Learning Credible Deep Neural Networks with Rationale Regularization

Mengnan Du, Ninghao Liu, Fan Yang, Xia Hu
Department of Computer Science and Engineering, Texas A&M University
{dumengnan, nhliu43, nacoyang, xiahu}@tamu.edu

***Abstract*—Recent explainability related studies have shown that state-of-the-art DNNs do not always adopt correct evidences to make decisions. It not only hampers their generalization but also makes them less likely to be trusted by end-users. In pursuit of developing more credible DNNs, in this paper we propose CREX, which encourages DNN models to focus more on evidences that actually matter for the task at hand, and to avoid overfitting to data-dependent bias and artifacts. Specifically, CREX regularizes the training process of DNNs with rationales, *i.e.*, a subset of features highlighted by domain experts as justifications for predictions, to enforce DNNs to generate local explanations that conform with expert rationales. Even when rationales are not available, CREX still could be useful by requiring the generated explanations to be sparse. Experimental results on two text classification datasets demonstrate the increased credibility of DNNs trained with CREX. Comprehensive analysis further shows that while CREX does not always improve prediction accuracy on the held-out test set, it significantly increases DNN accuracy on new and previously unseen data beyond test set, highlighting the advantage of the increased credibility.**

***Index Terms*—Deep neural network; Explainability; Credibility; Expert rationales**

I. Introduction

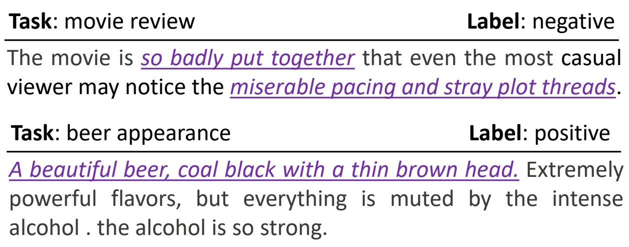
There has been an increasing interest recently in developing explainable deep neural networks (DNNs) [1]–[4]. To this end, a DNN model should be able to provide intuitive explanations for its predictions. Explainability could shed light into the decision making process of DNNs and thus increase their acceptance by end-users. However, explainability alone is insufficient for DNNs to be *credible* [5], unless the provided explanations conform with the well-established domain knowledge. That is to say, correct evidences should be adopted by the networks to make predictions. The incredibility issue has been observed in various DNN systems. For instance, in question answering (QA) tasks, DNNs rely more on function words rather than pay attention to task-specific verbs, nouns and adjectives to make decisions [6], [7]. Similarly, in image classification, CNNs may make decisions solely according to background within images, rather than paying attention to evidences relevant to the objects of interest [8].

In this work, we define credible DNNs as the models that could provide explanations to their predictions, while at the same time the explanations are consistent with the well-established domain knowledge. Considering that correct evidences are employed in decision making process, it would be easier for credible DNNs to build up trust among practitioners and end-users. In addition, credible DNNs could have better generalization capability comparing to untrustable ones. Since credible DNNs have truly grasped useful knowledge instead of memorizing unreliable *dataset-specific biases* and *artifacts*, they could maintain high prediction accuracy for those unseen data instances beyond the training dataset.

It is possible to enhance the credibility and generalization of DNNs from two perspectives: dataset and model training. The former category tackles this problem by constructing datasets with larger quantity and higher quality. Any training data may contain some biases, either intrinsic noise or additional signals inadvertently introduced by human annotators [9]. DNNs not only rely on these biases to make decisions, but also could amplify them [10], which partly leads to the low credibility and low generalization problem. Some work has developed debiased datasets either by filtering out bias data, or constructing new datasets in an adversarial manner [11]. Nevertheless, this scheme cannot fully eliminate bias, which still could affect model performance. The second category aims at regulating the training of DNNs using domain knowledge established by humans. This is motivated by the observation that purely data-driven learning could lead to counter-intuitive results [12]. Thus it is desirable to combine DNNs with the domain knowledge that humans utilize to understand the world, which has been proven beneficial in a lot of learning problems [12]–[14]. Therefore, we follow the second strategy using domain knowledge to enhance the credibility of DNNs.

Nevertheless, regulating the training of DNNs with domain knowledge to promote model credibility is still a technically challenging problem. First, one difficulty lies in how to accurately obtain and effectively utilize DNNs' attention towards input features. Although DNN local explanations could identify the contributions of each input feature towards a specific model prediction [8], it is still challenging to incorporate explanation into the end-to-end back-propagation procedure to influence model parameter update. The second challenge is how to use domain knowledge to regularize the models' attention and force models to focus on correct evidences. Previous work have demonstrated that domain knowledge is beneficial in terms of promoting prediction accuracy of DNNs. For instance, structured knowledge in the form of logical rules can be transferred to the weights of DNNs through iterative distillation process [12]. However, it is still unclear how to utilize knowledge to guide the attention of a DNN.

To overcome the above challenges, we propose to explore whether a specific kind of domain knowledge, called *rationale*,



Task: movie review **Label**: negative

The movie is *so badly put together* that even the most casual viewer may notice the *miserable pacing and stray plot threads*.

Task: beer appearance **Label**: positive

A beautiful beer, coal black with a thin brown head. Extremely powerful flavors, but everything is muted by the intense alcohol . the alcohol is so strong.

Fig. 1: Two examples of expert rationale: words marked with purple color, for movie review and product review respectively.

would be useful in terms of enhancing DNN credibility. A rationale is a subset of features highlighted by annotators and regarded to be more important in predicting an instance [15], [16], with illustrative examples shown in Fig. 1. The rationales are utilized to direct the model's attention, enabling it to tease apart useful evidence from noises and pushing it to pay more attention to relevant features. Rationales have been applied to the training process of SVMs [15], [17] to enhance predictive performance. Another benefit of rationales is that they require little effort to obtain [18], thus they are possible to be widely applied in different applications.

In this work, we propose CREX (CRedible EXplanation), an approach regularizing DNNs to utilize correct evidences to make decisions, in order to promote their credibility and generalization capability. The intuition behind CREX is to use external knowledge to regulate the DNN training process. For those training instances coupled with expert rationales, we require the DNN model to generate local explanations that conform with the rationales. Even when expert rationales are not available, CREX can still promote model performance by requiring the generated explanations to be sparse. Through experiments on text classification tasks, we demonstrate that our trained DNNs generally rely on correct evidences to make predictions. Besides, our trained DNNs generalize much better on new and previously unseen inputs beyond test set. The major contributions of this paper are summarized as follows:

- We propose a method to regularize the training of DNNs, called CREX, which aims to enable trained DNNs to focus on correct evidences to make decisions.
- CREX is widely applicable to different variants of DNNs. We demonstrate its applicability via three standard architectures, including CNN, LSTM and self-attention model.
- Experimental results on two text classification datasets validate that our trained DNNs could generate explanations aligning well with expert rationales and show good generalization properties on data beyond test set.

II. Related Work

In this section, we briefly present reviews for several research areas closely relevant to our work.

A. DNN Interpretability

DNNs are often regarded as black-boxes and criticized by the lack of interpretability. Towards this end, there is a wide range of work targeting to derive explanations and shed some insights into the decision making process of DNNs [2], [19]. These work can be grouped into two main categories: global and local explanation, depending on whether the goal is to understand how the DNN works globally or how DNN makes a specific prediction [1]. Most current work focus on augmenting DNNs with interpretability [8], [20], [21], while employing explanation to enhance the performance of DNN models has seldom been explored. In this work, we aim to take advantage of DNN local explanation to promote the generalization performance of DNN classifiers.

B. Model Credibility and Generalization

Despite the high performance of DNN models on test set, recent work shows that these models heavily rely on dataset bias instead of true evidences to make decisions [22]. For instance, a DNN local explanation approach analyzes three question answering models, showing that these models often ignore important part of the questions, e.g., verbs in questions carry little influence for the DNN decisions, and rely on irrelevant words to make decisions [6]. Similarly, for binary husky and wolf classification task, the CNN simply makes decisions according to whether there is snow within an image or not, rather than pays attention to evidences relevant to animals [8]. This makes the DNN models unreliable and hampers their generalization. In addition, this also makes these models fragile and easily broken by adversarial samples.

C. Unwanted Dataset Bias

Datasets may contain lots of unwanted bias and artifacts, either explicit ones, e.g., gender and ethnic biases, or implicit ones. DNNs not only rely on these biases to make decisions, but also could amplify them [10], which partly lead to the low credibility and low generalization of DNNs on unseen data. In order to alleviate the influence of unwanted dataset bias to models' performance, one line of work tackles this problem by regulating the training of models [23], [24], while some others consider to construct more challenging datasets by eliminating biases and annotation artifacts [11], [25].

D. Combining Human Knowledge with DNNs

Some work enhances DNN models with human-like common sense to make them more credible and robust. For instance, the attention of RNN is regularized with human attention values derived from eye-tracking corpora [26]. Structured knowledge such as logical rules are transfered to the weight of DNNs through iterative distillation process [12]. Besides, rationales are augmented to the training process of CNN models [14], linear classification model [5], and SVMs [17]. These work indicates that human knowledge has indeed promoted the credibility models to some extent. The most similar work to ours is using human rationales to improve neural predictions [27]. However, their work is exclusively designed to regularize the intrinsically interpretable model, i.e., attention model. In contrast, our method is widely applicable to different network architectures, including both interpretable models and black-box models, such as CNN and LSTM.

III. Problem Statement

In this section, we first introduce the basic notations used in this paper. Then we present the problem of learning credible deep neural network models.

Notations: Consider a typical multi-class text classification task. Given a training dataset which consists of N instances: $D = \{(x_1, y_1), ...(x_N, y_N)\}$. Each input text x_n is composed of a sequence of T words: $x_n = \{x_n^{(1)}, ..., x_n^{(T)}\}$, where $x_n^{(t)} \in \mathbb{R}^d$ denotes the embedding representation of the t-th word. Each $y_n \in \{1, 2, ..., C\}$ belongs to one of the C output classes. Part of the training data, with a number of N_r, contains not only input-label pairs (x_n, y_n), but also rationale r_n from domain expert, with two illustrate examples shown in Fig. 1. Each entry of the expert rationale $r_n^{(t)} \in \{0, 1\}$, where 1 indicate that word $x_n^{(t)}$ is actually responsible for the prediction task, and vice versa.

Learning Credible DNNs: The goal is to learn a DNN-based classification model which maps a text input x_n to the probability output $f(x_n)$. We expect a trained DNN to rely on correct evidences to make decisions and pay more attention to words within the rationales. That is, for a trained DNN, the generated local explanation for each testing instance should align well with expert rationales.

IV. Proposed CREX Framework

In this section, we introduce the CREX framework, which aims to regularize the local explanation when training a DNN for the task of interest, so as to promote its credibility and generalization. Besides feeding labels as supervised signals, we also enforce the explanations of the DNN predictions to conform with expert rationales, and encourage the explanations to be sparse if rationales are absent. In this way, the trained network could make predictions based on the correct evidences that we expect it to focus on.

A. Augmenting Local Explanation

The general idea of DNN local explanation is to attribute the prediction of a DNN to its input, producing a heatmap indicating the contribution of each feature in the input to the prediction. There are several key desiderata for the augmented local explanation method in this work:

- *Faithful*: The provided explanations should be of high fidelity with respect to predictions of the original model.
- *Differentiable*: We expect the explanation method to be end-to-end differentiable, amenable for training with back-propagation and updating DNN parameters.
- *Model-agnostic*: It is desirable that the explanation method to be agnostic to network architectures, and thus generally applicable to different networks, e.g., CNNs and LSTMs.

The explanation of prediction $f(x_n)$ for input x_n is a matrix $s_n \in \mathbb{R}^{T \times C}$, where $s_n^{(t,c)}$ denotes the contribution of word $x_n^{(t)}$ towards prediction $f_c(x_n)$ for output class c. We utilize an omission based method [28] to measure the contribution of $x_n^{(t)}$, denoted as below:

$$s_n^{(t,c)} = f_c(x_n) - f_c(x_n^{(\backslash t)}), \tag{1}$$

which quantifies the deviation of the prediction between the original input x_n and the partial input $x_n^{(\backslash t)} = x_n^{(1:t-1)} \oplus x_n^{(t+1:T)}$ with $x_n^{(t)}$ omitted. The motivation is that more important features, once being changed, will cause more significant variation to the prediction score. It is worth noting that the omission operation may lead to invalid input, which could trigger the adversarial side of DNNs. To reflect model behaviors under normal conditions, phrase omission is conducted instead of individual word omission. Formally, we compute the contribution of $x_n^{(t)}$ by averaging the prediction changes of deleting different length-m phrases that contain $x_n^{(t)}$:

$$s_n^{(t,c)} = \frac{1}{m} \sum_{j=1}^{m} [f_c(x_n) - f_c(x_n^{(1:t-1-m+j)} \oplus x_n^{(t+j:T)})]. \tag{2}$$

For long text classification, such as documents, we segment each original text into sentences and sequentially perform omission for each sentence. In such scenario, sentence-level contribution scores are obtained as explanation, rather than word-level scores. Both phrase omission and sentence omission could increase the faithfulness of explanation, compared with directly removing individual words [29].

B. Aligning Explanations with Rationales

The key idea of CREX is that DNNs should rely on reasonable evidences to make decisions rather than bias or artifacts. We encourage the explanation to align well with expert rationales when they are available, by considering two complementary conditions as follows. First, for the original input, we encourage the generated explanation to be *confident* and focus on the relevant features as indicated by rationales. Second, for the negative input, where the important features are suppressed, the explanation should be *uncertain* and have relatively uniform contribution across classes.

1) Confident Explanation: We first feed original input x_n to DNN and get model output $f(x_n)$ and explanation s_n. The rationale r_n points out which subset of features is important and the rest to be irrelevant. Intuitively, we achieve credibility by encouraging dense contribution scores on known important factors and encouraging sparse contribution scores on the remaining irrelevant features. We define a *confident explanation* loss (g_{conf}), which encourages the explanation to concentrate on rationales:

$$g_{conf}(x_n) = \frac{1}{C} \sum_{c=1}^{C} ||(1 - r_n) \odot s_n^{(:,c)})||_1. \tag{3}$$

The loss aims to shrink the contribution scores of irrelevant features, in order to discourage models from capturing training data specific biases. An implicit effect of this loss is to encourage f to give dense explanation scores to the relevant features, thus making f pay more attention to them. As a result, the final explanation scores tend to aligning well with

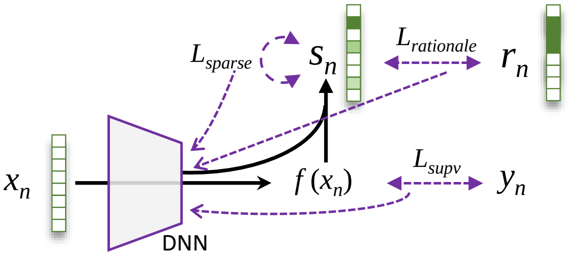


Fig. 2: Schematic of CREX. Black solid lines denote forward pass. Dashed line with arrows on both ends are losses. Dashed line with arrows on one side denote flow of gradients. Three vectors from left to right are input, explanation and rationale, respectively. CREX is DNN architecture agnostic, end-to-end trainable, and simple to implement.

rationales. In addition, we observe that summing all categories $\{1, ...C\}$ could yield better results comparing to only using label y_n when imposing confident explanation regularization to instance x_n.

2) Uncertain Explanation: When the subset of important features, as indicated in r_n, is deleted in the original input x_n, we expect the DNN model to become uncertain about which category to output. This kind of inputs, named as negative inputs, are generated as the Hadamard product between the original input x_n and the reversed rationale vector $(1 - r_n)$:

$$x'_n = x_n \odot (1 - r_n). \quad (4)$$

For instance, the negative input corresponding to the first input in Fig. 1 is *"The movie is that even the most casual viewer may notice the"*. The intuition is that after feeding the negative input x'_n to a DNN model, we expect its probability output for ground truth label y_n to be much smaller comparing to the probability value of original input x_n, since x'_n lacks the evidence supporting the prediction. At the same time, the contributions of different words/sentences should be distributed uniformly. Its implicit effect is to encourage the DNN model to give lower explanation scores to the features not belonging to rationales. We first calculate the absolute value of explanation for x'_n as $\hat{s}_n^{(:,y_n)} = |s_n^{(:,y_n)}|$, and then normalize it as:

$$e_n^{(t,y_n)} = \hat{s}_n^{(t,y_n)} / \sum_{k=1}^{T} \hat{s}_n^{(k,y_n)} \quad (5)$$

The resultant $e_n^{(:,y_n)}$ can be seen as the soft-attention assigned by DNN for x'_n. After that, we define an *uncertain explanation* loss (g_{unc}):

$$g_{unc}(x'_n) = -|f_{y_n}(x_n) - f_{y_n}(x'_n)| - \alpha \, \cos(e_n^{(:,y_n)}, q), \quad (6)$$

where q is the discrete uniform distribution denoted as $\mathcal{U}(1, T)$, and α is used to balance probability output and explanation distribution. The cosine similarity is employed to encourage explanation scores to be distributed uniformly.

We linearly combine the two loss functions at hand, and calculate the average value over all training instances with rationales as the *explanation rationale* loss, formulated as follows:

$$\mathcal{L}_{rationale} = \frac{1}{N_r} \sum_{n=1}^{N_r} [g_{conf}(x_n) + \beta g_{unc}(x'_n)]. \quad (7)$$

Parameter β is utilized to balance the confident explanation and uncertain explanation. By encouraging confident explanations to conform with rationales in original input x_n, and suppressing the probability output as well as explanation values in a negative input x'_n, $\mathcal{L}_{rationale}$ regulates a DNN to learn useful input representations from features belonging to rationales and omit information in the irrelevant feature subset.

Algorithm 1: Learning credible DNNs.

```
Input: Training data $D = \{(x_n, y_n)\}_{n=1}^{N}$, validation data
       $D_v = \{(x_n, y_n)\}_{n=1}^{N_v}$, and rationales $\{r_n)\}_{n=1}^{N_r}$.
Set hyperparameters $\alpha, \beta, \lambda_1, \lambda_2$, learning rate $\eta$,
  iteration number $max\_iter = 10$, sample index
  $i \in \{1, ..., n\}$, and epoch index $t = 0$;
Initialize DNN parameters $\mathbf{W}$;
while $t \leq max\_iter$ do
    $\mathcal{L}_{supv} = \frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} -\mathbb{1}(y_n = c) \cdot \log(f_c(x_n))$;
    $\mathcal{L}_{rationale} = \frac{1}{N_r} \sum_{n=1}^{N_r} g_{conf}(x_n) + \beta g_{unc}(x'_n)$;
    $\mathcal{L}_{sparse} = \frac{1}{(N-N_r) \cdot C} \sum_{n=N_r+1}^{N} \sum_{c=1}^{C} ||s_n^{(:,c)}||_1$;
    $\mathcal{L}(\theta, x, y, r) = \mathcal{L}_{supv} + \lambda_1 \mathcal{L}_{rationale} + \lambda_2 \mathcal{L}_{sparse}$
    $\mathbf{W}_{t+1} = Adam(\mathcal{L}(\theta, x, y, r), \eta)$;
    Get DNN accuracy on validation set $D_v$;
    $t = t + 1$;
Output: DNN $f$ with best accuracy on validation set.
```

C. Self-guidance When Rationale not Available

In last section, given expert rationales, we render the local explanation of each instance to conform with its rationale. However, expert rationales may not always be available. In practice, the experts may only annotate a small ratio of training data. This could be done either when annotating a new corpus, or when adding rationales post-hoc to an existing corpus. To guide the DNN model to focus on correct evidences in such scenario, we enforce the generated local explanation vector to be sparse for training instances without rationales. Simpler explanations are more credible, otherwise the dense dependencies could make it hard to disentangle the patterns in the input that actually trigger a prediction [30]–[32]. To achieve this, we propose the *sparse explanation* loss for those instances without rationales, denoted as follows:

$$\mathcal{L}_{sparse} = \frac{1}{(N - N_r) \cdot C} \sum_{n=N_r+1}^{N} \sum_{c=1}^{C} ||s_n^{(:,c)}||_1, \quad (8)$$

where the ℓ_1 norm helps produce sparse contribution vectors. Note that this summation is performed over the $(N$-$N_r)$ instances which have no rationales.

D. CREX Training

Besides regularizing the local explanations for DNN predictions, we also expect the DNN model to learn from the ground truth labels, which is defined using supervised cross-entropy loss function as follows:

$$\mathcal{L}_{supv} = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} -\mathbb{1}(y_n = c) \cdot \log(f_c(x_n)). \tag{9}$$

Our final model is learned by balancing the supervised approximation to the labels and the conformation to expert rationales. We propose the training objective by jointly minimizing the losses as below:

$$\mathcal{L}(\theta, x, y, r) = \mathcal{L}_{supv} + \lambda_1 \mathcal{L}_{rationale} + \lambda_2 \mathcal{L}_{sparse}. \tag{10}$$

Parameters λ_1 and λ_2 are utilized to balance the supervised loss, rationale loss and sparse loss. For those N_r inputs coupled with expert rationales, we impose rationale loss, while for the rest $N - N_r$ inputs we regularize them with sparse loss. The overall idea of CREX is illustrated in Fig. 2, and the learning algorithm of CREX is presented in Algorithm 1. Our framework is designed to train the DNN model which could make highly accurate predictions (the first term) as well as make decisions by relying on the correct evidences (the last two terms). In addition, our CREX training framework can be treated as knowledge distillation process that transfers expert knowledge from rationales to DNN parameters in order to yield more credible models. CREX is also general, and can be added to any DNN models, e.g., CNNs and LSTMs, in order to enhance model's credibility.

V. Experiments

In this section, we evaluate the proposed CREX framework on several real-world datasets and present experimental results in order to answer the following four research questions.

- **RQ1** - Does CREX enhance the credibility of DNNs by regularizing the local explanation using expert rationales in the training process?
- **RQ2** - Does CREX promote the generalization of DNNs when processing unseen instances, especially for those data beyond test set?
- **RQ3** - How do CREX components and hyperparameters affect DNNs' performance?
- **RQ4** - How do the quantity and quality of expert rationales influence the performance of DNNs trained by CREX?

A. Experimental Setup

In this section, we introduce the overall setup of the experiments, including: I. DNN architectures, II. datasets, III. baseline methods, and IV. implementation details.

1) DNN Architectures: We consider three representative DNN architectures for text classification, including CNN [33], LSTM [34], and Self-attention model [35].

CNN: This is a 2-D convolutional network. The convolution operation is performed on embedding input $\{x_n^{(1)}, ..., x_n^{(T)}\}$ using three sizes of kernel: [2, 3, 4]. We will use ReLU activation after the convolution operation and then apply max pooling operation for every channel. Finally, the resulting tensors will be concatenated as final input representation.

Dataset	Train	Dev	Test	Text length
Movie Review (MR)	1,500	100	200	794
Product Review (PR)	4,000	473	1,700	113

TABLE I: Dataset statistics of MR and PR dataset, including number for training, development and test set, as well as average text length.

LSTM: After feeding the input $x_n = \{x_n^{(1)}, ..., x_n^{(T)}\}$ to the LSTM model, T hidden state vectors $\{h_n^{(1)}, ..., h_n^{(T)}\}$ are obtained. The dimension of each hidden state vector is 150. Max pooling is performed after all T hidden vectors to obtain the final input representation.

Self-attention: A bidirectional LSTM is first utilized to learn input representations with hidden size of 300. Then the self-attention mechanism is applied on top of LSTM representations to produce a matrix embedding of the input sentence. This matrix contains 10 embeddings, where every embedding represents an encoding of the input sentence but giving an attention to a specific part of the sentence. These embeddings are concatenated as the final input representation.

For all three networks, after transforming variable length sentences into fixed size representations, fully connected layers are added after the representations to get logits [36] for multiple output classes. Finally, a softmax layer is added to convert logits to probability outputs.

2) Datasets and Rationales: We consider two benchmark text classification datasets. Both datasets are randomly split into training, development and test set, the statistics of which are reported in Tab. I.

Movie Review Dataset (MR): It is a binary sentiment classification dataset with movie reviews from IMDB [37]. Originally, this dataset is obtained by crawling movie reviews from the Internet Movie Database (IMDB), consisting 1000 positive and 1000 negative movie reviews [37]. Zaidan et al. [15] supplemented this dataset rationales for 1800 documents [1]. The rationales used in this dataset are sub-sentential snippets with a higher relevance for prediction task [2], with illustrative example shown in Fig. 1. The average length per rationale for per input text is 125, while the average text length is 794. Comparing to the whole text, the rationale is sparse.

Product Review Dataset (PR): It is a multi-aspect beer review dataset [38] with data derived from BeerAdvocate [3]. This dataset contains reviews for three aspects of beer: appearance, aroma and palate, where we only distinguish appearance. Originally this dataset contains reviews with rating in the range of $[0, 1]$. Similar to [27], we consider this as binary classifi-

[1] http://www.cs.jhu.edu/~ozaidan/rationales/

[2] In terms of the rationale collection process, the agreement among different annotators, as well as the time complexity of rationale annotations, we refer interested readers to the work by Zaidan et al. [15].

[3] https://www.beeradvocate.com/

	MR			PR		
Models	CNN	LSTM	Atten	CNN	LSTM	Atten
Vanilla DNN	2.86	2.67	2.40	3.96	3.77	3.73
Data Augment	2.75	3.20	2.29	3.85	3.70	4.16
Rationale Augment	2.52	2.45	2.25	3.65	3.61	3.59
CREX	**2.24**	**2.38**	**1.91**	**3.52**	**3.54**	**3.15**
Parameter λ_1	5e-2	1e-3	2e-4	1e-4	2e-4	1e-4
Parameter α	0.2	0.5	0.3	0.5	0.3	0.5

TABLE II: Credibility statistical comparisons of three DNN architectures on MR and PR test set, and corresponding optimal hyperparameter settings.

	MR			PR		
Models	CNN	LSTM	Atten	CNN	LSTM	Atten
Vanilla DNN	93.7	93.2	**94.7**	**94.9**	94.5	94.3
Data Augment	91.0	88.3	90.1	94.7	94.5	93.9
Rationale Augment	**94.0**	94.2	93.8	94.3	95.1	94.1
CREX	93.8	**94.3**	94.5	94.2	**94.8**	**94.5**

TABLE III: Accuracy comparisons (in percent) of CREX and baseline methods for three DNN architectures on MR and PR test set.

cation task, by labelling ratings $\leq$0.4 as negative category, while labeling those $\geq$0.6 as positive category. Rationales are provided by [16], which are also sub-sentential snippets indicating higher relevance for prediction (see Fig. 1). The rationale within this dataset is also sparse, with an average length of 19, comparing to average text length of 113.

3) Baseline Methods: We evaluate effectiveness of CREX by comparing it with three baseline approaches.

- *Vanilla DNN*: This is the most typical way to train DNN for text classification tasks. DNN models are trained with only standard cross entropy loss, optimizing parameters to minimize Eq. (9).
- *Data Augmentation*: Back translation is an effective data augmentation method to boost model performance, e.g., machine translation [39], [40]. The original text is first translated to an intermediate language (we use German) and then translated back to English via the Google Translate API [4]. The motivation is to use synonym replacement and sentence paraphrase to avoid overfitting to functional words.
- *Rationale Augmentation*: Expert rationales are extracted from the original text as additional training instances. These data are incorporated with original training data, resulting a final training dataset of double size comparing with original one. The intuition is to explicitly push DNNs to focus on rationales to make decisions.

4) Implementation Details: We use the pre-trained 300-dimensional word2vec [5] word embedding [41] to initialize the embedding layer for all three architectures. For those words that do not exist in word2vec, their embedding vectors are initialized with some random values. We tune the learning rate over the range $\{1e\text{-}4, 1e\text{-}3, 1e\text{-}2, 1e\text{-}1\}$ and utilize Adam optimizer [42] to optimize these models. For each model, all hyperparameters are tuned using the development set, according to the accuracy and credibility performance. Optimal values of α and λ_1 for different models are listed in Tab. II, while β and λ_2 are fixed as 1 and $1e$-5 respectively for all models. To avoid overfitting, we apply dropout to fully connected layers for all DNN models [43]. We implement all DNN models using the PyTorch library. Each model is trained for ten epoches and the one with the best performance on the development set is selected as the final model. In our experiments, all DNN models could converge within 10 epoches, and increasing the number may lead to overfitting. Besides, since all models use random initialization, which leads to variance in performances at different runs. Therefore, we report the average values over three runs for all DNNs in the following experiments.

B. Credibility and Accuracy on Test Set

In this section, we evaluate the performance of all trained DNNs on test set. Two metrics are employed for evaluation: credibility and prediction accuracy. The credibility here is defined as the extent of agreement between the generated DNN local explanations and expert rationales.

1) Quantitative Evaluation of Credibility: To measure credibility, we calculate the matching degree between local explanation of DNN prediction with rationale. Specifically, We use the symmetric KL divergence between the normalized absolute value of explanation s_n and the normalized rationale r_n:

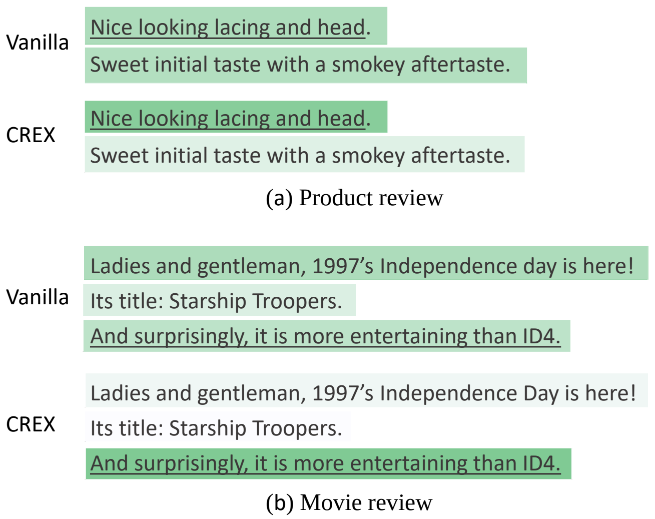
$$symKL(s'_n, r'_n) = \frac{1}{2}[KL(s'_n||r'_n) + KL(r'_n||s'_n)] \qquad (11)$$

where lower divergence means higher credibility [5]. We compare the credibility scores of CREX with three baseline methods on three DNN architectures over MR and PR dataset. The credibility results are presented in Tab. II. Comparing with Vanilla, the relative improvement of CREX is encouraging, with KL divergence drops ranging from 0.29 to 0.62 for DNNs in MR, from 0.23 to 0.58 for DNNs in PR. This ascertains the effectiveness of CREX in boosting the credibility of DNNs by pushing them to employ correct evidences to make decisions. The increased credibility of Rationale Augmentation comparing to Vanilla DNN also validates the value of expert knowledge, which succeeds to push models to focus more on evidences in the rationales to make decisions. In contrast, using back translation as Data Augmentation cannot always enhance the model credibility.

2) Quantitative Evaluation of Accuracy: DNNs trained via CREX have comparable predictive accuracy with the three baselines on MR and PR test set, as shown in Tab. III. Besides, the results of three comparing methods, including Vanilla training, Rationale augmentation, and CREX, are not

[4] https://pypi.org/project/googletrans

[5] https://code.google.com/archive/p/word2vec/


Vanilla

Nice looking lacing and head.

Sweet initial taste with a smokey aftertaste.

CREX

Nice looking lacing and head.

Sweet initial taste with a smokey aftertaste.

(a) Product review

Vanilla

Ladies and gentleman, 1997's Independence day is here!

Its title: Starship Troopers.

And surprisingly, it is more entertaining than ID4.

CREX

Ladies and gentleman, 1997's Independence Day is here!

Its title: Starship Troopers.

And surprisingly, it is more entertaining than ID4.

(b) Movie review

Fig. 3: Sentence-level explanation heatmap comparison between CREX and Vanilla DNN. Ground truth is annotated with underline. (a) Beer appearance review, positive label. (b) Movie review, negative sentiment label. Here ID4 denotes the movie Independence Day.

	Kaggle			Polarity		
Models	CNN	LSTM	Atten	CNN	LSTM	Atten
Vanilla DNN	74.3	73.6	74.7	60.7	62.6	64.8
Data Augment	75.7	70.3	75.0	62.5	58.1	65.4
Rationale Augment	76.5	73.9	**75.8**	63.1	63.2	65.3
CREX	**78.4**	**75.7**	75.2	**63.2**	**63.8**	**65.7**

TABLE IV: Generalization accuracy (in percent) of DNNs trained using MR dataset on two alternative datasets: Kaggle and Polarity.

Models	CNN	LSTM	Atten
Vanilla DNN	92.1	91.5	91.0
Data Augment	92.4	92.1	90.1
Rationale Augment	92.5	91.9	90.9
CREX	**92.7**	**92.3**	**91.2**

TABLE V: Generalization accuracy (in percent) of DNNs trained using PR dataset on an adversarial dataset.

substantially different. It means that the increased credibility does not sacrifice model performance on test set.

3) Qualitative Evaluation of Credibility: We provide case studies to qualitatively show the effectiveness of the increased credibility, as shown in Fig. 3. We show the sentence-level explanation scores, where deeper color means higher contribution to the prediction. For both cases, these two predictions are made by self-attention model, trained via Vanilla method and CREX method respectively.

For the first product review (PR) case shown in Fig. 3 (a), both DNNs give positive prediction for this testing instance, with 99.9% and 99.7% confidence respectively. We can observe that the Vanilla DNN pays nearly equal attention to the second sentence as the first one, even though the second sentence talks about the beer palate (*"sweet", "taste", "aftertaste"*) and has nothing to do with beer appearance. It indicates that the DNN classifier may have overfitted to bias in training set. In contrast, CREX could push the DNN to rely on correct evidences relevant to beer appearance, i.e., *"good looking"*, to make decisions. This explanation is consistent with our human cognition, and thus CREX is more likely to earn trust from end-users.

Similarly for the movie review case in Fig. 3 (b), although both self-attention models give correct predictions, they use distinct evidences to make decisions. Vanilla DNN pays nearly equal attention to the first and third sentence, where only the third sentence contains more generalizable features. One possible reason to explain this phenomenon is that the DNN may have memorized movie-unique terms to make decisions, which is supposed to perform poorly in movie reviews beyond training and test data. In contrast, CREX could focus mostly on the third sentence with task-relevant adjective i.e., *"entertaining"*, to make positive sentiment prediction. This finding demonstrates that CREX is able to disentangle useful knowledge from dataset specific biases. In next section, we will demonstrate the benefit of increased credibility of CREX on unseen testing data which are not drawn from test set.

C. Generalization Accuracy beyond Test Set

Currently, the generalization performance of DNNs is usually calculated using the prediction accuracy on the held-out test set. This is problematic due to the independent and identically distributed (i.i.d.) training-test split of data, especially in the presence of strong priors [22]. The DNN model can succeed by simply recognize patterns that only happen to be predictive on instances over the test set [44]. As evidenced by the example in Sec. V-B3, the DNN may rely on the aroma and palate as evidences to support appearance prediction, which is supposed to perform poorly in beer reviews outside of the training and test data. Consequently, test set fails to adequately measure how well DNN systems perform on new and previously unseen inputs. To assess the true generalization ability of DNN models as well as to demonstrate the benefit of increased credibility of CREX, we also evaluate the model performance using data beyond the test set.

1) Generalization for DNNs Trained on MR: For DNNs trained on MR, we use two alternative datasets:

- **Kaggle movie reviews dataset** [6] (**Kaggle**) It is a binary sentiment classification benchmark, with movie reviews from IMDB, consisting of 50,000 reviews.
- **Sentence polarity dataset** [7] (**Polarity**) [45]. Another binary sentiment classification dataset with data from IMDB, consisting of 10,662 reviews.

Note that none of the data from these two datasets is utilized to train DNN models or tune hyperparameters. They only serve

[6] https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset

[7] http://www.cs.cornell.edu/people/pabo/movie-review-data/

Models	Credibility	Kaggle	Polarity
CREX_conf	2.27	76.7	63.0
CREX_unc	2.37	77.6	62.2
CREX	2.24	78.4	63.2

TABLE VI: Ablation analysis of CNN trained on MR dataset. The first column is credibility score on MR test set, the last two columns denote generalization accuracy on two alternative datasets.

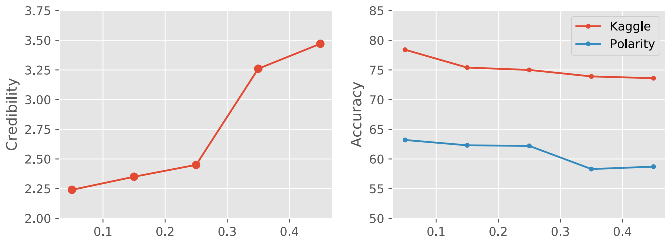
the testing purpose. The generalization accuracy statistics are shown in Tab. IV. There are several key observations. Firstly, comparing with the accuracy in Tab. III, there is a significant *generalization gap* between predictive accuracy on MR test set and Kaggle (or Polarity), for all three architectures. Almost most of the accuracy scores are above 90% on the corresponding test set. In contrast, all accuracy scores are below 80% for Kaggle and below 70% for Polarity dataset. Secondly, CREX could reduce this *generalization gap* comparing to baseline methods. In Tab. IV, CREX DNNs achieve substantial accuracy enhancements comparing to Vanilla DNNs, with relative accuracy improvement of 4.1%, 2.1%, 0.5% for three networks on Kaggle, and 2.5%, 1.2%, 0.9% for three networks on Polarity. These enhancements have validated the benefit of the increased credibility of our trained DNNs. Thirdly, an interesting observation is that there exists a positive correlation between the degree of credibility and the generalization accuracy on data not existing in test set. Rationale Augmentation has consistent accuracy improvement comparing with Vanilla, while Data Augment via back translation does not, as shown in Tab. IV. This conforms very well with the credibility performance in Tab. II.

2) Generalization for DNNs Trained on PR: To test generalization performance of DNNs trained on PR, we create an adversarial dataset by removing sentences relevant to beer aroma and palate. This is achieved via detecting sentences containing word *"taste", "smell", "aroma", "flavor", "drinking"* from the original PR test set. Note that we only differentiate beer appearance, thus description words about beer aroma and palate are considered as training set specific bias. The corresponding accuracy is shown in Tab. V, where CREX consistently outperforms baseline methods. Particularly, CREX DNNs have promoted the accuracy ranging from 0.2% to 0.8% comparing to Vanilla DNNs. It demonstrates that our trained DNNs rely more on correct evidences relevant to beer appearance rather than aroma and palate to make decisions, thus could achieve better generalization accuracy.

D. Ablation Study and Hyperparameters Analysis

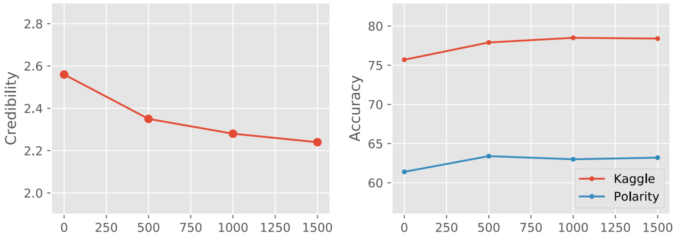
In this section, we utilize CNN trained on MR dataset to conduct ablation and hyperparameter analysis to study the impacts and contributions of different components of CREX.

1) Ablation Study: We compare CREX with its ablations to identify the contributions of different components. The ablations include (I). CREX_conf, using only confident explanation loss in Eq. (3), and (II). CREX_unc, using only uncertain explanation loss in Eq. (6). The comparison results between CREX and its ablations are listed in Tab. VI. We can observe that CREX outperforms the two ablations in terms of credibility as well as generalization accuracy on Kaggle and Polarity dataset. It indicates that these two components are complementary to each other in general, thus both are crucial in promoting model performance.



(a) Credibility (b) Accuracy

Fig. 4: CNN performance under different values of parameter λ_1. (a) credibility performance on MR test set. (b) generalization accuracy (in percent) on two alternative datasets.



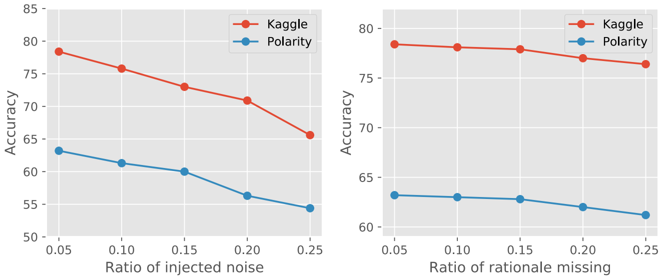
(a) Credibility (b) Accuracy

Fig. 5: CNN performance under different numbers of rationale. (a) credibility performance on MR test set. (b) generalization accuracy (in percent) on two alternative datasets.

2) Hyperparameter Analysis: We evaluate the effect of different degrees of rationale loss regularization towards models' performance, by altering the value of the hyperparameter λ_1. As shown in Tab. II, the optimal λ_1 for CNN trained on MR dataset is 5e-2. We are interested in how the model performance changes as we keep increasing the value of λ_1. The credibility and generalization accuracy are shown in Fig. 4. As the value of λ_1 increases, the CNN credibility begin to drop, i.e., KL divergence increases, and the model generalization accuracy on Kaggle and Polarity also decreases. Particularly, we observe a dramatic change of credibility and accuracy when λ_1 is larger than 0.25. This indicates that the models have overfitted to rationales, which also could sacrifice generalization performance.

E. Rationale Quantity and Quality Analysis

When incorporating human knowledge with DNN models, the quantity and quality of knowledge could have significant influences. In this section, we employ CNN trained on MR dataset to analyze how the performances of neural networks would be affected by different conditions of rationale.



(a) Mistakes

(b) Missing

Fig. 6: Rationale quality analysis using CNN generalization accuracy (in percent). (a) containing different ratios of mistakes. (b) missing different ratios of rationales.

1) Rationale Number Analysis: We study the effect of expert knowledge by altering the number of rationales N_r in the training set, and examine the credibility and accuracy change of the trained CNN. For those instances without rationales, we impose sparse regularization as in Eq. (8). The results are illustrated in Fig. 5. There are two interesting observations. Firstly, even when rationale number $N_r = 0$, our CNN could achieve improved performance comparing to the Vanilla CNN. The divergence has dropped from 2.86 to 2.58 comparing to Tab. II, and Kaggle and Polarity accuracy has increased 1.4% and 0.7% respectively comparing to Tab. IV, showing the effectiveness of sparse explanation loss in Eq. (8). Secondly, when the rationale number is 500, our CNN already has comparable accuracy comparing with $N_r = 1500$, indicating that a small ratio of rationales is sufficient for network performance promotion. Considering the annotation effort of expert rationales, this advantage of requiring small number of rationales is significant.

2) Rationale Quality Analysis: In this experiment, we analyze the effect of low quality rationales towards the DNN model performance. We consider two types of low quality: (I) containing mistakes (expert annotations could be sometimes wrong, and some irrelevant features are highlighted by the experts); (II) missing another set of important rationales. To simulate the first case, we inject different level of noise to the current rationales, and test model performance. Similarly, to test the second case, we delete different ratios of important features from current rationales to make the knowledge incomplete. We report CNN generalization accuracy over Kaggle and Polarity in Fig. 6. There are several key findings. Firstly, the model performances are highly sensitive to rationale noise (see Fig. 6 (a)), where a small ratio of mistakes, e.g., 10%, would significantly decrease generalization accuracy. Secondly, model performances are relatively robust to missing rationales (see Fig. 6 (b)). The reason for this phenomenon is that the remaining rationale still contains important features. By capturing sparse connections between input text and output, model could make reasonable predictions. Thirdly, considering that rationale missing is more common than containing crucial mistakes in real world rationale annotation, thus CREX is relatively robust to low-quality knowledge.

Models	Training time	Test time per input
Vanilla CNN	2.5 min	8e-3 seconds
CREX CNN	18.1 min	8e-3 seconds

TABLE VII: Running time comparison of Vanilla and CREX CNN. For training time, we report average value for three runs. Test time is the average over test set.

3) Running Efficiency Analysis: Due to the calculation of local explanation and regularization using rationales, the training speed of CREX is slightly slower than the Vanilla DNN training. As shown in Tab. VII, on average it takes 24 minutes to train CNN on the Movie Review dataset if using all the 1500 rationales in the training process (with our unoptimized code and using PyTorch GPU version). Even though CREX requires less training epoches to converge comparing to Vanilla, each epoch takes longer time than Vanilla training. To promote the training scalability of CREX, i.e., when the CREX is trained on a dataset which has much more training data comparing to MR and PR, We could reduce the ratio of rationales to speed up running of each epoch and make the total training time bearable. On the other hand, during the test stage, DNNs trained by CREX would need the same time (on average 8e-3 seconds) as Vanilla to yield prediction for an input, meaning that increased credibility of CREX would not sacrifice inference speed.

VI. Conclusion and Future Work

There has been an increasing interest recently in developing more trustworthy DNNs. In pursuit of this objective, we propose CREX, aiming to train credible DNNs which employ correct evidences to make decisions. We employ a specific kind of domain knowledge, called rationales, to guide the learning algorithms towards providing credible explanations, by pushing the explanation vectors to conform with rationales. CREX is DNN architecture agnostic, end-to-end trainable, and simple to implement. Experimental results show that our resulting DNN models have a higher probability to look at correct evidences rather than training dataset specific bias to make predictions. Although DNNs trained using CREX do not always improve prediction accuracy on held-out test set, they generalize much better on data which are beyond test set and which are representatives of underlying real-world tasks, highlighting the advantages of the increased credibility. High credibility and robustness of DNN are essential to earn trust of end-users towards a network model's predictions, and we believe the enhanced credibility and generalization will pave the way for their wider adoptions in real world.

On the other hand, it is not guaranteed that the incorporation of human knowledge with DNN models would always promote neural network performance, unless the knowledge have sufficiently high quality. Currently, we have explored the enhancement of DNNs via relatively high quality rationales. The low-quality knowledge issue is a challenging topic and would be explored in our future research.

References

[1] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing (DSP)*, 2018.

[2] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM (CACM)*, 2019.

[3] M. Du, N. Liu, Q. Song, and X. Hu, "Towards explanation of dnn-based prediction with guided feature inversion," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2018.

[4] M. Du, N. Liu, F. Yang, S. Ji, and X. Hu, "On attribution of recurrent neural network predictions via additive decomposition," in *The World Wide Web Conference (WWW)*, 2019.

[5] J. Wang, J. Oh, H. Wang, and J. Wiens, "Learning credible models," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2018.

[6] P. K. Mudrakarta, A. Taly, M. Sundararajan, and K. Dhamdhere, "Did the model understand the question?" *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[7] B. Rychalska, D. Basaj, P. Biecek, and A. Wroblewska, "Does it care what you asked? understanding importance of verbs in deep learning qa system," *EMNLP workshop*, 2018.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2016.

[9] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith, "Annotation artifacts in natural language inference data," *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

[10] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Thirtieth Conference on Neural Information Processing Systems (NIPS)*, 2016.

[11] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "Swag: A large-scale adversarial dataset for grounded commonsense inference," *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[12] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, "Harnessing deep neural networks with logic rules," *54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[13] T. Mihaylov and A. Frank, "Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge," *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[14] Y. Zhang, I. Marshall, and B. C. Wallace, "Rationale-augmented convolutional neural networks for text classification," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[15] O. Zaidan, J. Eisner, and C. Piatko, "Using annotator rationales to improve machine learning for text categorization," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2007.

[16] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[17] J. Donahue and K. Grauman, "Annotator rationales for visual recognition," *International Conference on Computer Vision (ICCV)*, 2011.

[18] T. McDonnell, M. Lease, M. Kutlu, and T. Elsayed, "Why is that relevant? collecting annotator rationales for relevance judgments," in *Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016.

[19] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[20] F. Yang, M. Du, and X. Hu, "Evaluating explanation without ground truth in interpretable machine learning," *arXiv preprint arXiv:1907.06831*, 2019.

[21] N. Liu, M. Du, and X. Hu, "Representation interpretation with spatial encoding and multimodal analytics," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM)*, 2019.

[22] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, "Don't just assume; look and answer: Overcoming priors for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach, "Women also snowboard: Overcoming bias in captioning models," in *15th European Conference on Computer Vision (ECCV)*, 2018.

[24] S. Ramakrishnan, A. Agrawal, and S. Lee, "Overcoming language priors in visual question answering with adversarial regularization," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[25] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[26] M. Barrett, J. Bingel, N. Hollenstein, M. Rei, and A. Søgaard, "Sequence classification with human attention," in *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL)*, 2018, pp. 302–312.

[27] Y. Bao, S. Chang, M. Yu, and R. Barzilay, "Deriving machine attention from human rationales," *2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[28] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," *arXiv preprint arXiv:1612.08220*, 2016.

[29] A. Kádár, G. Chrupała, and A. Alishahi, "Representation of linguistic form and function in recurrent neural networks," *Computational Linguistics*, pp. 761–780, 2017.

[30] B. Peters, V. Niculae, and A. F. Martins, "Interpretable structure induction via sparse attention," in *EMNLP Workshop*, 2018.

[31] C. Malaviya, P. Ferreira, and A. F. Martins, "Sparse and constrained attention for neural machine translation," *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[32] Z. C. Lipton, "The mythos of model interpretability," *arXiv preprint arXiv:1606.03490*, 2016.

[33] Y. Kim, "Convolutional neural networks for sentence classification," *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

[35] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *International Conference on Learning Representations (ICLR)*, 2017.

[36] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[37] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics (ACL)*, 2004.

[38] J. McAuley, J. Leskovec, and D. Jurafsky, "Learning attitudes and attributes from multi-aspect reviews," in *International Conference on Data Mining (ICDM)*. IEEE, 2012.

[39] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[40] A. Poncelas, D. Shterionov, A. Way, G. M. d. B. Wenniger, and P. Passban, "Investigating backtranslation in neural machine translation," *arXiv preprint arXiv:1804.06189*, 2018.

[41] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Conference on Neural Information Processing Systems (NIPS)*, 2013.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, 2014.

[44] P. Minervini and S. Riedel, "Adversarially regularising neural nli models to integrate logical background knowledge," *The SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2018.

[45] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Annual meeting on association for computational linguistics (ACL)*, 2005.