful fixed size sentence embeddings

[1]Internet Movie Database (Miller, Vandome, and McBrewster 2009)

[2]https://www.cs.jhu.edu/~ozaidan/rationales/

[3]tokenizers/punkt/english.pickle

(Reimers and Gurevych 2019). Every sentence is embedded as a vector of size 768. Every document is padded with as many sentences as needed to create similar-sized documents. All documents are transformed into a format based on the dimensions of the longest document in the dataset. The longest document in the dataset contains 116 sentences, and therefore every document is transformed to 116 vectors with a length of 768.

5 Model

To find faithful explanations in the form of machine-annotated rationales, the following approach is taken. Our work builds on BagNets by Brendel and Bethge (2019), who determine the contribution of chunks of images to a classification. We apply this bag-of-features concept to text classification.

We train a neural network to solve the classification task that uses Sentence-BERT embeddings as input, with Binary-Cross-Entropy loss and the ADAM (Kingma and Ba 2015) optimizer. Every sentence in the text is an input feature. Sentences that strongly contribute to a prediction form machine-annotated rationales.

We propose two variations of the model. The first variation uses all sentences (BagOfSentences) in the text, and the second variation uses a subset of sentences marked by the model as machine-annotated rationales (BagOfRationales) to base a prediction on. In Figure 1 a visualisation of both model variations is given.

5.1 BagOfSentences model

Our BagOfSentences (BoS) model uses an architecture similar to the BagNets (Brendel and Bethge 2019) architecture: the classification is done by dividing the input into multiple chunks. Then, the outputs of all chunks are combined to find a final prediction. The chunks are in the format of sentences. The bag-of-features concept is implemented as a bag-of-sentences in the BagOfSentences model.

In the BagOfSentences model, every Sentence-BERT embedded sentence in the document is passed through multiple convolutional layers, which are at the end passed through a linear layer. This linear layer gives one output for every sentence in the document. To make a prediction, the output for all sentences (116 outputs, see Section 4) are put through a sigmoid function, and the average of those values is used to decide the class. Eq. (1) shows the formula for the final prediction μ by the BagOfSentences model, where S is the set of sentences, $\hat{S}$ is the output of the BagOfSentences model and $pred : s \rightarrow \hat{s}$ is the bijective function of the prediction $\hat{s} \in \hat{S}$ for sentence $s \in S$. Note that the padding is also included in S.

$$\mu = pred_{BoS}(S) = \frac{\sum_{s \in S} pred(s)}{\|S\|} = \frac{\sum_{\hat{s} \in \hat{S}} \hat{s}}{\|S\|} \quad (1)$$

For the BagOfSentences model, the following steps are taken:

1. Input the Sentence-BERT embedded vectors.
2. Transpose the 116×768 tensor to 768×116 tensor for the convolutional layers[4].
3. Pass through 6 1D convolutional layers[5], consisting of 768 input channels and 768 output channels. The used kernel size is 1, which moves along (but does not merge) sentences. The activation function used is ReLU(Agarap 2019).
4. Transpose the 768×116 tensor back to a 116×768 tensor for the linear layer.
5. Use a linear layer with 768 inputs and 1 output to generate a single output $\hat{s}$ for each sentence.
6. Use the sigmoid function on all 116 sentence outputs $\hat{s} \in \hat{S}$ to obtain a class value for every sentence.
7. Take the average μ of all sentence outputs $\hat{S}$.
8. Compare the μ with the class label and use this to update the model's gradients. The μ is the final prediction.

The training process of the BagOfSentences model differs from the BagNets model by Brendel and Bethge (2019), in that the BagNets model is trained to predict using small image chunks, where every chunk receives feedback during the training process. Thus, a label for every chunk is available in the used dataset. Instead of training the BagOfSentences model to classify independent sentences, we only use the polarity of whole documents as feedback. This feedback is given after the final prediction is made, so after all sentence chunks are combined. Thus, the BagOfSentences learns the polarity of sentences, with the document class as a label. In a way, the model trains on noisy labels, because the class label does not apply to all input sentences, like padding or neutral sentences.

5.2 Extracting machine-annotated rationales

Selecting sentences that form machine-annotated rationales rationales can be done by for example taking the top t rationales, where t is the average number of annotator rationales in all documents in the training dataset (DeYoung et al. 2020), or a fixed percentage of the sentences in the document (Jain et al. 2020).

In this study, a different approach to find a variable number of machine-annotated rationales is used: Sentences are compared to the full distribution of all sentences in the document. This is done by creating a histogram of the distribution with n bins, and only selecting sentences in the leftmost or rightmost bin as rationales. Both sides of the histogram represent a class in a binary classification task. The number of bins n is determined by the bin-size: $\#bins = 1/\text{bin-size}$, and can be adjusted per model. A bin-size of 0.1 for instance would result in 10 bins. Using this approach, only sentences that show a considerable support for a class are selected as rationales, instead of a fixed set of more-or-less supporting sentences. Using this approach, machine-annotated rationales can be extracted from the BagOfSentences model.

[4]The number 116 is the maximum number of sentences in all documents in the dataset and 768 is the length of the Sentence-BERT vectors.

[5]The number 6 is chosen because of a time-accuracy trade-off during development.

These machine-annotated rationales form faithful explanations for predictions, because these sentences directly influence the prediction, and removing these sentences would alter the prediction.

The following steps are taken to find machine-annotated rationales:

1. Perform the steps described in Section 5.1 up to step 7.
2. Compare the sentence-values to the average of all sentence-values (the prediction). Select all sentences that have an output that is $\leq$ (positive prediction), or $\geq$ (negative prediction) than a threshold. Those sentences are treated as the machine-annotated rationales. The threshold is found using the distribution of the logits in the model's output and a bin-size of 0.2[6], as described above.

5.3 BagOfRationales: BagOfSentences with restricted input

Not all sentences in a preprocessed document are useful for polarity classification. For example, a padding sentence does not contain any information, and some sentences can be neutral. We introduce the BagOfRationales (BoR) model as a variation of the BagOfSentences model, where instead of using all sentences, only the sentences marked as relevant (rationales) are used to make the final prediction. After step 6 in the BagOfSentences model, continue as follows:

7. Take the average μ of all sentence outputs.
   (a) Find sentences that support[7] the found prediction (rounded μ). These are the machine-annotated rationales R for the given prediction.
   (b) Take the average $\hat{\mu}$ of R.
8. Compare the $\hat{\mu}$ with the class label and use this to update the model's gradients. The found $\hat{\mu}$ is the final prediction.

By adding these steps, the final prediction $\hat{\mu}$ will be closer to the class label than the prediction made by the BagOfSentences model, since only the sentences with values that (strongly) support the prediction μ are used. Sentences with different logits, like padding, will bring the prediction closer to 0.5, because they average the other sentences out.

See Eq. 2 for the formula used to calculate the final prediction $\hat{\mu}$, where S is the set of sentences, $pred : s \rightarrow \hat{s}$ is the bijective function of the prediction $\hat{s}$ for sentence $s \in S$, and R is the set of selected rationales.

$$\hat{\mu} = pred_{BoR}(S, R) : \frac{\sum_{s \in S}\{pred(s) | s \in R\}}{\|R\|} \quad (2)$$

6 Evaluation and discussion

In this section, we present the approach and results of our evaluation. We focus on the quality of explanation, and not on classification accuracy. As a baseline, a set of a (positive) random number of randomly selected sentences is used a rationales. This set is called the random rationale set.

The found machine-annotated rationales are evaluated by measuring their faithfulness, comparing them to annotator rationales using set theory, and through a user evaluation.

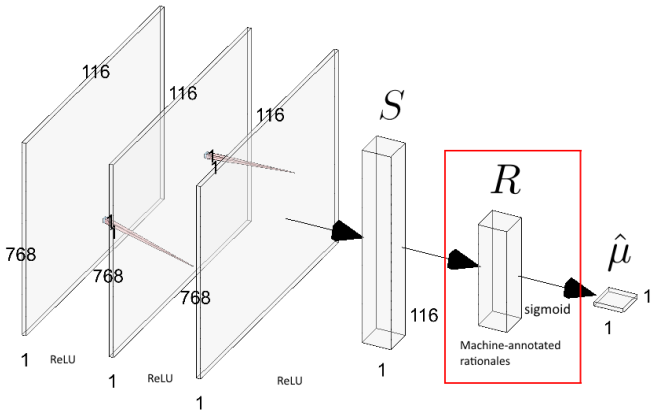

Figure 1: The BagOfSentences and BagOfRationales model architecture. The area in the square marks the additional steps taken in the BagOfRationales model.

6.1 Classification

Current state-of-the-art models on this classification task achieve accuracies above 95% (Thongtan and Phienthrakul 2019). The BagOfSentences model reaches a classification accuracy of 0.888, and the BagOfRationales model an accuracy of 0.847 on the test set. The BagOfRationales model uses less information to base a classification on (only rationales), which can explain the decrease in classification accuracy of this model. When relevant information is removed from the input, the classification accuracy decreases.

6.2 Faithfulness

An explanation is faithful when it explains according to the inner workings of a model. We measure the faithfulness of found machine-annotated rationales using the comprehensiveness metric proposed by DeYoung et al. (2020): Every sentence that is labelled as a machine-annotated rationale is replaced by padding, and the resulting document is used as input for the new prediction. The difference between the original prediction probability and the adjusted prediction probability is called the comprehensiveness. Removing rationales should result in a high comprehensiveness score. We do not use a fixed number of ranked rationales like DeYoung et al. (2020), but use the approach as described in Section 5.2 to select rationales. In addition, we use the sufficiency metric proposed by DeYoung et al. (2020), which removes all sentences that are not in the selected set of rationales from the input, and compares the new prediction to the original one. We interpret a low sufficiency score as a sign that the model mainly uses the selected rationales in the prediction and thus explain the prediction. We use normalized predic-

[6]The number 0.2 is chosen arbitrarily and needs to be finetuned per dataset.

[7]The sentences that fall in the top 20% of the distribution for the given class, see Section 5.2.

	Rand. + BoS	Rand. + BoR	BagOfSentences (BoS)			BagOfRationales (BoR)		
	total	**total**	**total**	corr.	incorr.	**total**	corr.	incorr.
# docs	400	400	400	355	45	400	339	61
Comp.	0.066	0.135	0.138	0.139	0.108	0.569	0.544	0.705
Suff.	0.069	0.133	0.039	0.035	0.06	0.047	0.045	0.059

Table 1: Comprehensiveness and sufficiency for BagOfSentences and BagOfRationales models on the test set. A distinction between rationales for correctly classified documents (correct) and incorrectly classified documents (incorrect) is made. Random rationales (rand.) are used as a baseline.

tions in both the comprehensiveness and sufficiency metrics to compare models.

Results See Table 1 for the faithfulness scores of the BagOfSentences and BagOfRationales models. The baseline of randomly selected rationales is used to give an indication of faithfulness scores. The comprehensiveness is higher for the BagOfRationales model. One explanation for this difference in comprehensiveness scores is that the BagOfRationales classifies using only rationales, and removing rationales therefore changes the prediction more notably. The sufficiency metric is similarly low for both models, showing that both models do not use most of the non-rationales in their prediction.

6.3 Comparison to annotator rationales

We now compare the machine-annotated rationales to human annotator rationales. Annotator rationales form a benchmark of interpretable explanations for this classification task, since they are explanations by humans for humans and therefore very interpretable to humans. These explanations are subjective and may vary per annotator (Bao et al. 2018). DeYoung et al. (2020) evaluate found rationales by measuring how much they agree with human rationales. We adopt their metrics, but compare rationales on sentence-level instead of token-level. We use the Jaccard Index to measure the overlap between two sets of rationales. This measure is sometimes called Intersection-Over-Union (IOU) (DeYoung et al. 2020). In addition, we use sentence-level precision, recall, and F1-score to measure the similarity of annotator- and machine-annotated rationale sets.

Results Our results are presented in Table 2. The machine-annotated rationales found by the BoS and BoR models are quite similar, with an average Jaccard index of 0.678. On average, the sets of machine-annotated rationales contain fewer rationales than the set of annotator rationales. In Table 1 on the Github page[8] an overview of the average number of selected rationales per set is given. A difference between

[8]https://git.science.uu.nl/e.herrewijnen/machine-annotated-rationales

	Random rationales	BagOfSentences (BoS)			BagOfRationales (BoR)		
	total	**total**	corr.	incorr.	**total**	corr.	incorr.
# docs	400	400	355	45	400	339	61
Jacc.	0.171	0.326	0.361	0.052	0.316	0.362	0.060
Prec.	0.226	0.546	0.603	0.102	0.550	0.629	0.110
Rec.	0.505	0.448	0.483	0.085	0.405	0.460	0.098
F1	0.312	0.492	0.536	0.092	0.466	0.531	0.103

Table 2: Rationale quality for random rationale baseline, BoS, and BoR rationales compared to annotator rationales. A distinction between rationales for correctly classified documents (correct) and incorrectly classified documents (incorrect) is made.

correct and incorrect classifications by the models and similarity to annotator rationales is visible: the similarity scores for correctly classified are much better than for incorrectly classified documents (see Table 2). This difference indicates that when a ML model predicts in a way that results in the same classification result as a human classification (i.e. correct), the explanation is more similar to a human explanation. We select machine-annotated rationales that stand out relatively to the distribution of all sentences. This way, only the sentences that have a clear influence on the prediction are selected as rationales. The found machine-annotated rationales are more concise (i.e. contain fewer sentences) than annotator rationales, and occasionally too concise in the user evaluation. Increasing the bin-size might improve the precision of the found MaRs, but decrease recall since the chance of selecting non-annotator rationales also increases.

6.4 User evaluation

Since explanation is subjective (Bao et al. 2018), we use a user evaluation to measure the usefulness of machine-annotator rationales. Ehsan et al. (2019) evaluate post-hoc generated rationales by asking users to score explanations on confidence, human-likeness, adequate justification, and understandability.

We measure the quality of our selected rationales using a blind study, where users do not know the source (human or ML model) of the explanation. By doing this, the usefulness of the found rationales as explanation can be measured without potential user bias[9] for the explanation source. Our user study consisted of 45 users (students and colleagues) that classified one or two[10] sets of machine-annotated rationales for one of 8 different documents. Users were asked to voluntarily fill out an online survey.

Users first perform a sanity check to find out whether they understand the concept of annotator rationales, where they have to select all sentences in a document that form rationales. This document contains a set of obvious rationales, which forms a baseline for the sanity check. Then users are presented with a set of rationales (annotator rationales or

[9]E.g. humans are better at explaining than ML models.

[10]More sets of machine-annotated rationales where compared in the blind study, but their results are not relevant for this study.

machine-annotated rationales) without knowing their source (human or ML model), and asked to make a classification based on the given explanation. This approach is similar to forward simulation (Doshi-Velez and Kim 2017; Nguyen 2018), but uses only explanation as a classification base, instead of using the input and the explanation. We call this task the blind study task. Users have three options for classifying a document after being presented with a set of rationales: 1) Positive, 2) Negative, 3) I need more information. These options reflect whether a user understands an explanation and whether all the required information is present in the explanation. When the user's prediction agrees with the model's prediction, the explanation is understood. When the predictions disagree, the explanation is misleading. When a user needs more information, a prediction does not contain required information.

Only machine-annotated rationales found by the BagOfRationales model are used, since these rationales scored higher on the comprehensiveness metric, but are not very different from the BagOfSentences rationales (Jaccard index of 0.678). To gain insight in cases where a ML model predicts differently from a human, we also presented users with machine-annotated rationales for a subset of incorrectly classified documents.

Results In Figure 2 the results of the blind study task are presented. As a benchmark, annotator rationales were also used in the blind study task. For correct classification by the ML model, the percentage of correct classifications by users should be as high as possible. Incorrect and incomplete classification by users imply that the explanation is respectively misleading and incomplete.

In cases where the ML model makes a wrong classification, the model predicts differently than humans. Then, the percentage of *correct* user classifications indicates that the given explanation does *not* reflect the model's prediction. Alternatively, an incorrect classification shows that the explanation does reflect and support the model's prediction. A high percentage of classifications marked as incomplete shows that the given explanation does not contain enough information for the user to base a classification on. A high percentage of incorrect classifications implies that the explanation supports the model's prediction. These results show that the found machine-annotated rationales are mostly useful when the model classifies correctly. Thus, when the models predicts somewhat similarly to a human, the explanations found are interpretable to humans. When the model made a wrong classification, the users marked the machine-annotated rationales as incomplete most often. If the model predicts differently (incorrectly) from a human, the found explanation does not help users to understand why an action has been performed. Guaranteeing that a model predicts similarly to a human and comes to the same classification result can make found explanations more interpretable.

7 Conclusion

We proposed two models that faithfully identify rationales: BagOfSentences and BagOfRationales. We showed that using a bag-of-features approach for text classification can bring relevant parts of text to the surface, which can be used as explanation. Finding these machine-annotated rationales does not require human explanation examples during the training process. Furthermore, the extraction of machine-annotated rationales is done during the prediction process, which makes the method computationally inexpensive.

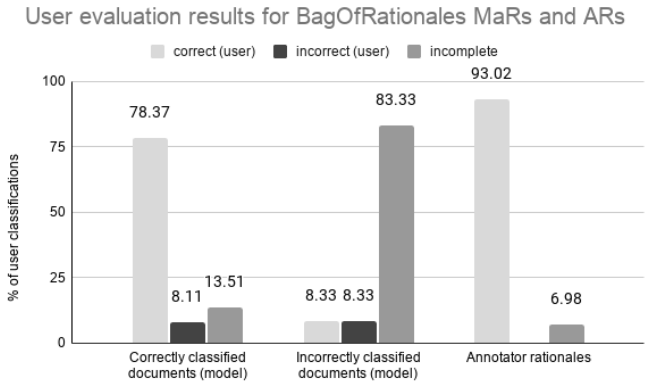

Figure 2: User-classifications based on machine-annotated rationales. Left: correct model classifications (37 documents). Center: incorrect model classifications (12 documents). Right: annotator rationale classifications (43 documents).

Found machine-annotated rationales show some overlap with annotator rationales, especially when the ML model reasons somewhat similar to humans (i.e. correct classifications). Using annotator rationales as a rationale quality benchmark might be a very ambitious benchmark.

User evaluation results show that found machine-annotated rationales can be useful as explanation, even though they are more concise and occasionally found incomplete by users. By using a blind study approach, the machine-annotated rationales can be compared to annotator rationales, without requiring user to have experience with explainable artificial intelligence.

References

Agarap, A. F. 2019. Deep Learning using Rectified Linear Units (ReLU).

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Bao, Y.; Chang, S.; Yu, M.; and Barzilay, R. 2018. Deriving Machine Attention from Human Rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1903–1913.

Berk, R.; Heidari, H.; Jabbari, S.; Kearns, M.; and Roth, A. 2018. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods & Research* 0049124118782533.

Brendel, W.; and Bethge, M. 2019. Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. *International Conference on Learning Representations* .

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

DeYoung, J.; Jain, S.; Rajani, N. F.; Lehman, E.; Xiong, C.; Socher, R.; and Wallace, B. C. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. *Transactions of the Association for Computational Linguistics* .

Doshi-Velez, F.; and Kim, B. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *stat* 1050: 2.

Ehsan, U.; Tambwekar, P.; Chan, L.; Harrison, B.; and Riedl, M. O. 2019. Automated Rationale Generation: A Technique for Explainable AI and its Effects on Human Perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 263–274. ACM.

Jacovi, A.; and Goldberg, Y. 2020. Towards Faithfully Interpretable NLP Systems: How should we define and evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4198–4205. Online: Association for Computational Linguistics.

Jain, S.; and Wallace, B. C. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, 3543–3556. Minneapolis, Minnesota: Association for Computational Linguistics.

Jain, S.; Wiegreffe, S.; Pinter, Y.; and Wallace, B. C. 2020. Learning to Faithfully Rationalize by Construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4459–4473. Online: Association for Computational Linguistics.

Kindermans, P.-J.; Hooker, S.; Adebayo, J.; Alber, M.; Schütt, K. T.; Dähne, S.; Erhan, D.; and Kim, B. 2019. *The (Un)reliability of Saliency Methods*, 267–280. Cham: Springer International Publishing. ISBN 978-3-030-28954-6.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference for Learning Representations*.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 107–117. Austin, Texas: Association for Computational Linguistics.

Li, J.; Monroe, W.; and Jurafsky, D. 2016. Understanding Neural Networks through Representation Erasure. *arXiv* arXiv–1612.

Lipton, Z. C. 2018. The Mythos of Model Interpretability. *Queue* 16(3): 31–57.

Loper, E.; and Bird, S. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, 63–70.

Miller, F. P.; Vandome, A. F.; and McBrewster, J. 2009. *Internet Movie Database*. Alpha Press. ISBN 6130099681.

Miller, T.; Howe, P.; and Sonenberg, L. 2017. Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences.

Nguyen, D. 2018. Comparing Automatic and Human Evaluation of Local Explanations for Text Classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1069–1078. New Orleans, Louisiana: Association for Computational Linguistics.

Pang, B.; and Lee, L. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the ACL*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Robnik-Šikonja, M.; and Kononenko, I. 2008. Explaining Classifications for Individual Instances. *IEEE Transactions on Knowledge and Data Engineering* 20(5): 589–600.

Rudin, C. 2019. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence* 1(5): 206–215.

Sen, C.; Hartvigsen, T.; Yin, B.; Kong, X.; and Rundensteiner, E. 2020. Human Attention Maps for Text Classification: Do Humans and Neural Networks Focus on the Same Words? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4596–4608. Association for Computational Linguistics.

Thongtan, T.; and Phienthrakul, T. 2019. Sentiment Classification Using Document Embeddings Trained with Cosine Similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 407–414. Florence, Italy: Association for Computational Linguistics.

Tjoa, E.; and Guan, C. 2020. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems* .

Wiegreffe, S.; and Pinter, Y. 2019. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 11–20.

Yessenalina, A.; Choi, Y.; and Cardie, C. 2010. Automatically Generating Annotator Rationales to Improve Sentiment Classification. In *Proceedings of the ACL 2010 Conference Short Papers*, 336–341.

Zaidan, O.; Eisner, J.; and Piatko, C. 2007. Using "Annotator Rationales" to Improve Machine Learning for Text Categorization. In *NAACL HLT 2007; Proceedings of the Main Conference*, 260–267.

Zaidan, O. F.; and Eisner, J. 2008. Modeling Annotators: A Generative Approach to Learning from Annotator Rationales. In *Proceedings of EMNLP 2008*, 31–40.

Zhang, Y.; Marshall, I.; and Wallace, B. C. 2016. Rationale-Augmented Convolutional Neural Networks for Text Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 795–804. Austin, Texas: Association for Computational Linguistics.