Human-grounded Evaluations of Explanation Methods for Text Classification

Piyawat Lertvittayakumjorn and **Francesca Toni**
Department of Computing
Imperial College London, UK
{pl1515, ft}@imperial.ac.uk

Abstract

Due to the black-box nature of deep learning models, methods for explaining the models' results are crucial to gain trust from humans and support collaboration between AIs and humans. In this paper, we consider several model-agnostic and model-specific explanation methods for CNNs for text classification and conduct three human-grounded evaluations, focusing on different purposes of explanations: *(1)* revealing model behavior, *(2)* justifying model predictions, and *(3)* helping humans investigate uncertain predictions. The results highlight dissimilar qualities of the various explanation methods we consider and show the degree to which these methods could serve for each purpose.

1 Introduction

Explainable Artificial Intelligence (XAI) is aimed at providing explanations for decisions made by AI systems. The explanations are useful for supporting collaboration between AIs and humans in many cases (Samek et al., 2018). Firstly, if an AI outperforms humans in a certain task (e.g., AlphaGo (Silver et al., 2016)), humans can learn and distill knowledge from the given explanations. Secondly, if an AI's performance is close to human intelligence, the explanations can increase humans' confidence and trust in the AI (Symeonidis et al., 2009). Lastly, if an AI is duller than humans, the explanations help humans verify the decisions made by the AI and also improve the AI (Biran and McKeown, 2017).

One of the challenges in XAI is to explain prediction results given by deep learning models, which sacrifice transparency for high prediction performance. This type of explanations is called *local explanations* as they explain individual predictions (in contrast to *global explanations* which explain the trained model independently of any specific prediction). There have been several methods proposed to produce local explanations. Some of them are model-agnostic, applicable to any machine learning model (Ribeiro et al., 2016; Lundberg and Lee, 2017). Others are applicable to a class of models such as neural networks (Bach et al., 2015a; Dimopoulos et al., 1995) or to a specific model such as Convolutional Neural Networks (CNNs) (Zhou et al., 2016).

With so many explanation methods available, the next challenge is how to evaluate them so as to choose the right methods for different settings. In this paper, we focus on human-grounded evaluations of local explanation methods for text classification. Particularly, we propose three evaluation tasks which target different purposes of explanations for text classification – *(1)* revealing model behavior to human users, *(2)* justifying the predictions, and *(3)* helping humans investigate uncertain predictions. We then use the proposed tasks to evaluate nine explanation methods working on a standard CNN for text classification. These explanation methods are different in several aspects. For example, regarding granularity, four explanation methods select words from the input text as explanations, whereas the other five select n-grams as explanations. In terms of generality, one of the explanation methods is model-agnostic, two are random baselines, another two (newly proposed in this paper) are specific to 1D CNNs for text classification, and the rest are applicable to neural networks in general. Overall, the contributions of our work can be summarized as follows.

- We propose three human-grounded evaluation tasks to assess the quality of explanation methods with respect to different purposes of usage for text classification. (Section 3)
- To increase diversity in the experiments, we develop two new explanation methods for

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 5195–5205,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

CNNs for text classification. One is based on gradient-based analysis (**Grad-CAM-Text**). The other is based on model extraction using **decision trees**. (Section 4.3.1-4.3.2)

- We evaluate both new methods as well as random baselines and well-known existing methods using the three evaluation tasks proposed. The results highlight dissimilar qualities of the explanation methods and show the degree to which these methods could serve for each purpose. (Section 5)

1.1 Terminology Used

We use the following terms throughout the paper. (1) **Model**: a deep learning classifier we want to explain, e.g., a CNN. (2) **Explanation**: an ordered list of text fragments (words or n-grams) in the input text which are most relevant to a prediction. Explanations for and against the predicted class are called **evidence** and **counter-evidence**, respectively. (3) **(Local) explanation method**: a method producing an explanation for a model and an input text. (4) **Evaluation method**: a process to quantitatively assign to explanations scores which reflect the quality of the explanation method.

2 Background and Related Work

This section discusses recent advances of explanation methods and evaluation for text classification as well as background knowledge about 1D CNNs – the model used in the experiments.

2.1 Local Explanation Methods

Generally, there are several ways to explain a result given by a deep learning model, such as explaining by examples (Kim et al., 2014) and generating textual explanations (Liu et al., 2019). For text classification in particular, most of the existing explanation methods identify parts of the input text which contribute most towards the predicted class (so called *attribution methods* or *relevance methods*) by exploiting various techniques such as input perturbation (Li et al., 2016), gradient analysis (Dimopoulos et al., 1995), and relevance propagation (Arras et al., 2017b). Besides, there are other explanation methods designed for specific deep learning architectures such as attention mechanism (Ghaeini et al., 2018) and extractive rationale generation (Lei et al., 2016).

We select some well-known explanation methods (which are applicable to CNNs for text classification) and evaluate them together with two new explanation methods proposed in this paper.

2.2 Evaluation Methods

Focusing on text classification, early works evaluated explanation methods by *word deletion* – gradually deleting words from the input text in the order of their relevance and checking how the prediction confidence drops (Arras et al., 2016; Nguyen, 2018). Arras et al. (2017a) and Xiong et al. (2018) used relevance scores generated by explanation methods to construct document vectors by weighted-averaging word vectors and checked how well traditional machine learning techniques manage these document vectors. Poerner et al. (2018) proposed two evaluation paradigms – hybrid documents and morphosyntactic agreements. Both check whether an explanation method correctly points to the (known) root cause of the prediction. Note that all of the aforementioned evaluation methods are conducted with no humans involved.

For human-grounded evaluation, Mohseni and Ragan (2018) proposed a benchmark which contains a list of relevant words for the actual class of each input text, identified by human experts. However, comparing human explanations with the explanations given by the tested method may be inappropriate since the mismatches could be due to not only the poor explanation method but also the inaccuracy of the model or the model reasoning differently from humans. Nguyen (2018) asked humans to guess the output of a text classifier, given an input text with the highest relevant words highlighted by the tested explanation method. Informative (and discriminative) explanations will lead to humans' correct guesses. Ribeiro et al. (2016) asked humans to choose a model which can generalize better by considering their local explanations. Also, they let humans remove irrelevant words, existing in the explanations, from the corpus to improve the prediction performance. Compared to previous work, our work is more comprehensive in terms of the various human-grounded evaluation tasks proposed and the number and dimensions of explanation methods being evaluated.

2.3 CNNs for Text Classification

CNNs have been found to achieve promising results in many text classification tasks (Johnson and Zhang, 2015; Gambäck and Sikdar, 2017; Zhang et al., 2019). Figure 1 shows a standard 1D CNN

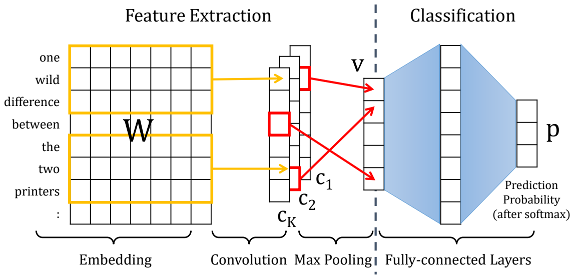


Figure 1: CNN for text classification.

for text classification which consists of four main steps: *(i)* embedding an input text into an embedded matrix $\mathbf{W}$; *(ii)* applying K fixed-size convolution filters to $\mathbf{W}$ to find n-grams that possibly discriminate one class from the others; *(iii)* pooling only the maximum value found by each filter, corresponding to the most relevant n-gram in text, to construct a filter-based feature vector, $\mathbf{v}$, of the input; and *(iv)* using fully-connected layers (FC) to predict the results, and applying a softmax function to the outputs to obtain predicted probability of the classes ($\mathbf{p}$), i.e., $\mathbf{p} = softmax(FC(\mathbf{v}))$. While the original version of this model uses only one linear layer as FC (Kim, 2014), more hidden layers can be added to increase the model capacity for prediction. Also, more than one filter size can be used to detect n-grams with short- and long-span relations (Conneau et al., 2017).

3 Human-grounded Evaluation Methods

We propose three human tasks to evaluate explanation methods for text classification as summarized in Table 1. Figure 2 gives an example question for each task, discussed next.

3.1 Revealing the Model Behavior

Task 1 evaluates whether explanations can expose irrational behavior of a poor model. This property of explanation methods is very useful when we do not have a labelled dataset to evaluate the model quantitatively. To set up the task, firstly, we train two models to make them have different performance on classifying testing examples (i.e., different capability to generalize to unseen data). Then we use these models to classify an input text and apply the explanation method of interest to explain the predictions – highlighting top-m evidence text fragments on the text for each model. Next, we ask humans, based on the highlighted texts from the two models, which model is more reasonable? If the performance of the two models is clearly different, good explanation methods should enable humans to notice the poor model, which is more likely to decide based on non-discriminative words, even though both models predict the same class for an input text.

Additionally, there are some important points to note for this task. First, the chosen input texts must be classified into the same class by both models so the humans make decisions based only on the different explanations. However, it is worth to consider both the cases where both models correctly classify and where they misclassify. Second, we provide the choices for the two models along with confidence levels for the humans to select. If they select the right model with high confidence, the explanation method will get a higher positive score. In contrast, a confident but incorrect answer results in a large negative score. Also, the humans have the option to state no preference, for which the explanation method will get a zero score (See the last row of Table 1).

3.2 Justifying the Predictions

Explanations are sometimes used by humans as the reasons for the predicted class. This task tests whether the evidence texts are truly related to the predicted class and can distinguish it from the other classes, so called *class-discriminative* (Selvaraju et al., 2017). To set up the task, we use a well-trained model and select an input example classified by this model with high confidence ($\max_c \mathbf{p}_c > \tau_h$ where τ_h is a threshold parameter), so as to reduce the cases of unclear explanations due to low model accuracy or text ambiguity. (Note that we will look at low-confidence predictions later in Task 3.) Then we show only the top-m evidence text fragments generated by the method of interest to humans and ask them to guess the class of the document containing the evidence. The explanation method which makes the humans surely guess the class predicted by the model will get a high positive score. As in the previous task, this task considers both the correct and incorrect predictions with high confidence to see how well the explanations justify each of the cases. For incorrect predictions, an explanation method gets a positive score when a human guesses the same incorrect class after seeing the explanation. In real applications, convincing explanations for incorrect classes can help humans understand the

	Task 1 (Section 3.1)	**Task 2** (Section 3.2)	**Task 3** (Section 3.3)
Assumption	Good explanations can reveal model behavior	Good explanations justify the predictions	Good explanations help humans investigate uncertain predictions
Model(s)	Two classifiers with different performance on a test dataset	One well-trained classifier	One well-trained classifier
Input text	A test example for which both classifiers predict the same class	A test example which the classifier predicts with high confidence ($\max_c \mathbf{p}_c > \tau_h$)	A test example which the classifier predicts with low confidence ($\max_c \mathbf{p}_c < \tau_l$)
Information displayed	1. The input text 2. The predicted class 3. (Highlighted) top-m evidence texts of each model	1. Top-m evidence texts	1. The predicted class 2. The predicted probability $\mathbf{p}$ 3. Top-m evidence and top-m counter-evidence texts
Human task	Select the more reasonable model and state if they are confident or not	Select the most likely class of the document which contains the evidence texts and state if they are confident or not	Select the most likely class of the input text and state if they are confident or not
Scores to the explanation method	(-)1.0: (In)correct, confident (-)0.5: (In)correct, unconfident 0.0: No preference	(-)1.0: (In)correct, confident (-)0.5: (In)correct, unconfident 0.0: No preference	(-)1.0: (In)correct, confident (-)0.5: (In)correct, unconfident

Table 1: A summary of the proposed human-grounded evaluation tasks.

model's weakness and create additional fixing examples to retrain and improve the model.

3.3 Investigating Uncertain Predictions

If an AI system makes a prediction with low confidence, it may need to raise the case with humans and let them decide, but with the analyzed results as additional information. This task aims to check if the explanations can help humans comprehend the situation and correctly classify the input text or not. To set up, we use a well-trained model and an input text classified by this model with low confidence ($\max_c \mathbf{p}_c < \tau_l$ where τ_l is a threshold parameter). Then we apply the explanation method of interest to find top-m evidence and top-m counter-evidence texts of the predicted class. We present both types of evidence to humans[1] together with the predicted class and probability $\mathbf{p}$ and ask the humans to use all the information to guess the actual class of the input text, without seeing the input text itself. The scoring criteria of this task are similar to the previous tasks except that we do not provide the "no preference" option as the humans can still rely on the predicted scores when all the explanations are unhelpful.

4 Experimental Setup

4.1 Datasets

We used two English textual datasets for the three tasks.

(1) **Amazon** Review Polarity is a sentiment analysis dataset with positive and negative classes (Zhang et al., 2015). We randomly selected 100K, 50K, and 100K examples for training, validating, and testing the CNN models, respectively.

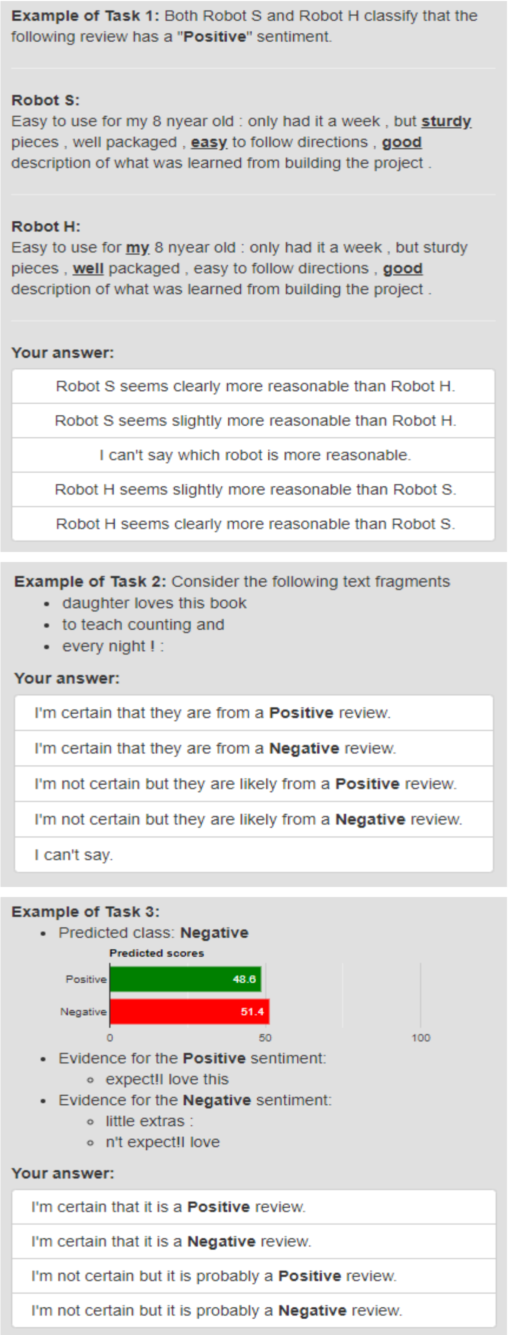
(2) **ArXiv** Abstract is a text classification dataset we created by collecting abstracts of scientific articles publicly available on ArXiv[2]. Particularly, we collected abstracts from the "Computer Science (CS)", "Mathematics (MA)", and "Physics (PH)" categories, which are the three main categories on ArXiv. We then created a dataset with three disjoint classes removing the abstracts which belong to more than one of the three categories. In the experiments, we randomly selected 6K, 1.5K, and 10K examples for training, validating, and testing the CNN model, respectively.

4.2 Classification Models: 1D CNNs

As for the classifiers, we used 1D CNNs with the same structures for all the tasks and datasets. Specifically, we used 200-dim GloVe vectors as non-trainable weights in the embedding layer (Pennington et al., 2014). The convolution layer had three filter sizes [2, 3, 4] with 50 filters for each size, while the intermediate fully-connected layer had 150 units. The activation functions of the filters and the fully-connected layers are ReLU (except the softmax at the output layer). The models were implemented using Keras and trained with Adam optimizer. The macro-average F1 are 0.90 and 0.94 for the Amazon and the ArXiv datasets, respectively. Overall, the ArXiv appears to be an easier task as it is likely solvable by looking at individual keywords. In contrast, the Ama-

[1] We present counter-evidence as evidence for the other classes to simplify the task questions.

[2] https://arxiv.org



Example of Task 1: Both Robot S and Robot H classify that the following review has a "**Positive**" sentiment.

Robot S:
Easy to use for my 8 nyear old : only had it a week , but **sturdy** pieces , well packaged , **easy** to follow directions , **good** description of what was learned from building the project .

Robot H:
Easy to use for **my** 8 nyear old : only had it a week , but sturdy pieces , **well** packaged , easy to follow directions , **good** description of what was learned from building the project .

Your answer:

Robot S seems clearly more reasonable than Robot H.

Robot S seems slightly more reasonable than Robot H.

I can't say which robot is more reasonable.

Robot H seems slightly more reasonable than Robot S.

Robot H seems clearly more reasonable than Robot S.

Example of Task 2: Consider the following text fragments

- daughter loves this book
- to teach counting and
- every night ! :

Your answer:

I'm certain that they are from a **Positive** review.

I'm certain that they are from a **Negative** review.

I'm not certain but they are likely from a **Positive** review.

I'm not certain but they are likely from a **Negative** review.

I can't say.

Example of Task 3:

- Predicted class: **Negative**

- Evidence for the **Positive** sentiment:
  - expect!I love this
- Evidence for the **Negative** sentiment:
  - little extras :
  - n't expect!I love

Your answer:

I'm certain that it is a **Positive** review.

I'm certain that it is a **Negative** review.

I'm not certain but it is probably a **Positive** review.

I'm not certain but it is probably a **Negative** review.

Figure 2: Example questions and user interfaces.

zon sentiment analysis is not quite easy. Many reviews mention both pros and cons of the products, so a classifier needs to analyze several parts of the input to reach a conclusion. However, this is still manageable by the CNN architecture we used.

Also, in task 1, we need another model which performs worse than the well-trained model. In this experiment, we trained the second CNNs (i.e., the worse models) for the two datasets in different ways to examine the capability of explanation methods in two different scenarios. For the Amazon dataset, while the first (well-trained) CNN needed eight epochs until the validation loss converges, we trained the second CNN for only one epoch to make it underfitting. For the ArXiv dataset, we trained the second CNN using the same number of examples as the first model but with more specific topics. To explain, we randomly selected only examples from the subclass 'Computation and Language', 'Dynamical Systems', and 'Quantum Physics' as training and validation examples for the class 'Computer Science', 'Mathematics', and 'Physics', respectively. In other words, the training and testing data of the worse CNN came from different distributions. As a result, the macro-average F1 of the worse CNNs are 0.81 and 0.85 for the Amazon and the ArXiv datasets, respectively.

Method Name	Approach	Granularity
Random (W)	Random Baselines	Words
Random (N)		N-grams
LIME	Perturbation	Words
LRP (W)	Relevance Propagation	Words
LRP (N)		N-grams
DeepLIFT (W)		Words
DeepLIFT (N)		N-grams
Grad-CAM-Text	Gradient	N-grams
Decision Trees (DTs)	Model Extraction	N-grams

Table 2: Nine explanation methods evaluated.

4.3 Explanation Methods

We evaluated nine explanation methods as summarized in Table 2. First, we used Random (W) and Random (N) as two baselines selecting words and non-overlapping n-grams randomly from the input text as evidence and counter-evidence. For the n-gram random baseline (and other n-gram based explanation methods in this paper), n is one of the CNN filter sizes [2, 3, 4].

Second, we selected LIME which is a well-known model-agnostic perturbation-based method (Ribeiro et al., 2016). It trains a linear model using samples (5,000 samples in this paper) around the input text to explain the importance of each word towards the prediction. The importance scores can be either positive (for the predicted class) or negative (against the predicted class).

Third, we selected layer-wise relevance propagation (LRP), specifically ϵ-LRP (Bach et al., 2015b), and DeepLIFT (Shrikumar et al., 2017) which are applicable to neural networks in general and performed very well in several evaluations using proxy tasks (Xiong et al., 2018; Poerner et al., 2018). LRP propagates the output of the target class (before softmax) back through layers to find attributing words, while DeepLIFT does the same but propagates the difference between the output and the predicted output of the reference input (i.e., all-zero embeddings in this paper). These two methods assign relevance scores to every word in the input text. Words with the highest and the lowest scores are selected as evidence for and counter-evidence against the predicted class, respectively. Also, we extended LRP and DeepLIFT to generate explanations at an n-gram level. We considered all possible n-grams in the input text where n is one of the CNN filter sizes. Then the explanations are generated based on the relevance score of each n-gram, i.e., the sum of relevance scores of all words in the n-gram.

Next, we searched for model-specific explanation methods which target 1D CNNs for text classification. We found that Jacovi et al. (2018) proposed one: listing only n-grams corresponding to feature values in **v** (see Figure 1) that pass thresholds for their filters. Each of the thresholds is set subject to sufficient purity of the classification results above it. However, their method is applicable to CNNs with only one linear layer as FC, while our CNNs have an additional hidden layer (with ReLU activation). So, we could not compare with their method in this work. To increase diversity in the experiments, we therefore propose two additional model-specific methods applicable to 1D CNNs with multiple layers in FC, presented next.

4.3.1 Grad-CAM-Text

We adapt Grad-CAM (Selvaraju et al., 2017), originally devised for explaining 2D CNNs, to find the most relevant n-grams for text classification. Since each value in the feature vector **v** corresponds to an n-gram selected by a filter, we use $E_{j,k}$ to show the effect of an n-gram selected by the k^{th} filter towards the prediction of class j:

$$E_{j,k} = |\max(\frac{\partial FC(\mathbf{v})_j}{\partial \mathbf{v}_k}, 0)| \times \mathbf{v}_k. \quad (1)$$

The partial derivative term shows how much the prediction of class j changes if the value from the k^{th} filter slightly changes. As we are finding the evidence for the target class j, we consider only the positive value of the derivative. Then $E_{j,k}$ combines this term with the strength of $\mathbf{v}_k$ to show the overall effect of the k^{th} filter for the input text. Next, we calculate the effect of each word w_i in the input text by aggregating the effects of all the n-grams containing w_i.

$$E_{j,w_i} = \sum_k (E_{j,k} \times \mathbb{I}[w_i \in N_k])$$

where N_k is an n-gram detected by the k^{th} filter. Lastly, we select, as the evidence, non-overlapping n-grams which are detected by at least one of the filters and have the highest sums of the effects of all the words they contain. For example, to decide whether we will select the n-gram N_k as an evidence text or not, we consider $\sum_{w_i \in N_k} E_{j,w_i}$. Note that we can find counter-evidence by changing, in equation (1), from $\max$ to $\min$.

4.3.2 Decision Trees

This explanation method is based on model extraction (Bastani et al., 2017). We create a decision tree (DT) which mimics the behavior of the classification part (fully-connected layers) of the trained CNN. Given a filter-based feature vector **v**, the DT needs to predict the same class as predicted by the CNN. Formally, we want

$$DT(\mathbf{v}) = \underset{j}{\operatorname{argmax}}\, \mathbf{p}_j = \underset{j}{\operatorname{argmax}}\, FC(\mathbf{v})_j.$$

For multi-class classification, we construct one DT for each class (one vs. rest classification). We employ CART with Gini index for learning DTs (Leo et al., 1984). All the training examples are generated by the trained CNN using a training dataset, whereas a validation dataset is used to prune the DTs to prevent overfitting.

Also, for each feature v_j in **v**, we calculate the Pearson's correlation between v_j and the output of each class (before softmax) in $FC(\mathbf{v})$ using the training dataset, so we know which class is usually predicted given a high score of v_j (i.e., correlated most to this feature). We use c_j denoting the most correlated class of the feature v_j.

We can consider the DTs as a global explanation of the model as it explains the CNN in general. To create a local explanation, we use the DT of the predicted class to classify the input. At each decision node, we collect associated n-grams passing the nodes' thresholds to be evidence for (or counter-evidence against) the predicted class (depending on the most correlated class of each splitting feature). For example, an input text X is classified to class a, so we use the DT of class a to predict the input. If a decision node checks whether feature v_j of this input is greater than 0.25 and assume it is true for this input, the n-gram corresponding to v_j will be evidence if the most correlated class of v_j is class a (i.e., $c_j = a$). Otherwise, it will be counter-evidence if $c_j \neq a$.

An example from the Amazon dataset, Actual: Pos, Predicted: Pos, (Predicted scores: Pos 0.514, Neg 0.486):
"*OK but not what I wanted: These would be ok but I didn't realize just how big they are. I wanted something I could actually cook with. They are a full 12" long. The handles didn't fit comfortably in my hand and the silicon tips are hard, not rubbery texture like I'd imagined. The tips open to about 6" between them. Hope this helps someone else know ...*"

Method	**Top-3 evidence texts**	**Top-3 counter-evidence texts**
LIME (W)	comfortably / wanted / helps	not / else / someone
LRP (W)	are / not / 6	: / tips / open
LRP (N)	are hard , not / about 6" between / not what I wanted	. The tips open / : These would / in my hand and
Grad-CAM-Text (N)	comfortably in my hand / I wanted : These / . The tips open	not what I wanted / not rubbery texture like / Hope this helps someone
DTs (N)	imagined . The tips	'd imagined . / are . I wanted / would be ok

Table 3: Examples of evidence and counter-evidence texts generated by some of the explanation methods.

4.4 Implementations[3]

We used public libraries of LIME[4], LRP (Alber et al., 2018), and DeepLIFT[5] in our experiments. Besides, the code for computing Grad-CAM-Text was adapted from keras-vis[6], whereas we used scikit-learn (Pedregosa et al., 2011) for decision tree construction. All the DTs achieved over 80% macro-F1 in mimicking the CNNs' predictions.

For the task parameters, we set $m = 3$, $\tau_h = 0.9$, and $\tau_l = 0.7$. For each task and dataset, we used 100 input texts, half of which were classified correctly by the model(s) and the rest were misclassified. So, with nine explanation methods being evaluated, each task had 900 questions per dataset for human participants to answer. Examples of questions for each task are given in Figure 2.

For the Amazon dataset, we posted our tasks on Amazon Mechanical Turk (MTurk). To ensure the quality of crowdsourcing, each question was answered by three workers and the scores were averaged. For the ArXiv dataset which requires background knowledge of the related subjects, we recruited graduates and post-graduate students in Computer Science, Mathematics, Physics, and Engineering to perform the tasks, and each question was answered by one participant. In total, we had 367 and 121 participants for the Amazon and the ArXiv datasets, respectively.

5 Results and Discussion

Examples of the generated explanations are shown in Table 3 and a separate appendix. Table 4 shows the average scores of each explanation method for each task and dataset, while Figure 3 displays the distributions of individual scores for all three tasks. We do not show the distributions of tasks 2 and 3 of the Amazon dataset as they look similar to the associated ones of the ArXiv dataset.

5.1 Task 1

For the Amazon dataset, though Grad-CAM-Text achieved the highest overall score, the performance was not significantly different from other methods including the random baselines. Also, the inter-rater agreement for this task was quite poor. It suggests that existing explanation methods cannot apparently reveal irrational behavior of the underfitting CNN to lay human users. So, the scores of most explanation methods distribute symmetrically around zero, as shown in Figure 3(a).

For the ArXiv dataset, LRP (N) and DeepLIFT (N) got the highest scores when both CNNs predicted correctly. Hence, they can help humans identify the poor model to some extent. However, there was no clear winner when both CNNs predicted wrongly. One plausible reason is that evidence for an incorrect prediction, even by a well-trained CNN, is usually not convincing unless we set a (high) lower bound of the confidence of the predictions (as we did in task 2).

Additionally, we found that psychological factors fairly affect this task. Based on the results, for two explanations with comparable semantic quality, humans prefer the explanation with more evidence texts and select it as more reasonable. This is consistent with the findings by Zemla et al. (2017). Conversely, the DTs method performed in the opposite way. The DT of the better model usually focuses on a few most relevant texts in the input and outputs fewer evidence texts. This possibly causes the low performance of the DTs method in this task. Also, we got feedback from the participants that they sometimes penalized an evidence

[3] The code and datasets of this paper are available at https://github.com/plkumjorn/CNNAnalysis

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

[5] https://github.com/kundajelab/deeplift

[6] https://github.com/raghakot/keras-vis

Explanation Method	Task 1						Task 2						Task 3					
	Amazon			ArXiv			Amazon			ArXiv			Amazon			ArXiv		
	$\mathcal{A}$	✔	✘	$\mathcal{A}$	✔	✘	$\mathcal{A}$	✔	✘	$\mathcal{A}$	✔	✘	$\mathcal{A}$	✔	✘	$\mathcal{A}$	✔	✘
Random (W)	.02	.00	.04	-.11	-.05	-.17	.06	.10	.02	.07	.09	.04	.05	.53	-.43	.01	.32	-.30
Random (N)	.02	.02	.02	-.12	-.16	-.07	.12	.13	.12	.29	.32	.25	-.01	.54	-.55	.02	.29	**-.25**
LIME (W)	-.02	.02	-.06	.03	.02	**.03**	**.69**	**.74**	.64	**.70**	**.75**	**.64**	.02	.50	-.45	-.02	.31	-.34
LRP (W)	.00	-.01	.02	-.03	-.01	-.05	.13	.26	-.01	.26	.36	.16	-.02	.50	-.54	-.06	.33	-.44
LRP (N)	-.07	-.04	-.09	**.12**	**.24**	-.01	.26	.45	.08	.44	.49	.39	.08	.60	-.43	**.17**	**.60**	-.26
DeepLIFT (W)	.04	.03	.04	.07	.13	.00	.21	.37	.04	.26	.35	.16	-.03	.47	-.53	-.08	.28	-.44
DeepLIFT (N)	.06	.06	**.05**	.06	.22	-.10	.23	.47	-.01	.38	.47	.28	.05	.59	-.49	.02	.33	-.30
Grad-CAM-T (N)	**.07**	**.11**	.03	-.03	-.04	-.01	.65	.64	**.66**	.53	.65	.41	.05	.51	-.42	.06	.56	-.45
DTs (N)	-.05	-.02	-.08	-.13	-.22	-.03	.64	.68	.59	.51	.69	.32	**.10**	**.60**	**-.40**	-.11	.29	-.50
Fleiss κ (Amazon)	0.050 / 0.054			N/A			0.274 / 0.371			N/A			0.212 / 0.499			N/A		

Table 4: The average scores of the three evaluation tasks. The score range is [-1,1] in which 1 is better. $\mathcal{A}$, ✔, and ✘are for all, correctly classified, and misclassified input texts, respectively. Boldface numbers are the highest average scores in the columns. A number is underlined when there is no statistically significant difference between the scores of the corresponding method and the best method in the same column (at a significance level of 0.05). The last row reports inter-rater agreement measures (Fleiss' kappa) in the format of α / β where α considers answers with human confidence levels (5 categories for task 1-2 and 4 categories for task 3) and β considers answers regardless of the human confidence levels (3 categories for task 1-2 and 2 categories for task 3).

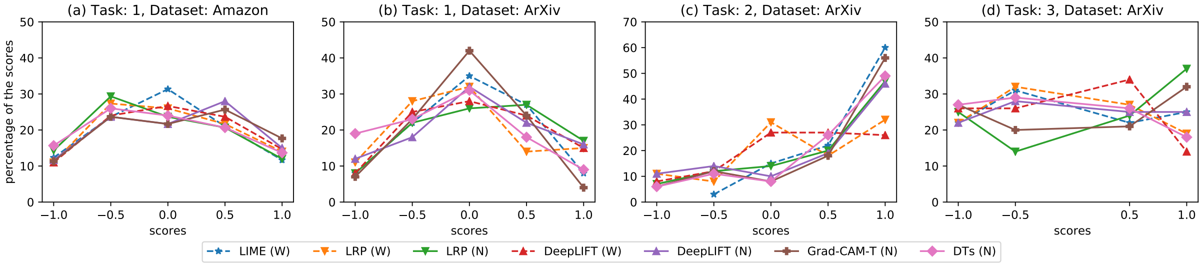


Figure 3: Score distributions from task 1 of the Amazon dataset and from all the tasks of the ArXiv dataset.

text which is highlighted in a strange way, such as "... greedy algorithm. In this paper, we ...". Hence, in real applications, syntax integrity should be taken into account to generate explanations.

5.2 Task 2

LIME clearly achieved the best results in task 2 followed by Grad-CAM-Text and DTs. These methods are class discriminative, being able to find good evidence for the predicted class regardless of whether the prediction is correct.

We believe that LIME performed well because it tests that the missing of evidence words from the input text greatly reduces the probability of the predicted class, so these words are semantically related to the predicted class (given that the model is accurate). Meanwhile, the DTs method selects evidence based on the most correlated class of the splitting features. So, the evidence n-grams are more likely related to the predicted class than the other classes. However, they may be less relevant than LIME's as the evidence is generated from a global explanation of the model (DTs). Besides, Grad-CAM-Text worked relatively well here probably because it preserves the class discriminative property of Grad-CAM (Selvaraju et al., 2017).

By contrast, LRP and DeepLIFT generated acceptable evidence only for the correct predictions. Also, LRP (N) and DeepLIFT (N) performed better than LRP (W) and DeepLIFT (W) in both datasets. This might be because one evidence n-gram contains more information than one evidence word. Nevertheless, even the Random (N) method surpasses the LRP (W) and the DeepLIFT (W) for the ArXiv dataset. Thereby, whenever we use LRP and DeepLIFT, we should present to humans the most relevant words together with their contexts.

5.3 Task 3

The negative scores under the ✘columns of task 3 show that using explanations to rectify the predictions is not easy. Hence, the overall average scores of many explanation methods stay close to zero.

DTs performed well only on the Amazon dataset. The average numbers of n-grams per explanation, generated by the DTs, are 2.00 and 1.77

for the Amazon and ArXiv datasets, respectively. Also, the reported n-grams could be repetitive and overlapping. This reduced the amount of useful information displayed, and it may be insufficient for humans to choose one of the CS, MA, and PH categories, which are more similar to one another than the positive and negative sentiments.

Meanwhile, LRP (N) performed consistently well on both datasets. This is reasonable considering our discussions in task 2. First, LRP (N) generates good evidence for correct predictions, so it can gain high scores in the ✔columns. On the other hand, the evidence for incorrect predictions (✘) is usually not convincing, so the counter-evidence (which is likely to be the evidence of the correct class) can attract humans' attention. Furthermore, the fact that LRP is not class discriminative does not harm it in this task as humans can recognize an evidence text even if it is selected by the LRP (N) as counter-evidence (and vice versa).

For example, in the ArXiv dataset, we found a case in which the predicted class is PH (score = 0.48) but the correct class is CS (score = 0.07). LRP (N) selected 'armed bandit settings with', 'the Wasserstein distance', and 'derive policy gradients' as evidence for the class PH. These n-grams, however, are not truly related to PH. Rather, they revealed the true class of this text and made a human choose the CS option with high confidence despite the low predicted score.

Regarding LIME, the situation is reversed as LIME can find both good evidence and counter-evidence. These make humans be indecisive and, possibly, select a wrong option as the explanation is presented at a word level (without any contexts).

5.4 Model Complexity

Apart from the results of the three tasks, it is worth to discuss the size of the DTs which mimic the four CNNs in our experiments. As shown in Table 5, the size of the DTs can reflect the complexity of the CNNs. Although the well-trained CNN of the Amazon dataset got 0.9 F1 score, the DTs of this CNN needed more than 5,500 nodes to achieve 85% fidelity (compared to only hundreds of nodes required for the ArXiv dataset). This illustrates the high complexity of the Amazon task compared to the ArXiv task even though both tasks were managed effectively by the same CNN architecture.

For the ArXiv dataset, the DTs of the poor CNN are smaller than the ones of the well-trained CNN. This is likely because the poor CNN was trained on a specific dataset (i.e., selected subtopics of the main categories), so it had to deal with fewer discriminative patterns in texts compared to the first CNN trained using texts from all subtopics.

	#Nodes	Depth	#Leaves
Amazon: 1st CNN (well-trained) – *F* = 0.85			
Negative	5535	38	2768
Positive	5537	45	2769
Amazon: 2nd CNN (underfitting) – *F* = 0.82			
Negative	6405	40	3203
Positive	6369	40	3185
ArXiv: 1st CNN (well-trained) – *F* = 0.89			
Computer Science	363	25	182
Mathematics	565	24	283
Physics	325	24	163
ArXiv: 2nd CNN (specific data) – *F* = 0.84			
Computer Science	107	17	54
Mathematics	263	28	132
Physics	237	29	119

Table 5: Metadata of the DTs in the experiments. *F* refers to fidelity of the DTs (macro-average F1).

Further studies of this quality of the DTs would be useful for some applications, e.g., measuring model complexity (Bianchini and Scarselli, 2014) and model compression (Cheng et al., 2018).

6 Conclusion

We proposed three human tasks to evaluate local explanation methods for text classification. Using the tasks in this paper, we experimented on 1D CNNs and found that *(i)* LIME is the most class discriminative method, justifying predictions with relevant evidence; *(ii)* LRP (N) works fairly well in helping humans investigate uncertain predictions; *(iii)* using explanations to reveal model behavior is challenging, and none of the methods achieved impressive results; *(iv)* whenever using LRP and DeepLIFT, we should present to humans the most relevant words together with their contexts and *(v)* the size of the DTs can also reflect the model complexity. Lastly, we consider evaluating on other datasets and other advanced architectures beneficial future work as it may reveal further interesting qualities of the explanation methods.

Acknowledgments

We would like to thank all the participants in the experiments as well as Alon Jacovi and anonymous reviewers for helpful comments. Also, the first author would like to thank the support from Anandamahidol Foundation, Thailand.

References

Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. 2018. innvestigate neural networks! *arXiv preprint arXiv:1808.04260*.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in nlp. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7. Association for Computational Linguistics.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017a. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142.

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017b. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the EMNLP'17 Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA)*, pages 159–168. Association for Computational Linguistics.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015a. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015b. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. 2017. Interpretability via model extraction. *CoRR*, abs/1706.09773.

Monica Bianchini and Franco Scarselli. 2014. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, 25(8):1553–1565.

Or Biran and Kathleen R McKeown. 2017. Human-centric justification of machine learning predictions. In *IJCAI*, pages 1461–1467.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2018. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*, 35(1):126–136.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116. Association for Computational Linguistics.

Yannis Dimopoulos, Paul Bourret, and Sovan Lek. 1995. Use of some sensitivity criteria for choosing networks with good generalization ability. *Neural Processing Letters*, 2(6):1–4.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90. Association for Computational Linguistics.

Reza Ghaeini, Xiaoli Fern, and Prasad Tadepalli. 2018. Interpreting recurrent and attention-based neural models: a case study on natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4952–4957, Brussels, Belgium. Association for Computational Linguistics.

Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 56–65. Association for Computational Linguistics.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112. Association for Computational Linguistics.

Been Kim, Cynthia Rudin, and Julie Shah. 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 1952–1960, Cambridge, MA, USA. MIT Press.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117. Association for Computational Linguistics.

Breiman Leo, Jerome H Friedman, Richard A Olshen, and Charles J Stone. 1984. Classification and regression trees. *Wadsworth International Group*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5570–5581. Association for Computational Linguistics.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774.

Sina Mohseni and Eric D Ragan. 2018. A human-grounded evaluation benchmark for local explanations of machine learning. *arXiv preprint arXiv:1801.05075*.

Dong Nguyen. 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)*, pages 1069–1078.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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

Nina Poerner, Hinrich Schütze, and Benjamin Roth. 2018. Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 340–350.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA. ACM.

Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2018. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services*, 1(1):39–48.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153, International Convention Centre, Sydney, Australia. PMLR.

David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503.

Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2009. Moviexplain: a recommender system with explanations. *RecSys*, 9:317–320.

Wenting Xiong, Iftitahu Ni'mah, Juan MG Huesca, Werner van Ipenburg, Jan Veldsink, and Mykola Pechenizkiy. 2018. Looking deeper into deep learning model: Attribution-based explanations of textcnn. *arXiv preprint arXiv:1811.03970*.

Jeffrey C Zemla, Steven Sloman, Christos Bechlivanidis, and David A Lagnado. 2017. Evaluating everyday explanations. *Psychonomic bulletin & review*, 24(5):1488–1500.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1031–1040. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA. MIT Press.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929.