Knowledge-Based Systems 275 (2023) 110685

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

PERCY: A post-hoc explanation-based score for logic rule dissemination consistency assessment in sentiment classification

Shashank Gupta [a], Mohamed Reda Bouadjenek [a,*], Antonio Robles-Kelly [b]

[a] School of Information Technology, Deakin University, Waurn Ponds Campus, Geelong, VIC 3216, Australia
[b] Defense Science and Technology Group, Australia

ARTICLE INFO

Article history:
Received 14 March 2023
Received in revised form 23 May 2023
Accepted 27 May 2023
Available online 3 June 2023

Keywords:
Logic rules dissemination
Sentiment classification
Explainable AI

ABSTRACT

Disseminating and incorporating logic rules into deep neural networks has been extensively explored for sentiment classification in recent years. In particular, most methods and algorithms proposed for this purpose rely on a specific component that aims to capture and model logic rules, followed by a sequence model to process the input sequence. While the authors of these methods claim that they effectively capture syntactic structures that affect sentiment classification, they only show improvement in accuracy to support their claims without further analysis. Focusing on various syntactic structures, particularly contrastive discourse relations such as the *A-but-B* structure, we introduce the PERCY score, a novel Post-hoc Explanation-based Rule ConsistencY Score to analyze and study the ability of several of these methods to identify these structures in a given sentence, and to make their classification decisions based on the appropriate conjunct. Specifically, we explore the use of model-agnostic post-hoc explanation frameworks to explain the predictions of any classifier in an interpretable and faithful manner. These model explainability frameworks provide feature attribution scores to estimate each word's impact on the final classification decision. Then, they are combined to check whether the model has based its decision on the right conjunct. Our experiments show that (a) accuracy – or any other performance metric – can be misleading in assessing the ability of logic rule dissemination methods to base their decisions on the right conjunct, (b) not all analyzed methods effectively capture syntactic structures, (c) often, the underlying sequence model is what captures the structure, and (d) for the best method, less than 25% of the test examples are classified based on the appropriate conjunct, indicating that a lot of research needs to be done on this topic. Finally, we experimentally demonstrate that the PERCY scores calculated are robust and stable w.r.t. the feature-attribution frameworks used.

1. Introduction

Deep Neural Networks (DNNs) provide extraordinary performance across a broad spectrum of Natural Language Processing (NLP) tasks such as Sentiment Classification [1], Machine Translation [2], Text Summarizing [3], etc. This is mainly due to their characteristic of hierarchical feature representation [4], which can be learned automatically through purely data-driven approaches (i.e., without any external supervision) using a gradient-based optimization algorithm with a task-specific objective.

However, these hierarchical representations of features, when learned through purely data-driven approaches, suffer from several drawbacks including: (i) their complexity, which often leads to the extraction of human-uninterpretable features and hinders their application in high-stakes domains where automated decision-making systems must have a human understanding of their internal process, requiring the user to trust their outputs [5], (ii) DNNs are treated as essentially *black-box* models, where no meaningful relationship in terms of *"how?"* and *"why?"* can be established between inputs and outputs; (iii) a huge amount of labeled training data is required to construct these models, which is both expensive and time-consuming [6]; and (iv) previous ablation studies on DNNs [7,8] have shown that purely data-driven training may also lead to the learning of spurious feature representations, which can provide unreasonable outputs and make them prone to malicious attacks based on adversarial examples [9,10].

To combat these drawbacks, several solutions aim to make these networks inherently interpretable by augmenting them with some task-specific or domain-specific expert prior knowledge. These solutions are collectively called Neural-Symbolic

* Corresponding author.
E-mail addresses: guptashas@deakin.edu.au (S. Gupta), reda.bouadjenek@deakin.edu.au (M.R. Bouadjenek), antonio.robleskelly@defence.gov.au (A. Robles-Kelly).

https://doi.org/10.1016/j.knosys.2023.110685

methods [11] as they aim to combine symbolic knowledge represented by logical rules with Deep Neural Networks. They have been extensively explored for various NLP tasks such as sentiment classification [12], question answering [13], machine translation [14], and information extraction [15], where the ultimate goal is to model and transfer various human interpretable logical rules to a neural network in order to improve its accuracy and causal interpretability. These methods usually rely on (1) a component aimed at capturing and modeling logic rules (e.g., the teacher network in the Iterative Knowledge Distillation method [12], the ELMo component in the Contextualized Word Embeddings approach [16], or the Semantic Composition Module in SentiBERT [17]), (2) followed by a Neural Network model to process the input sequence, (e.g., 1-D CNN [18], RNN, etc.).

While authors of these methods claim that they effectively capture syntactic structures in an input sentence that affect the outcome of a particular task (e.g., sentiment classification), they have only shown improvement in terms of accuracy to support their claim with no further analysis provided. However, achieving a high classification accuracy does not necessarily indicate that a method has effectively captured and encoded such logical syntactic structures. For example, let us consider the sentence *"the casting was not bad but the movie was horrible"* that has an *A-but-B* structure – a component *A* being followed by *but*, which is followed by a component *B*. In this example, the conjunction is interpreted as an argument for the second conjunct, with the first functioning concessively [19,20]. While a sentiment classifier can correctly identify that this sentence has a negative sentiment, it may fail to infer its decision based only on the *B* part of the sentence (i.e., *"the movie was horrible"*), but instead, it may base it's decision on individual negative words also present in Part *A* (i.e., *"bad"*). Thus, we argue in this paper that the high accuracy of a classifier does not necessarily indicate that it has effectively captured textual structures of input sentences.

Focusing on various syntactic structures, in particular contrastive discourse relations such as the *A-but-B*, *A-yet-B*, *A-though-B*, or *A-while-B* structures, we introduce the PERCY score, a novel Post-hoc Explanation-based Rule ConsistencY Score, which is used to evaluate both the task-specific performance and logic-rule dissemination performance of a Neural-Symbolic system. In particular, *PERCY* is used to analyze and study the ability of various knowledge dissemination methods[1] to: (i) effectively identifying a syntactic structure in an input sentence, (ii) encode and model these structures in sequences, and (iii) make their classification decisions based on the appropriate conjunct. Specifically, we explore the use of post-hoc explanation frameworks that are agnostic to the underlying model such as LIME [21], SHAP [22], and Integrated-gradients (IG) [23], which explain the predictions of any classifier by providing feature-attribution scores. Furthermore, we use these scores to evaluate the impact of each conjunct in a sentence with a syntactic structure on the decision made by a classifier. Going back to the example mentioned above (i.e., "the casting was not bad but the movie was horrible"), PERCY helps users understand if a classification model has made its decision based on the B conjunct or individual words of this sentence.

The contributions of this paper are summarized as follows:

- We present a novel Post-hoc Explanation-based Rule ConsistencY Score called *"PERCY"*, that we use to quantitatively assess the ability of various knowledge dissemination methods to encode logic rules and text syntactic structures.
- We conduct an exhaustive experimental evaluation on two datasets, Sentiment140 [24] and SST2 [25], on which we compare various methods for logic rule dissemination with diverse classification methods – a total of 40 sentiment classifiers are evaluated. Briefly, we demonstrate that:
    1. Accuracy or any other performance-based metric can be misleading in assessing methods for capturing logic rules.
    2. Not all methods are effectively capturing syntactic structures as they claim to do.
    3. Their sequence model is often what captures the syntactic structure.
    4. The best method makes its decision based on the appropriate conjunct in less than 25% of the test examples.
- We experimentally demonstrate that the PERCY scores calculated are robust and stable w.r.t. the Feature-attribution based Local Post-hoc Explanation frameworks used in this study.

The rest of the paper is organized as follows: Section 2 covers related work and puts our work in perspective. Section 3 describes the post-hoc explanation frameworks that we use in this paper, followed by a detailed presentation of the *PERCY* score. Section 4 gives a brief overview of the analyzed sentiment classifiers and a thorough presentation of the investigated logic rule dissemination methods. Sections 5 and 6 respectively provide a description of our experimental setup and an analysis of the results we obtained. Finally, in Section 7 we conclude and suggest future research directions.

2. Related work

There is a substantial body of research related to disseminating and incorporating logic rules in deep neural networks. Below, we first describe the main text syntactic structures we consider in this paper, and then, we review *Neural-Symbolic models*, which are DNN models augmented with symbolic domain knowledge related to a specific task. Next, we discuss *Neural-Symbolic models* for NLP and then we provide a description of how these augmented models are often evaluated — mainly by focusing on their performance.

2.1. Logic rules for sentence-level sentiment classification

Text sentiment classification has a long and rich history of research due to its various practical applications, e.g., e-commerce, social media analysis, etc. In particular, sentence-level sentiment classification is the task that consists of determining the sentiment of a sentence by classifying it often as Positive, Negative, or Neutral. One important challenge in this regard is to model discourse relations between phrases and clauses in a sentence and to identify which part of a sentence will determine its overall sentiment [26–28].

In linguistics, a discourse relation is a description of how two segments of a sentence are logically connected to each other through a discourse marker or connector. Prasad et al. [29] have classified discourse markers as follows:

1. **Contingency relations:** which include markers like *because, therefore, if, so, since* to convey cause–effect relations between segments.
2. **Contrast relations:** which include markers like *but, although, though, however, whereas, while* to convey contrastive sense relations between segments.

[1] In this paper, we use the term "knowledge dissemination methods" to refer specifically to techniques for incorporating explicit logic rules into machine learning models, with the aim of improving their performance on specific tasks.

Table 1
List of syntactic structures that we consider with the PERCY score. *Rule conjunct* denotes the dominant clause that determines the overall sentiment of the sentence.

Logic rule	Keyword	Rule conjunct	Example
A − **but** − B	*but*	B [26]	Yes there is an emergency called covid-19 **but** *victory is worth celebration*
A − **yet** − B	*yet*	B [26]	Even though we can't travel **yet** *we can enjoy each other and what we have*
A − **though** − B	*though*	A [26]	*You are having an amazing time* **though** we are having this awful pandemic
A − **while** − B	*while*	A [31]	*Stupid people are not social distancing* **while** there's a global pandemic

3. **Temporal relations:** which include markers like *before, after, when, since, while* to convey the order of occurrence of segments.
4. **Expansion relations:** which include markers like *and, in addition* to convey that a latter segment elaborates on the former.

Previous work has shown that Contrastive Discourse Relations (CDRs) are hard to capture by general DNN models like CNNs or RNNs for sentence-level binary sentiment classification through purely data-driven learning [16,30]. Thus, Prasad et al. [29] define such relations as sentences containing *A-keyword-B* syntactic structure where two clauses *A* and *B* are connected through a discourse marker (the *keyword*) and have contrastive polarities of sentiment. Mukherjee and Bhattacharyya [26] argue that these relations need to be learned by the model while determining the overall sentence sentiment.

Table 1 summarizes all logic rules that we consider in this paper with the PERCY score, where we show the structure, the rule conjunct, and an example sentence. We selected these structures because they are examples of contrastive discourse relations that are commonly used in natural language to express complex ideas and opinions. These structures introduce a sense of contrast or opposition between two clauses or ideas, which can have a significant impact on the overall sentiment expressed in a sentence. Our study focuses on these structures in particular because they have been shown to be particularly challenging for sentiment classification models to accurately identify and classify. As a result, we believe that exploring how different knowledge dissemination methods perform on these specific structures will help shed light on the strengths and weaknesses of these methods when dealing with complex and nuanced linguistic phenomena.

2.2. Neural symbolic models

While traditional DNN models provide state-of-the-art performance on various pattern recognition tasks, they lack reasoning capabilities and act as black-box function approximators. On the other hand, symbolic models such as Decision Trees [32] or Inductive Logic Reasoning-based approaches [33,34] are inherently interpretable as they manipulate discrete categorical variables, but they show lower performance capabilities compared to DNNs [35]. Hence, a hybrid model called *Neural Symbolic Model* that combines both approaches has been proposed with the aim of equipping the hierarchical feature representation learning of Neural Networks with some real-world rules to make their prediction, coherent, consistent, and easily interpretable [11,36,37]. However, even before the advent of modern Neural Networks, constructing such knowledge and rule-augmented models has been extensively explored. For example, Towell and Shavlik [38] developed Knowledge-Based Artificial Neural Networks (KBANN) to combine symbolic domain knowledge abstracted as propositional logic rules with neural networks via a three-step pipelined framework. Later on, França et al. [36] constructed a neural model called CLIP++, which learns first-order relations from structured data through Inductive Logic Programming. More recently, Evans and Grefenstette [39] proposed a differentiable Inductive Learning framework to train a neural network via back-propagation on unstructured data. Instead of integrating Logic Rules as hard constraints, Manhaeve et al. [40] and Xu et al. [41] convert them into probabilistic soft-logic and integrate them with Deep Learning frameworks as soft-constraints. More recently, Lin et al. [42] proposed to fuse domain knowledge as topology contexts and logical rules of Knowledge Graphs into Language Models as soft constraints via Knowledge Distillation [43]. In this paper, our contribution is more analytical and focused on effectively testing such methods for their ability to encode and disseminate knowledge.

2.3. Neural symbolic models for natural language processing

A lot of research has been done on developing Neural-Symbolic Models for various Natural Language Processing tasks, where performance metrics have been mainly used to report their efficacy. For example, Hu et al. [12] fused domain knowledge abstracted as First Order Logic Rules with Deep Neural Networks via EM style algorithm called *Iterative Knowledge Distillation*. An updated version of this algorithm called *Mutual Distillation* [30] introduced some learnable parameters with Logic Rules to incorporate the fuzzy nature of domain knowledge. Zhang et al. [44] introduced a *critic learning* framework to augment a CNN-based model with various syntactical logical rules via Knowledge Distillation [43]. Cambria et al. [45] introduces a model called *SenticNet 7* which builds a hierarchical knowledge graph from the input sentence using kernel methods and auto-regressive language models and uses linguistic patterns to determine the sentiment polarity. Also, Chen et al. [46] introduced a *feedback masking* method where redundant parts of the input sequence (i.e., *A* tokens of a sentence with an *A-but-B* structure) are masked out before being fed to a Recurrent Neural Network fusing it with logical knowledge. More recently, Wang and Pan [15,47] developed a *discrepancy loss* to fuse First Order Logic Rules with Neural Networks for Information Extraction tasks like Opinion Target Extraction and Relation Extraction. On the other hand, instead of imposing constraints on loss function, Li and Srikumar [48] developed *constrained neural layers*, where logical constraints govern the forward computation operations in each neuron. Instead of changing either the loss function or the architecture, Wang and Poon [49] and Gu et al. [50] performed manipulation on input training data to induce logical domain knowledge. Finally, while not proposed to construct Neural-Symbolic models, Krishna et al. [16] showed that *contextualized word embeddings* constructed from large pretrained models like ELMo [51] can inherently capture the logical relationships like *A-but-B* for sentiment classification, but again, they have use accuracy to prove their claims.

2.4. Evaluation of neural-symbolic models

All of the work mentioned above has reported results using performance metrics such as accuracy [12,30,44,46,48–50] or F1-Score [15,47,48] to support the claim that their methods have effectively captured logical domain knowledge. In this paper, we claim that while these performance metrics reflect the ability of a model to correctly identify the true class, they may fail to assess whether the classifier has actually captured logic rules and other syntactic structures. Hence, in this paper, we follow a different approach to evaluate such logic rule-augmented models by exploring the use of model-agnostic post-hoc explanation frameworks such as LIME [21], SHAP [22], and Integrated-gradients (IG) [23], which gives a local explanation for each output in terms of input features. This approach helps to provide a causal explanation as quantifiable feature-attribution scores for an output given an input sentence with a certain rule syntactic structure, which we use to formulate our metric. To the best of our knowledge, this is the first work to provide a *quantitative* evaluation of such models using post-hoc explanation methods. While there have been various proposed Neural-Symbolic models for a wide range of tasks, our paper specifically concentrates on sentence-level sentiment classification. It is important to note that the analysis of other models for different tasks is beyond the scope of our study.

3. Methodology

As mentioned earlier, our main goal in this paper is to assess a sentiment classifier for its ability to correctly classify a test example with a logical syntactic structure on the basis of the appropriate conjunct. There are many methods proposed for generating explanations and incorporating interpretability and transparency into machine learning models [52]. Some methods provide explanations prior to their training, which typically involves designing models that are inherently more interpretable, such as rule-based systems [53,54] or decision trees [55–57], or incorporating specific features or constraints into the learning process to ensure that the resulting model is more transparent and easier to understand [58]. On the other hand, model-specific explainable AI involve developing techniques for analyzing and interpreting the internal workings of these models to better understand how they arrive at their predictions. Some common approaches include model simplification [59,60], visual attribution methods [61–63], feature relevance estimation [64,65], and other attention-based methods [66]. In contrast to these explainable methods, our approach requires generating explanations post-modeling. This is because we aim to answer the question of why a given model produces a certain output for a given input, regardless of the specific model architecture used. Therefore, we use *local post-hoc explanation frameworks, whose* output is a causal mapping from the input datapoint to the model prediction. We distinguish diverse frameworks that are different depending upon the nature of the explanation provided, for instance: feature attribution scores [21–23], natural language explanations like Counterfactuals [67,68], or *if–then–else* type logical rules such as Scoped-rules in [69].

In this paper, we rely on Feature Attribution (also called Feature Importance) to calculate our *PERCY* score. Feature Attributions are obtained using model-agnostic Local Post-hoc explanation frameworks, which operate at the level of an individual input/prediction pair, producing an explanation for why a model predicted an output for a particular input. Fig. 1 provides a visual representation of the method we propose in this paper. It offers a general overview of the steps involved in our approach and how they relate to each other. By referring to this figure, readers can get a better understanding of the PERCY score calculation process.

3.1. Feature attribution based local post-hoc explanation frameworks

Local Post-hoc Explanation Frameworks have been used to explain outputs of various machine learning models ranging from a simple logistic regression to complex deep neural networks like Inception network [70–73]. The output of these frameworks is usually a list of weights, where each reflects the contribution of a particular feature to the prediction of a test datapoint. This provides local interpretability, and it also allows to determine which feature changes will have most impact on the prediction. Such approaches can be built on different types of features, such as manual features obtained from feature engineering, lexical features including words/tokens and n-gram, or latent features learned by NNs. In the next sections, we provide details about three local post-hoc explanation frameworks used in this paper and how feature-attribution scores are calculated using these frameworks.

3.1.1. LIME

Local Interpretable Model-agnostic Explanations (LIME) is a framework developed by Ribeiro et al. [21] that can explain the output prediction of any classifier or a regressor in a faithful way, by approximating it locally with a simpler and interpretable model on an input instance. LIME learns surrogate models using an operation called input perturbation and can be used to achieve either local [21] or global explanations [74].

Let's consider a model f and a sentence $\mathbf{s} \in \mathbb{S}$ represented with an n-dimensional token sequence vector $\mathbf{s} = \{t_1 t_2 \cdots t_n\}$. LIME proceeds by assigning to each token t_i a weight w_i that reveals its importance in influencing the output prediction of f. LIME assigns these weights as "Sparse Linear Models", which are surrogate linear models learned in the vicinity of the input $\mathbf{s}$. These surrogate models are computed by solving the following optimization:

$$\arg\min_{g \in G} \sum_{\mathbf{z},\mathbf{z}' \in \mathbb{Z}} exp(-cos(\mathbf{s}, \mathbf{z})^2/\sigma^2)(f(\mathbf{z}) - g(\mathbf{z}'))^2 + \Omega(g) \tag{1}$$

where $\mathbb{Z}$ is a set of all perturbations computed for $\mathbf{s}$, cos is the cosine distance, G is the set of interpretable linear surrogate models, and $\Omega(g)$ denotes the measure of complexity of g (for explanations as linear models, it is the number of weights in every model). The optimal solution of Eq. (1) denotes the preciseness of surrogate model g in approximating model f around the locality of $\mathbf{s}$ defined by $\Omega(g)$.

3.1.2. SHAP

Shapley Additive Explanations (SHAP) is a framework developed by Lundberg and Lee [22] to provide model-agnostic local explanations based on feature-attribution. SHAP is based on the game theoretically optimal Shapley values. Specifically, given a model f and an input sentence $\mathbf{s}$, to produce an interpretable model, SHAP defines an output model $g(\mathbf{s}')$ with simplified input $\mathbf{s}'$ as a linear addition of all input tokens as follows:

$$f(\mathbf{s}) = g(\mathbf{s}') = \phi_0 + \sum_{t_i \in \mathbf{s}'} \phi_i t_i' \tag{2}$$

where $\mathbf{s}$ is the original sentence, $\mathbf{s}'$ is a simplified input sentence with a mapping function $\mathbf{s} = h_s(\mathbf{s}')$ between $\mathbf{s}$ and $\mathbf{s}'$, and $\phi_0 = f(h_s(0))$ is the model output without all of the simplified inputs. A detailed description of SHAP and a possible solution for Eq. (2) can be found in the literature [22].

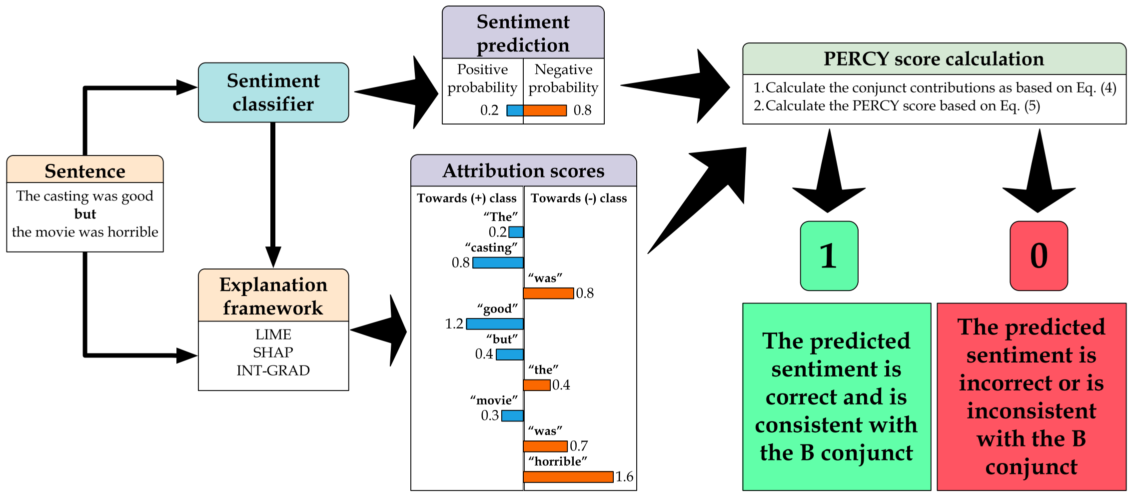


Fig. 1. Workflow diagram to illustrate the various steps involved in the calculation of the PERCY score for a classifier that predicts the sentiment of a sentence with an *A-but-B* syntactic structure.

3.1.3. IG

Integrated Gradients (IG) is a simple, yet powerful axiomatic attribution method developed by [23], which provides feature importance scores using product of their gradients and values. While the previous two frameworks are based on local perturbations, IG is based on a Gradient perturbation method to calculate Feature Attribution scores. Specifically, let's suppose we aim to explain the prediction of a model f for an input sentence $\mathbf{s}$. The integrated gradient for the token t_i of the input sentence is defined as follows:

$$IG(s) = (t_i - t_i')\int_{\alpha=0}^{1} \frac{\partial f(t_i' + \alpha(t_i - t_i'))}{\partial t_i} d\alpha \tag{3}$$

where the gradient of f for the token t_i is denoted by $\frac{\partial f(\mathbf{s})}{t_i}$, and t_i' is the ith token in a selected sentence baseline $\mathbf{s}'$. For most models, it is recommended to choose a baseline such that the prediction at the baseline is near zero ($f(\mathbf{s}') \approx 0$). A more detailed explanation of IG can be found in [23].

3.2. PERCY: Post-hoc explanation-based rule consistency score

Let's consider a model f and a sentence $\mathbf{s} \in \mathbb{S}$ that is represented with an n-dimensional token sequence vector $\mathbf{s} = \{t_1 t_2 \cdots t_n\}$. All feature attribution frameworks described above assign a weight w_i for each term $t_i \in \mathbf{s}$ to estimate its contribution to the prediction of f. Often, a positive weight $w_i > 0$ indicates that t_i contributes and supports the positive class, whereas a negative weight $w_i < 0$ indicates a contribution of t_i towards a negative prediction. Hence, given a sentence $\mathbf{s}$ that contains an *A-keyword-B* syntactic structure, we first define the sub-sequences $\mathbf{a} = \{t_1 \cdots t_{k-1}\}$ and $\mathbf{b} = \{t_{k+1} \cdots t_n\}$ as respectively the left and right sub-sequences w.r.t. the word *"but"* indexed by k.

3.2.1. Calculating the conjunct contribution

Next, we estimate the contribution of each sub-sequence $\mathbf{a}$ and $\mathbf{b}$ to the prediction of f as an expectation over $\mathbf{a}$ and $\mathbf{b}$ using a weighted average of all tokens in each part as follows:

$$\begin{aligned} \mathbb{E}[\mathbf{a}] &= \overbrace{\sum_{i=1}^{k-1} w_i \times p(y=1|\mathbf{s})}^{\mathbb{E}[\mathbf{a}]^+} + \overbrace{\sum_{i=1}^{k-1} |w_i| \times p(y=0|\mathbf{s})}^{\mathbb{E}[\mathbf{a}]^-} \\ \mathbb{E}[\mathbf{b}] &= \overbrace{\sum_{i=k+1}^{n} w_i \times p(y=1|\mathbf{s})}^{\mathbb{E}[\mathbf{b}]^+} + \overbrace{\sum_{i=k+1}^{n} |w_i| \times p(y=0|\mathbf{s})}^{\mathbb{E}[\mathbf{b}]^-} \end{aligned} \tag{4}$$

where k is the index of the *"keyword"*, $p(y = 0|\mathbf{s})$ and $p(y = 1|\mathbf{s})$ are the probabilities to predict respectively the class 0 and 1 given a sentence $\mathbf{s}$, w_i is the feature attribution weight given to a term t_i, and $\mathbb{E}(\cdot)^+$ ($\mathbb{E}(\cdot)^-$) is the expected value over terms contributing to the positive class (resp. negative class). Following these estimations, the *PERCY* score of a sentence $\mathbf{s}$ is calculated depending on the rule as detailed in Table 2.

In Table 2, "$f(\mathbf{s}) = y$" aims to check that the prediction is correct — accuracy, "$\mathbb{E}[\mathbf{a}] < \mathbb{E}[\mathbf{b}]$" ensures that the sub-sequence $\mathbf{b}$ has contributed more to the prediction of f, and the p-value aims to make sure that the difference between $\mathbb{E}[\mathbf{a}]$ and $\mathbb{E}[\mathbf{b}]$ is statistically significant. When a sentence contains multiple syntactic structures, such as a combination of "A-but-B" and "not only", the PERCY score can be calculated for each syntactic structure separately.

Finally, the *PERCY* score of a collection of sentences $\mathbb{S}$ is calculated by averaging as follows:

$$PERCY(\mathbb{S}) = \frac{1}{|\mathbb{S}|}\sum_{i=1} PERCY(\mathbf{s}_i) \tag{7}$$

We note that in the experimental evaluation we present in Section 6, we mainly report the *PERCY* at the collection level (Eq. (7)) to compare and contrast the different sentiment classification methods we describe in the next section. In Section 6.3, we provide justification behind each step involved in the calculation of PERCY score in 2 using qualitative analysis of *A-but-B* type sentences.

4. Sentiment classification methods

In this section, we provide a succinct description of the sentiment classification methods used in our experimental analysis.

4.1. Logic rules dissemination methods

In this section, we describe the main methods we analyze to disseminate logic rule knowledge into the Neural Network models described in Section 4.2.

4.1.1. Iterative knowledge distillation

The Iterative rule knowledge distillation method proposed by Hu et al. [12] aims to transfer the domain knowledge encoded in first order logic rules into a neural network defined by a conditional probability $p_\theta(y|x)$ where θ is a parameter to

Table 2
PERCY score.

Logic rule	Rule conjunct	Equation	
$A-\textbf{but}-B$ $A-\textbf{yet}-B$	B [26]	$PERCY(\mathbf{s})=\begin{cases}1, & \text{if } (f(\mathbf{s})=y) \text{ AND } [(\mathbb{E}[\mathbf{a}]<\mathbb{E}[\mathbf{b}]) \text{ AND } (p\text{-value} \leq 0.05)] \\ 0, & \text{otherwise}\end{cases}$	(5)
$A-\textbf{though}-B$ $A-\textbf{while}-B$	A [26]	$PERCY(\mathbf{s})=\begin{cases}1, & \text{if } (f(\mathbf{s})=y) \text{ AND } [(\mathbb{E}[\mathbf{a}]>\mathbb{E}[\mathbf{b}]) \text{ AND } (p\text{-value} \leq 0.05)] \\ 0, & \text{otherwise}\end{cases}$	(6)

learn. To integrate the information encoded in the rules, Hu et al. [12] have proposed to train the network via knowledge distillation as proposed in Hinton et al. [43] where hard targets are provided through labeled training data and soft targets are constructed through rule constrained projection of posterior $p_\theta(y|x)$ as proposed in Posterior Regularization [75].

Specifically, during training, a posterior $q(y|x)$ is constructed by projecting $p_\theta(y|x)$ into a subspace constrained by the rules to encode the desirable properties as follows:

$$\begin{aligned} \min_{q,\xi\geq 0} \quad & KL(q(y|x) \parallel p_\theta(y|x)) + C\sum_{x\in X}\xi_x \\ s.t. \quad & (1-\mathbb{E}_{y\leftarrow q(y|x)}[r_\theta(x,y)]) \leq \xi_x \end{aligned}$$

where $q(y|x)$ denotes the distribution of (x, y) when x is drawn uniformly from the train set X and y is drawn according to $q(y|x)$, $r_\theta(x, y) \in [0, 1]$ is a variable that indicates how well labeling x with y satisfies the rule, $\xi_x \leq 0$ is the slack variable for respective logic constraint, and C is the regularization parameter. The closed form solution for $q(y|x)$ is used as soft targets to imitate the outputs of a rule-regularized projection of $p_\theta(y|x)$, which explicitly includes rule knowledge as regularization terms.

Next, the rule knowledge is transferred to the posterior $p_\theta(y|x)$ through knowledge distillation optimization objective:

$$(1-\pi)\times \mathcal{L}(p_\theta, P_{true}) + \pi \times \mathcal{L}(p_\theta, q)$$

where P_{true} denotes the distribution implied by the ground truth, $\mathcal{L}(\bullet, \bullet)$ denotes the cross-entropy function, and π is a hyper-parameter that needs to be tuned to calibrate the relative importance of the two objectives. Following the terminologies used by authors in Hinton et al. [43], p_θ is called a "student" network and q is called a "teacher" network, which is intuitively analogous to human education where a teacher is aware of systematic general rules and instructs students. Overall, the Iterative rule knowledge distillation method is agnostic to the network architecture, and thus is applicable to general types of neural models such as those depicted in Fig. 2.

4.1.2. Word embeddings

Traditional word embedding methods like Word2vec [76] and Glove [77] provide a unique and fixed vector for each word in the vocabulary. However, language is complex and context can completely change the meaning of a word in a sentence. Hence, contextual word embeddings methods have emerged as a way to capture the different nuances of the meaning of words given the surrounding text. Krishna et al. [16] have advocated that word embeddings when fine-tuned with downstream sentiment analysis task might capture logic rules and thus disseminate that latent information, for example in the 1D CNN sequence models of the neural network in Fig. 2(a). In this paper, we experiment with the following word embedding methods:

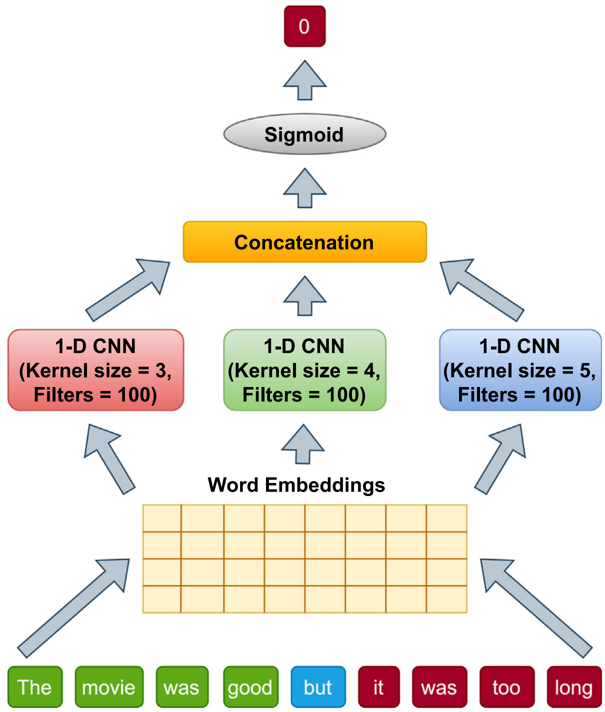
1. **Word2vec:** which is one of the most popular methods to efficiently create word embeddings developed by Mikolov et al. [76]. Briefly, word2vec embeddings are computed from a two-layer neural network. Word2vec maps each token to a vector space, typically of several hundred dimensions, where word vectors are positioned in the vector space such that words that share common contexts (semantically similar) are located close to each other in the space.
2. **Glove:** is an unsupervised learning algorithm for obtaining vector representations for words developed by Pennington et al. [77]. Training is performed on the non-zero entries of a global word-to-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. A matrix factorization algorithm is applied to efficiently extract the embeddings.
3. **ELMo:** stands for Embeddings from Language Models is a pre-trained model developed by Peters et al. [51]. Instead of using a fixed embedding for each word, ELMo looks at the entire sentence before assigning each word in it an embedding. It uses a bi-directional LSTM trained on a specific task to be able to create those embeddings. Krishna et al. [16] proposed to use ELMo in their method.
4. **BERT:** stands for Bidirectional Encoder Representations from transformers. This is also a pre-trained model developed by Devlin et al. [78]. Briefly, the BERT is a model based on Encoder Transformer blocks [79], which processes each element of the input sequence by incorporating and estimating the influence of other elements in the sequence to create embeddings.

4.1.3. Modeling semantic composition using self-attention

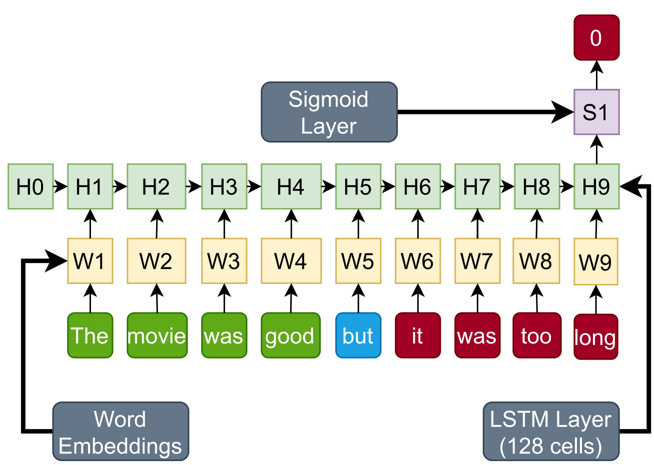
Large pre-trained models like BERT [78] and GPT [80] have achieved state-of-the-art performance on various NLP tasks. Usually, these models follow a pre-training step on a large language corpus and then fine-tuning on the downstream NLP task coupled with a smaller Neural Network model. Recent work [81,82] has shown that inducing domain or task specific knowledge during their pre-training phase improves performance on the downstream task. Following this line of research, other work [50,83–85] has sought to develop methods and frameworks to induce domain-specific or task knowledge into pre-training of large language models.

In particular, *SentiBERT* developed by Yin et al. [17] focuses on sentence-level sentiment analysis task and develops a self-attention based mechanism on top of BERT to capture rule-syntactic structures like *A-but-B* in input sentences. The authors argue that combining contextual information generated from a language model like BERT [78] with constituency parse-trees like that generated by Socher et al. [25] can better capture composition semantic relations in an input sentence. In this paper, we use the pre-trained weights of SentiBERT provided by the authors[2] instead of training the Language Model from scratch.

[2] The implementation and weights can be found here: https://github.com/WadeYin9712/SentiBERT.



(a) 1-D CNN model



(b) Single LSTM Layer model

Fig. 2. Backbone Neural Network models used for the construction of the sentiment classifiers.

4.2. Backbone models

We use in this paper the following two backbone neural network models:

CNN model: which is depicted in Fig. 2(a) and used in [18,86] for sentence-level sentiment classification. It takes as input a sequence of tokens, which are first processed by an embedding layer and converted into dense vectors of fixed size. Next, three 1D CNN sequence models (kernel size of 3, 4, and 5) process the embeddings in parallel in order to extract diverse features from the input sequence. These 1D CNN sequence models may learn various internal properties of the sequence that are useful for sentiment classification. Finally, the outputs of the three 1D CNNs are concatenated before being fed into a feed-forward binary classification layer with a sigmoid activation to extract the sentiment of the input sentence – 0 for a negative sentiment and 1 for a positive sentiment.

LSTM model: which is illustrated in Fig. 2(b) and is based on recurrent neural networks [87]. Similar to the CNN model, it also takes as input a sequence of tokens, which are converted into dense vectors by an embedding layer. Next, the token embeddings are passed to a many-to-one sequence model layer consisting of 128 LSTM units, which learn hidden features in the sequence relevant to the understanding of the sentiment of the sentence. Finally, the output corresponding to the last token of the sequence model layer is fed to a dense layer consisting of a single sigmoid activation unit which classifies the entire sequence as — 0 for a negative sentiment and 1 for a positive sentiment.

4.3. Sentiment classification methods

To conduct a thorough evaluation, we consider all possible configuration options that we discussed above as follows: {CNN, LSTM} × {Distillation, No Distillation} × {Fine-tuning, No Fine-tuning } × {word2vec, glove, elmo, bert, sentibert}, which gives a total of 40 sentiment classifiers that are summarized in Table 3. For example, the classifier *CDFB* in Table 3 indicates that the base neural network used is the CNN model, word embeddings are created using BERT, which is fine-tuned on the downstream sentiment classification task, and the training was done using Iterative Knowledge Distillation method.

5. Experimental setup

In this section, we describe the experimental setup we use in our evaluations, including a description of the datasets, the metrics used, and the details of our implementation.

5.1. Datasets

We train the sentiment classification models discussed in the previous section on two popular sentence-level sentiment classification datasets.

Stanford Sentiment Treebank (SST2): This dataset proposed in [25] is a binary sentiment classification dataset and consists of 9,613 single sentences extracted from movie reviews, where sentences are labeled as either positive or negative each accounting for about 51.6% and 48.3%. A total of 1,078 sentences contain a syntactic structure, which accounts for about 11.2% of the dataset. We report our results only on test examples that contain a syntactic structure to demonstrate the ability of a classifier to capture the pattern.

Sentiment140: Since SST2 dataset contains low amount of sentences containing a syntactic structure, we complement it with Sentiment140 dataset, which contains a significant proportion of such sentences. Constructed from twitter corpus, Go et al. [24] released this dataset to perform sentence-level sentiment analysis on public domain tweets. It consists of 1.6M tweets scrapped from twitter using their API[3] divided into 3 categories — positive, negative and neutral. For our evaluation, we rejected the neutral tweets and randomly selected 50,000 tweets containing equiproportion distribution of positive and negative sentiment tweets. Out of these 50,000 tweets, approximately 51% tweets contain a syntactic structure. Again, we report our results only on test examples that contain these syntactic structures.

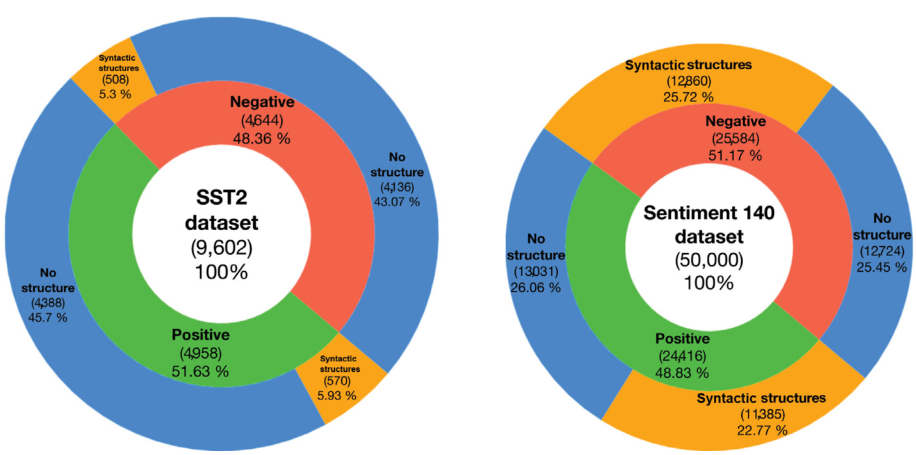
Fig. 3 shows the complete distribution of these two datasets.

[3] More information about the Twitter API can be found at http://apiwiki.twitter.com/.

Table 3
Summary of the sentiment classification methods used in our experimental evaluation.

Model no.	Classifier	Base model	Distillation	Fine-tuning WE	WE	LRD
1	CW	CNN (C)	x	x	word2vec (W)	x
2	CG	CNN (C)	x	x	glove (G)	x
3	CE	CNN (C)	x	x	elmo (E)	x
4	CB	CNN (C)	x	x	bert (B)	x
5	CsB	CNN (C)	x	x	sentibert (sB)	✓
6	CFW	CNN (C)	x	✓(F)	word2vec (W)	x
7	CFG	CNN (C)	x	✓(F)	glove (G)	x
8	CFE	CNN (C)	x	✓(F)	elmo (E)	✓
9	CFB	CNN (C)	x	✓(F)	bert (B)	✓
10	CFsB	CNN (C)	x	✓(F)	sentibert (sB)	✓
11	CDW	CNN (C)	✓(D)	x	word2vec (W)	✓
12	CDG	CNN (C)	✓(D)	x	glove (G)	✓
13	CDE	CNN (C)	✓(D)	x	elmo (E)	✓
14	CDB	CNN (C)	✓(D)	x	bert (B)	✓
15	CDsB	CNN (C)	✓(D)	x	sentibert (sB)	✓
16	CDFW	CNN (C)	✓(D)	✓(F)	word2vec (W)	✓
17	CDFG	CNN (C)	✓(D)	✓(F)	glove (G)	✓
18	CDFE	CNN (C)	✓(D)	✓(F)	elmo (E)	✓
19	CDFB	CNN (C)	✓(D)	✓(F)	bert (B)	✓
20	CDFsB	CNN (C)	✓(D)	✓(F)	sentibert (sB)	✓
21	LW	LSTM (L)	x	x	word2vec (W)	x
22	LG	LSTM (L)	x	x	glove (G)	x
23	LE	LSTM (L)	x	x	elmo (E)	x
24	LB	LSTM (L)	x	x	bert (B)	x
25	LsB	LSTM (L)	x	x	sentibert (sB)	✓
26	LFW	LSTM (L)	x	✓(F)	word2vec (W)	x
27	LFG	LSTM (L)	x	✓(F)	glove (G)	x
28	LFE	LSTM (L)	x	✓(F)	elmo (E)	✓
29	LFB	LSTM (L)	x	✓(F)	bert (B)	✓
30	LFsB	LSTM (L)	x	✓(F)	sentibert (sB)	✓
31	LDW	LSTM (L)	✓(D)	x	word2vec (W)	✓
32	LDG	LSTM (L)	✓(D)	x	glove (G)	✓
33	LDE	LSTM (L)	✓(D)	x	elmo (E)	✓
34	LDB	LSTM (L)	✓(D)	x	bert (B)	✓
35	LDsB	LSTM (L)	✓(D)	x	sentibert (sB)	✓
36	LDFW	LSTM (L)	✓(D)	✓(F)	word2vec (W)	✓
37	LDFG	LSTM (L)	✓(D)	✓(F)	glove (G)	✓
38	LDFE	LSTM (L)	✓(D)	✓(F)	elmo (E)	✓
39	LDFB	LSTM (L)	✓(D)	✓(F)	bert (B)	✓
40	LDFsB	LSTM (L)	✓(D)	✓(F)	sentibert (sB)	✓

WE=Word Embeddings used by the model.
LRD=Whether model is proposed for Logic Rule Dissemination or not.
F=Word embeddings were fined tuned on the downstream task.
D=Trained via Iterative Knowledge Distillation [12].



(a) SST2 dataset. (b) Sentiment140 dataset.

Fig. 3. Distributions of datasets used in our experimental evaluation. The inner-most layer gives the total number of instances in each dataset, 2nd layer indicates the total number of instances with positive and negative sentiment labels, and the outer-most layer gives the number of instances containing syntactic structures for each sentiment label.

5.2. Metrics

The performance is measured using the following conventional classification evaluation metrics: (i) Accuracy, (ii) Precision, (iii) Recall, and (iv) F1-score. In addition, we report the PERCY scores as described in Section 3 using the three explanation frameworks: LIME [69], SHAP [22] and IG [23], which we refer to as *L-PERCY*, *S-PERCY*, and *I-PERCY* scores respectively. We aim to assess the robustness of PERCY w.r.t. the explanation framework used and to compare the classification performance metrics with the PERCY score to ultimately assess how correlated these metrics are.

5.3. Implementation details

We divide the Sentiment140 dataset into train, val, and test splits using 60%, 20%, and 20% proportion of sentences respectively. Each split contains similar distributions for various subsets – no structure-positive, no structure-negative, syntactic structure-positive, syntactic structure-negative – as present in the complete dataset in Fig. 3(b). For SST2 dataset, since the sample size of test instances is very small (1,078 sentences), all classifier are trained, tuned, and tested using stratified nested k-fold cross-validation and evaluated primarily according to accuracy. This is done to increase the size of test instances and to reduce high variance. For Sentiment140, we use standard training procedure with Early-Stopping.

We optimize all models using mini-batch gradient descent with batch size = 50 using an Adam optimizer [88] with learning rate $\eta = 3e - 5$. We also use early stopping and a dropout $= 0.5$ regularization techniques to get the best weights for the models. For nested k-fold cross-validation in SST2, we set the value of $k = 5$ and the value of inner fold $l = 3$. The code of our implementation can be found at: https://github.com/shashgpt/PERCY.

6. Experimental evaluation

In this section, we discuss and analyze the results obtained for the sentiment classification models described in Section 4.3. Briefly, we first discuss their performance using conventional classification performance metrics as detailed in Section 5.2 and PERCY scores. Next, we analyze the correlation of PERCY with respect to the classification performance metrics. Finally, we discuss and evaluate the consistency and robustness of the PERCY score across the different explainability frameworks we use — LIME, SHAP, and IG.

6.1. Performance evaluation

Tables 4 and 5 show the performance of all classifiers described in Table 3 on SST2 and Sentiment140 respectively. We show the performance using all sentiment classification metrics (Classification Accuracy, Weighted Precision, Weighted Recall and Weighted F-1 scores) and the PERCY score with three explainability frameworks (LIME, SHAP and IG), which are called L-PERCY, S-PERCY and I-PERCY respectively. Briefly, we make the following key observations:

1. For all classifiers, PERCY score values are less than 25%, which indicates that less than 25% of the test examples are correctly classified based on the correct conjunct. This suggests that the performance claimed by the logic rule dissemination methods analyzed in Sections 4.1.1, 4.1.2, and 4.1.3 is far from being achieved and that there is a lot of research that needs to be done on this topic.
2. There is major discrepancy between the classification performance metrics and the PERCY score values — the values of the classification performance metrics are much higher. The reason for this difference is discussed with anecdotal examples further in Section 6.3. In short, we observe that often, the two conjuncts contain the same number of sentiment-sensitive words. Hence, we argue that the classifiers are using those individual tokens to base their sentiment decision.
3. The BERT word-embeddings dissemination method (Section 4.1.2) provides the best classification performance values, whereas ELMo word-embeddings provides the best PERCY score values. This indicates that contextualized word-embeddings are better at capturing and disseminating logic rules.
4. We note that the classifiers that use the Iterative Knowledge Distillation method show almost no improvement on all metrics, e.g., in Table 4, *CW* and *CDW* classifiers provide similar values for all metrics. This simply suggests that [12] is not efficient and that it is the underlying sequence model that is capturing to some extent the syntactic structure.
5. Finally, we also observe that the SentiBERT method (Section 4.1.3) is not efficient at capturing syntactic structures as the performance using the PERCY score is always very low.

6.2. Correlation between PERCY scores and performance metrics

To analyze the correlation between performance metric values and PERCY score values, we show in Figs. 4(a) and 4(b) scatterplots with best fit linear regression of the rankings obtained using PERCY scores vs. Accuracy scores of all classifiers described above. The correlation is quantified using the Kendall's correlation coefficient (τ) [89].

Briefly, at first glance we observe that in all plots there is a little to no correlation between the rankings obtained using PERCY scores and Accuracy scores — the highest Kendall τ is 0.1487. This indicates clearly that higher accuracy cannot be used to claim that a classifier is performing well on rule dissemination since there is no-correlation between them. Probable reason for no correlation can be attributed to the classifiers using individual sentiment sensitive tokens in A conjunct. We note that we do not show the rank correlation plots for precision, recall and F1-scores vs. PERCY scores as we found them to be identical to those we show in Figs. 4(a) and 4(b).

6.3. Qualitative analysis

In order to provide insight into the factors behind specific PERCY scores, explore the lack of correlation between PERCY scores and Accuracy scores, and strengthen our analysis, we present in Table 6 examples of sentences exhibiting an *A-but-B* syntactic structure divided into three categories:

1. Examples where the sentiment prediction of the classifier is correct and the decision was based on the "B" conjunct to provide intuitive sense on why the sentiment of sentences containing *A-but-B* syntactic structures should be based on the "B" conjunct. These examples are shown in Table 6a.
2. Examples where the sentiment prediction of the classifier is correct but the decision was based on the "A" conjunct according to the PERCY score to show that accuracy can be misleading to assess rule-dissemination performance. These examples are shown in Table 6b.

Table 4
Performance of the classifiers described in Table 3 on the SST2 dataset.

Classifier	Sentiment classification specific performance metrics				PERCY scores		
	Accuracy	Precision	Recall	F1-score	L-PERCY	S-PERCY	I-PERCY
CW	0.7537	0.7543	0.7537	0.7539	0.1336	0.0765	0.0629
CG	0.7326	0.7326	0.7326	0.7326	0.1383	0.0888	0.0548
CE	0.8193	0.8193	0.8193	0.8192	0.2229	0.2028	0.1927
CB	0.8258	0.8258	0.8258	0.8258	0.2038	0.1837	0.1737
CsB	0.8811	0.8811	0.8811	0.8811	0.0888	0.0686	0.0587
CFW	0.763	0.7629	0.763	0.7629	0.1493	0.0803	0.055
CFG	0.7495	0.7496	0.7495	0.7496	0.1483	0.09	0.0622
CFE	0.9752	0.9752	0.9752	0.9752	0.2262	0.2061	0.1962
CFB	0.9833	0.9833	0.9833	0.9833	0.2266	0.2064	0.1964
CFsB	0.8771	0.8774	0.8771	0.8772	0.0698	0.0495	0.0397
CDW	0.743	0.7448	0.743	0.7433	0.1316	0.1009	0.0882
CDG	0.712	0.712	0.712	0.712	0.1252	0.1164	0.0781
CDE	0.8221	0.8231	0.8221	0.8223	**0.2468**	**0.2267**	**0.2166**
CDB	0.8242	0.8245	0.8242	0.8243	0.2059	0.1858	0.1758
CDsB	0.878	0.878	0.878	0.878	0.0839	0.0637	0.0538
CDFW	0.7523	0.7524	0.7523	0.7523	0.1418	0.1181	0.0877
CDFG	0.7365	0.7366	0.7365	0.7357	0.1273	0.1215	0.079
CDFE	0.9726	0.9727	0.9726	0.9726	0.1458	0.1356	0.1157
CDFB	**0.9833**	**0.9833**	**0.9833**	**0.9833**	0.2338	0.2235	0.2036
CDFsB	0.8823	0.8823	0.8823	0.8823	0.0762	0.0661	0.0461
LW	0.7203	0.725	0.7203	0.7203	0.101	0.0713	0.0829
LG	0.7189	0.7216	0.7189	0.7191	0.088	0.0647	0.0555
LE	0.7136	0.7154	0.7136	0.7138	0.165	0.1348	0.1448
LB	0.8103	0.8104	0.8103	0.8103	0.1652	0.135	0.145
LsB	0.8553	0.8557	0.8553	0.855	0.0791	0.0489	0.0591
LFW	0.7491	0.75	0.7491	0.7493	0.0914	0.0603	0.0789
LFG	0.7426	0.7426	0.7426	0.7426	0.0893	0.0571	0.0513
LFE	0.8998	0.9008	0.8998	0.8996	0.1288	0.0986	0.1086
LFB	0.9722	0.9722	0.9722	0.9722	0.2149	0.1848	0.1948
LFsB	0.8759	0.8765	0.8759	0.876	0.0704	0.0403	0.0501
LDW	0.7124	0.7172	0.7124	0.7124	0.113	0.0994	0.1365
LDG	0.7141	0.7151	0.7141	0.7143	0.0981	0.0807	0.1009
LDE	0.654	0.6682	0.654	0.6391	0.1484	0.1183	0.1282
LDB	0.8031	0.803	0.8031	0.8031	0.1599	0.1297	0.1397
LDsB	0.8593	0.8593	0.8593	0.8592	0.0808	0.0507	0.0607
LDFW	0.7539	0.7552	0.7539	0.7541	0.1202	0.1095	0.1269
LDFG	0.7289	0.7289	0.7289	0.7289	0.0986	0.0796	0.0974
LDFE	0.7795	0.7823	0.7795	0.7778	0.1531	0.1329	0.1329
LDFB	0.9791	0.9792	0.9791	0.9791	0.2324	0.2122	0.2122
LDFsB	0.8794	0.8796	0.8794	0.8795	0.0746	0.0546	0.0546

C=CNN, **L**=LSTM.
D=Distillation.
F=Fine-tuning.
W=word2vec, **G**=Glove, **E**=elmo, **B**=bert, **sB**=sentibert.

3. Examples where the sentiment prediction is correct and conjunct contribution of "A" is greater than "B" but the "B" conjunct contains a single token having a higher score than all "A" conjunct tokens, i.e., $\mathbb{E}[\mathbf{a}] > \mathbb{E}[\mathbf{b}]$ but $max[\mathbf{a}] < max[\mathbf{b}]$. These examples show that using an additive operation like "expectation" is better suited from a robustness point of view than using any other operation like "max" to calculate the conjunct contribution in PERCY scores. These examples are shown in Table 6c.

For each category, we provide two examples in which one has a positive ground-truth sentiment and the other has a negative ground-truth sentiment. Briefly, we observe that:

1. In Table 6a, *A* and *B* conjuncts of both examples contain a comparable amount of sentiment-sensitive tokens to each other and the feature attribution scores assigned to *B* conjunct tokens are higher than the scores assigned to tokens in the conjunct *A*. Observing the nature of the sentiment switch from *A* to *B*, we can see that it makes sense to base the decision on the *B* conjunct to determine the sentence-level sentiment. This observation is consistent with the general Linguistics study of contrastive discourse relations like *A-but-B* done in [19,20]. Thus, there are neural-symbolic methods (Section 4.1) proposed to disseminate this *A-but-B* rule knowledge into a general DNN model (Section 4.2) to force the model to make sentiment-prediction as per the *B* conjunct.
2. In Table 6b, we observe that *A* conjunct contain more sentiment-sensitive tokens than *B* conjunct and have a similar sense of sentiment i.e. they do not have any contrastive sentiment polarities. Thus, the classifier uses the individual tokens in *A* conjunct to base its decision, which is consistent with the ground-truth sentiment. This observation proves that *A-but-B* rule-dissemination performance and sentiment classification performance cannot be interlinked as the former checks whether the methods of rule-dissemination actually enable the classifier to learn and recognize *A-but-B* syntactic structures and forces the model to base its decision on the *B* conjunct.
3. Finally, in Table 6c, we observe that in both examples, *A* conjuncts contain more sentiment-sensitive tokens than *B* conjuncts and the conjuncts do not have contrastive sentiment polarities. Moreover, in both examples, the tokens that get the highest feature-attribution score in *B* conjunct are non-sentiment sensitive tokens (e.g., "value" and "too"). We note that the sentiment of the sentence is consistent with the sentiment of A conjunct and it makes

Table 5
Performance of the classifiers described in Table 3 on the Sentiment140 dataset.

Classifier	Sentiment classification specific performance metrics				PERCY scores		
	Accuracy	Precision	Recall	F1-score	L-PERCY	S-PERCY	I-PERCY
CW	0.6697	0.6704	0.6697	0.668	0.0876	0.0664	0.084
CG	0.6626	0.6629	0.6626	0.6611	0.0803	0.0639	0.0868
CE	0.7552	0.7554	0.7552	0.7549	0.1091	0.099	0.0891
CB	0.7603	0.7602	0.7603	0.7602	0.115	0.105	0.0949
CsB	0.7229	0.7338	0.7229	0.7177	0.064	0.0538	0.0438
CFW	0.7003	0.7003	0.7003	0.6998	0.1027	0.0819	0.0954
CFG	0.6976	0.6975	0.6976	0.6971	0.1014	0.0729	0.0876
CFE	0.7636	0.7635	0.7636	0.7635	0.0828	0.0727	0.0577
CFB	**0.7798**	0.7822	**0.7798**	**0.7798**	0.0901	0.0801	0.065
CFsB	0.749	0.7584	0.749	0.7451	0.0663	0.0561	0.0412
CDW	0.6727	0.676	0.6727	0.6689	0.1058	0.0861	0.0943
CDG	0.6725	0.6745	0.6725	0.6696	0.0903	0.0767	0.096
CDE	0.7607	0.7609	0.7607	0.7603	**0.1473**	0.1371	0.1072
CDB	0.7647	0.7646	0.7647	0.7646	0.1083	0.0983	0.0683
CDsB	0.7267	0.7291	0.7267	0.725	0.0602	0.0501	0.0201
CDFW	0.7052	0.7054	0.7052	0.7043	0.1123	0.0943	0.1071
CDFG	0.6938	0.6945	0.6938	0.6926	0.1113	0.0821	0.0956
CDFE	0.7615	0.7615	0.7615	0.7613	0.0912	0.0811	0.0512
CDFB	0.7726	0.7734	0.7726	0.7727	0.1182	0.1081	0.0782
CDFsB	0.7611	0.7623	0.7611	0.7612	0.0691	0.059	0.0289
LW	0.7056	0.7073	0.7056	0.7038	0.0637	0.0505	0.0522
LG	0.7272	0.7276	0.7272	0.7264	0.0635	0.0497	0.0557
LE	0.7251	0.7259	0.7251	0.724	0.1044	0.0943	0.0843
LB	0.7399	0.741	0.7399	0.74	0.1025	0.0924	0.0824
LsB	0.7173	0.7206	0.7173	0.7149	0.0656	0.0556	0.0455
LFW	0.7165	0.7169	0.7165	0.7156	0.0702	0.0675	0.0507
LFG	0.7194	0.7203	0.7194	0.7183	0.0484	0.0335	0.0247
LFE	0.746	0.746	0.746	0.7458	0.0853	0.0752	0.0551
LFB	0.7783	**0.7831**	0.7783	0.778	0.093	0.083	0.0629
LFsB	0.7636	0.7639	0.7636	0.7637	0.0695	0.0593	0.0394
LDW	0.6991	0.7047	0.6991	0.695	0.105	0.141	0.0922
LDG	0.7219	0.7248	0.7219	0.7198	0.1162	**0.1632**	0.2173
LDE	0.7221	0.7254	0.7221	0.7198	0.1344	0.1442	**0.2343**
LDB	0.7416	0.7417	0.7416	0.7417	0.0922	0.1021	0.1922
LDsB	0.7066	0.7163	0.7066	0.701	0.0605	0.0704	0.1604
LDFW	0.7179	0.7198	0.7179	0.7162	0.0617	0.0501	0.0155
LDFG	0.7228	0.7267	0.7228	0.7202	0.0717	0.0792	0.0712
LDFE	0.7569	0.7576	0.7569	0.7562	0.0893	0.0991	0.0591
LDFB	0.7588	0.7594	0.7588	0.7582	0.1304	0.1402	0.1002
LDFsB	0.7586	0.7593	0.7586	0.7587	0.0692	0.0791	0.039

C=CNN, **L**=LSTM.
D=Distillation.
F=Fine-tuning.
W=word2vec, **G**=Glove, **E**=elmo, **B**=bert, **sB**=sentibert.

intuitive sense to use an additive operation like "expectation over weights" as shown in Section 3.2.1 to calculate the conjunct contribution, which determines the overall contribution of all the tokens in a conjunct.

6.4. Robustness of explanation frameworks for PERCY

Feature attribution based local post-hoc explanation E on a sentence $\mathbf{s} \in \mathbb{S}$ for the model f can be viewed on a higher level as a function of both $\mathbf{s}$ and f [90] as follows:

$$E_{\mathbf{s}} = g(\mathbf{s}, f) \tag{8}$$

This is true for all frameworks that we use in our analysis — LIME [21], SHAP [22] and IG [23]. As we can see in Eq. (8), the explanation E depends on both the sentence $\mathbf{s}$ and the model to be explained f.

Previous studies like [90,91] have shown that these explanation frameworks suffer from non-robustness issues, i.e., they provide substantially different explanations on a locally perturbed sample $\mathbf{z}$ of the input sentence $\mathbf{s}$ (the sample sentence $\mathbf{z}$ is locally perturbed to $\mathbf{s}$ if $\|\mathbf{s} - \mathbf{z}\| \approx 0$). Alvarez-Melis and S. Jaakkola [90] argues that the explanations can only be considered meaningful or valid if they fulfill the criteria of being *robust* to the local perturbations of the input sentence $\mathbf{s}$. Intuitively, similar inputs should provide similar explanations. Hence, in the following, we analyze the robustness of explanation frameworks for PERCY using two methods.

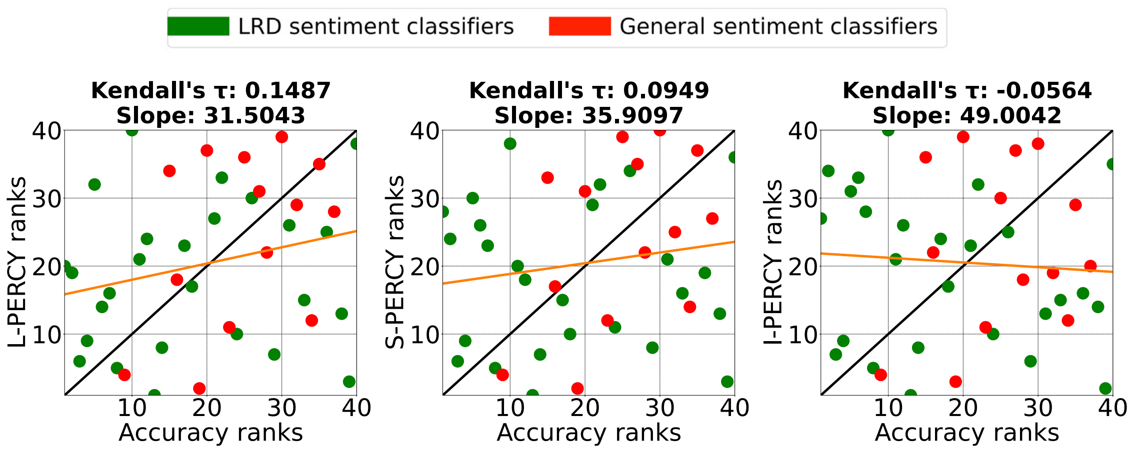
6.4.1. Local lipschitz estimates

Mathematically, a post-hoc explanation E in Eq. (8) is robust if $\|E_{\mathbf{s}} - E_{\mathbf{z}}\| \approx 0$ for $\|\mathbf{s} - \mathbf{z}\| \approx 0$ given $\|P(y|\mathbf{s}; w) - P(y|\mathbf{z}; w)\| \approx 0$. To quantify this robustness, Alvarez-Melis and S. Jaakkola [90] propose to calculate the *Local Lipschitz Estimate* of E. Inspired by Lipschitz continuity in calculus, which measures relative changes in function output with respect to function input in the entire domain, Alvarez-Melis and S. Jaakkola [90] propose to calculate the point-wise, neighborhood-based *Local Lipschitz Estimate* of E on an input sentence $\mathbf{s}$ of interest. Specifically, given a set of N sentences $\mathbb{S} = \{\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(N)}\}$, they propose to define for every sentence $\mathbf{s}^{(i)} \in \mathbb{S}$ a set of all local perturbations to $Z^{(i)}$ as follows:

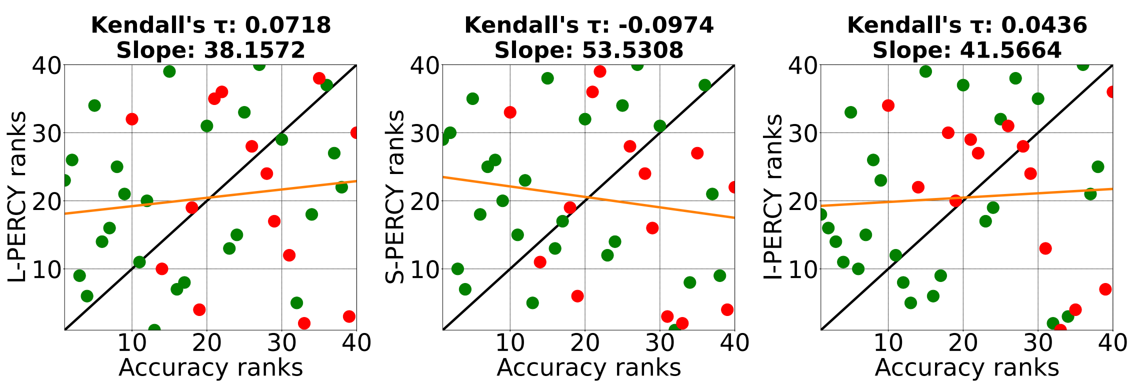
$$Z^{(i)}_{\epsilon} = \{\mathbf{z}^{(i,j)} \mid \|\mathbf{s}^{(i)} - \mathbf{z}^{(i,j)}\| \leq \epsilon\} \tag{9}$$

where they set $\epsilon = 0.1$ to obtain local perturbations. Then, they propose to calculate the Lipschitz Estimate for each sentence $\mathbf{s}^{(i)} \in \mathbb{S}$ as follows:

$$L_{\mathbb{S}}(\mathbf{s}^{(i)}) = \underset{\mathbf{z}^{(i,j)} \in Z^{(i)}_{\epsilon}}{\arg\max} \frac{\|E_{\mathbf{s}} - E_{\mathbf{z}}\|_2}{\|\mathbf{s} - \mathbf{z}\|_2} \tag{10}$$



(a) Classification Accuracy vs PERCY scores rank correlation plots on SST2 dataset



(b) Classification Accuracy vs PERCY scores rank correlation plots on Sentiment140 dataset

Fig. 4. Rank correlation scatter plots between sentiment classification accuracy values vs. PERCY score values on SST2 and Sentiment140 datasets. Green points represent classifiers proposed for Logic Rule Dissemination (LRD) in Table 3 whereas red points represent classifiers constructed for general sentiment classification (non-LRD).

Intuitively, the fraction $\frac{\|E_{\mathbf{s}}-E_{\mathbf{z}}\|_2}{\|\mathbf{s}-\mathbf{z}\|_2}$ in Eq. (10) should be bounded by a constant value L where $L \approx 0$ for all $\mathbf{s} \in \mathbb{S}$. Hence, the higher the value of L, the lower the stability of explanations E, which in turn means that the explanatory framework that generated E is less robust. Although the Lipschitz Estimate is a unit-less quantity, Alvarez-Melis and S. Jaakkola [90] states that it has no "ideal" universally desirable value and its acceptable value will depend on the end use of the generated explanations. In our case, we use the generated explanations to compute PERCY scores as shown in Section 3. Moreover, as it can be seen in Eq. (8), the generated explanations are dependent upon the dataset and model, which means lower values of Lipschitz scores on one set of {dataset, model} does not mean it will be lower on another set of {dataset, model}.

In Fig. 5, we show the obtained local Lipschitz scores for all the classifiers in Table 3 using the three explanation frameworks used in our analysis – LIME, SHAP, and IG – on our two datasets — Sentiment140 and SST2. Overall, we make the following key observations:

1. We observe that SHAP has the lowest overall scores as the median values are lowest and thus, has the highest stability among all frameworks.
2. Moreover, the two other frameworks provide comparable scores as well which are low as compared to the ones reported in Alvarez-Melis and S. Jaakkola [90]. We note that these results are not contradictory but as stated by Alvarez-Melis and S. Jaakkola [90] and can be seen in Eq. (8), Local Lipschitz Estimates depend on the dataset and model to be explained, which are different in our paper compared to [90]. Thus, in our case, we can reach the conclusion that the explanation frameworks *seems to provide robust-enough* explanations on datasets and classifiers used.

6.4.2. Correlation between PERCY scores

As mentioned earlier, the end use of the generated explanations from all frameworks is to calculate PERCY scores as detailed in Section 3. While Lipschitz scores in Fig. 5 are quite low, they are still unable to tell whether the generated explanations are *robust-enough* so as not to influence the final PERCY score calculation. To measure the impact of explanations instability on final PERCY scores calculations, we compute correlations between PERCY scores calculated from all three explanation frameworks – LIME, SHAP, and IG – as shown in Figs. 6 and 7.

In particular, we show in Fig. 6 scatterplots with best fit linear regression of the rankings obtained using PERCY scores with the different explanation frameworks of all classifiers described above. Also, the correlation is quantified using Kendall's correlation coefficient (τ) [89]. On top of these plots, we also calculate the Pearson's correlation between PERCY scores from different frameworks as distributions, i.e., PERCY scores calculated from each explanation framework – LIME, SHAP, and IG – are represented respectively as:

$$PERCY_{fr_{dist}} = \begin{cases} 1, & \text{if } PERCY_{fr}(x_i) = 1 \\ 0, & \text{if } PERCY_{fr}(x_i) = 0 \end{cases} \tag{11}$$
$$\forall x_i \in X_{test} = \{X_{test_{c1}} + X_{test_{c2}} + \cdots + X_{test_{cn}}\}$$

Table 6
Anecdotal examples containing *A-but-B* syntactic structures. In each conjunct, we highlight tokens based on their feature-attribution weights assigned from an explanation framework. Darker color indicates a higher token score while lighter color indicates a lower token score. We show the scores obtained for the CDE classifier in Table 4 since it is an LRD classifier and it has the highest PERCY score values on all three explanation frameworks.

(a) Examples where the predicted sentiment was correct and the decision was based on the *B* conjunct.

Sentences	Ground truth sentiment
lots of effort and intelligence are on display **but** in execution it is all awkward , static , and lifeless rumblings .	Negative
often messy and frustrating , **but** very pleasing at its best moments , it 's very much like life itself.	Positive

(b) Examples where the predicted sentiment was correct but the decision was based on the *A* conjunct.

Sentences	Ground truth sentiment
" analyze that " is one of those crass , contrived sequels that not only fails on its own , **but** makes you second guess your affection for the original .	Negative
a gorgeously strange movie , heaven is deeply concerned with morality , **but** it refuses to spell things out for viewers .	Positive

(c) Examples to support using the "Expectation" operation instead of the "Max" for calculating conjunct contribution. The sentiments of these examples were correctly predicted.

Sentences	Ground truth sentiment
a fine , rousing , g rated family film , aimed mainly at little kids **but** with plenty of entertainment value to keep grown ups from squirming in their seats .	Positive
tries to add some spice to its dull sentiments **but** the taste is all too familiar .	Negative

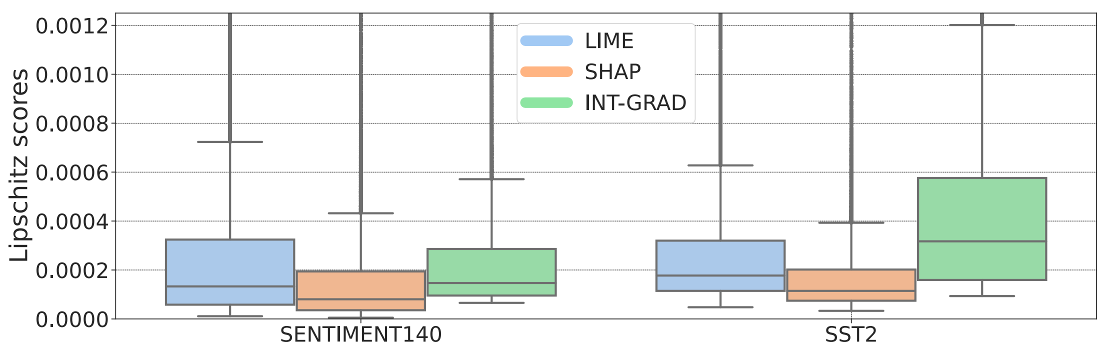


Fig. 5. Lipschitz scores of the explanation frameworks we use in our analysis. Each box plot denotes the Lipschitz values for all the classifiers in Table 3 of all the test data points on a particular dataset from a particular explanation framework.

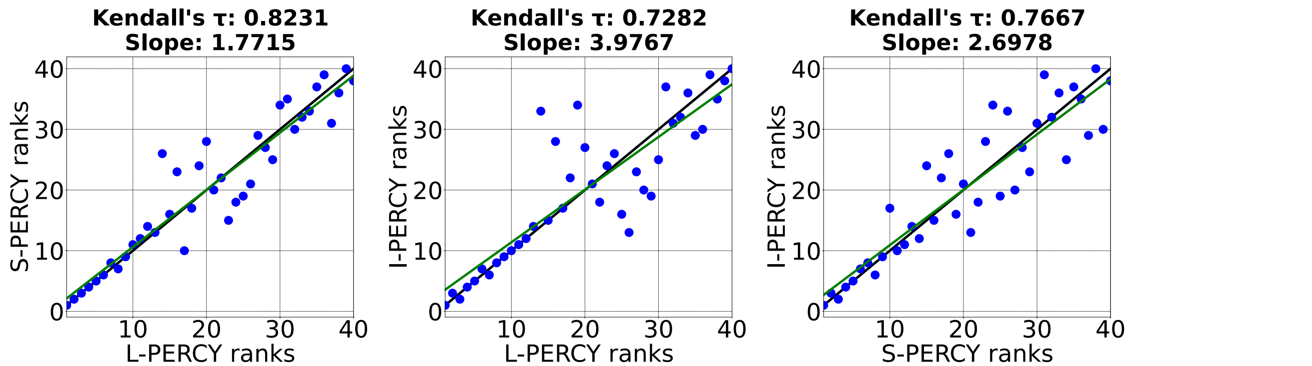
where $PERCY_{fr}$ means PERCY score calculated from a particular explanation framework (LIME, SHAP, or IG) and $X_{test_{ci}}$ means set of test-datapoints for a ith classifier in Table 3 on which PERCY scores were calculated from $PERCY_{fr}$. The results are shown in Fig. 7.

Overall, we observe that there is a significant correlation between the PERCY scores estimated using the three explanation methods as shown in Fig. 6 – all Kendall's τ values are above 0.5 indicating the classifiers share similar ranks of PERCY scores and the explanations instability do not influence the final PERCY scores calculation. These findings can be further supplemented with more granular level correlation results between PERCY scores as shown in Fig. 7 where LIME & SHAP, LIME & IG and SHAP & IG frameworks have significant positive correlation denoted by high positive Pearson's values. These values further denote that even on the data-point level, the frameworks provide similar PERCY score values.

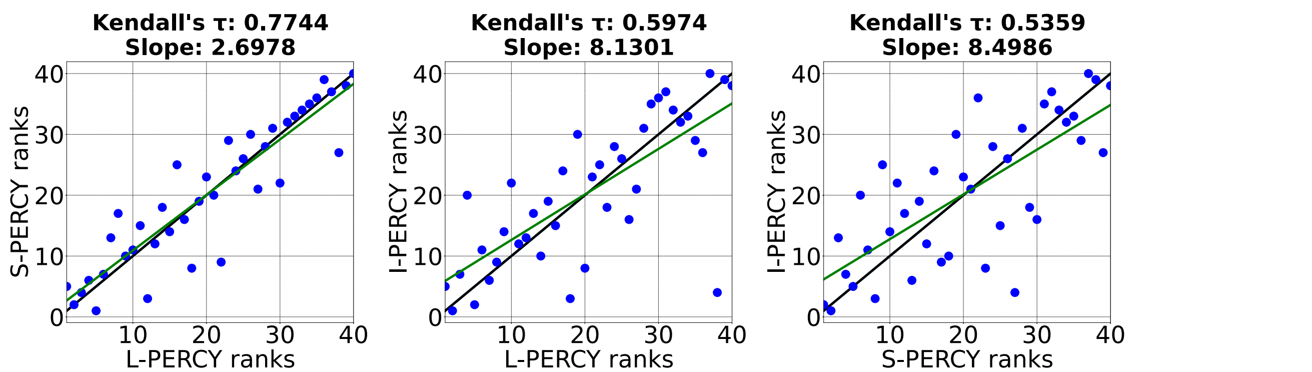
6.5. Limitations and discussion

While we believe that PERCY can provide valuable insights into the ability of knowledge dissemination methods to identify and classify contrastive discourse relations, we acknowledge that there are several limitations to our method that should be noted.

One limitation is that PERCY relies on post-hoc explanation frameworks to analyze the predictions of a given classifier. While these frameworks provide valuable insights into how a model arrived at its decision, they are not perfect and may not capture all the relevant syntactic structures in a given sentence. Additionally, PERCY assumes that the correct conjunct can be identified and extracted from the sentence, which may not always be the case in practice. Another limitation is that our study focuses specifically on contrastive discourse relations, and may not generalize to other syntactic structures or linguistic phenomena. For example, PERCY may not be effective at identifying and classifying more complex syntactic structures such as:

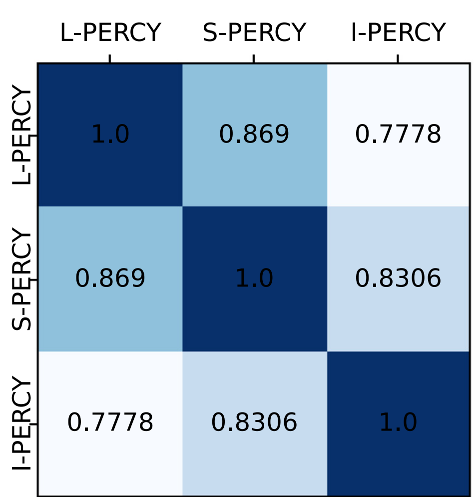


(d) Ranked correlation between PERCY scores on SST2 dataset as shown in Table 4

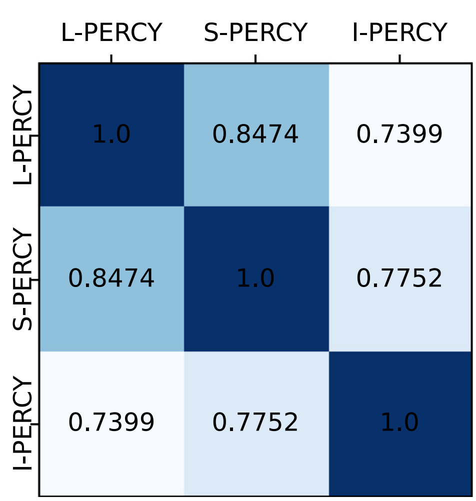


(h) Ranked correlation between PERCY scores on Sentiment140 dataset as shown in Table 5

Fig. 6. Ranked correlation scatter plots between L-PERCY, S-PERCY, and I-PERCY scores on SST2 and Sentiment140 datasets — each dot point represents a method in Table 3 with its ranking on each axis. The line $y = x$ (black) is also included to convey whether the rank of a particular method is consistent with respect to the different explainability frameworks used. Also, we include the regression line (green) to show the general trend of the data points, making it easier to observe the positive relationship between the rankings.



(a) SST2 dataset



(b) Sentiment140 dataset

Fig. 7. Pearson Correlation Heatmaps between L-PERCY, S-PERCY, and I-PERCY scores distributions on SST2 and Sentiment140 datasets respectively.

- **Complex subordination structures:** PERCY relies on identifying and extracting the correct conjunct in a sentence, which may be difficult or impossible in cases where the sentence contains complex subordination structures such as relative clauses or nested clauses.
- **Ambiguous discourse relations:** Some sentences may contain ambiguous discourse relations where it is not clear which conjunct should be considered as the main clause. For example, consider the sentence "Although he was tired, he went for a run and felt better." It is not clear whether the main clause is "he went for a run" or "he felt better", which may make it difficult to identify the appropriate conjunct to use for sentiment classification.
- **Negation and polarity:** Our method assume that the sentiment expressed in a sentence can be determined based on the sentiment words and discourse relations present in the sentence. However, in cases where the sentence contains negation or conflicting polarity, the overall sentiment may not be easily inferred from these features alone.
- **Irony and sarcasm:** Our method focuses on identifying and classifying sentiment expressed in a straightforward manner. However, some sentences may contain irony, sarcasm, or other forms of figurative language that may require a more nuanced approach to sentiment analysis.

It is important to note that there may be many other types of syntactic structures that PERCY may not be able to process effectively. We believe that further research is needed to address these limitations and to develop more robust and effective methods for incorporating logic rules into machine learning models.

7. Conclusion

This paper provides an analysis and a study of neural-symbolic methods focused on their ability to effectively disseminate logic rule knowledge in a DNN model for sentence-level binary sentiment classification task. This includes enabling a DNN model to effectively identify syntactic structures in a sentence and force the DNN model to base its decision on the appropriate conjunct. We show that accuracy or task-specific performance metric can be misleading in effectively assessing this ability. Hence, we proposed an alternative metric called PERCY, which stands for *Post-hoc Explanation-based Rule ConsistencY Score* to effectively assess the ability of a method to encode syntactic structures. We conducted an exhaustive set of experiments to support our hypothesis and concluded that the high performance of sentiment classification metrics does not necessarily indicate high rule-dissemination performance. Specific findings of our paper include that (a) accuracy – or any other performance metric – can be misleading in assessing the ability of logic rule dissemination methods to base their decisions on the right conjunct, (b) not all analyzed methods effectively capture syntactic structures, (c) often, the underlying sequence model is what captures the syntactic structure, and (d) for the best method less than 25% of test examples are classified based on the right conjunct indicating a lot of research needs to be done on this topic. Last but not least, we experimentally demonstrate that the PERCY scores calculated are robust and stable w.r.t. the feature-attribution frameworks used.

Our experiments demonstrated that in cases where a weaker sentiment is expressed in the rule conjunct of a discourse relations (e.g., after the "but" in a sentence), a naive model (e.g., CW in Table 3) that is solely based on the number or intensity of sentiment words may incorrectly classify the sentiment of the sentence based on the stronger sentiment that comes in the other conjunct (e.g., before the "but"). To address this issue, a model could incorporate a mechanism that takes into account the discourse relations in the sentence. One interesting approach to explore is to use a Rule-Mask Mechanism with, which given an input sequence predicts a vector that captures there exists an applicable logic rule on the input sequence [92]. Another approach is to use attention mechanisms that selectively focus on the important parts of the sentence, taking into account the discourse relations. For example, a model could use self-attention to weigh the importance of different words in the sentence based on their relation to other words, allowing the model to give more weight to the sentiment expression that is in the rule conjunct (e.g., after the "but"). Incorporating discourse relations and attention mechanisms can help improve the PERCY score of sentiment classification. Future work includes exploring the use of light-wise explanation frameworks to ease the calculation of the PERCY score.

CRediT authorship contribution statement

Shashank Gupta: Methodology, Software,Formal analysis, Validation, Data curation, Investigation, Writing – original draft, Visualization. **Mohamed Reda Bouadjenek:** Methodology, Formal analysis, Validation, Writing – review & editing, Supervision. **Antonio Robles-Kelly:** Methodology, Formal analysis, Validation, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used is already publicly available.

References

[1] Cicero Dos Santos, Maira Gatti, Deep convolutional neural networks for sentiment analysis of short texts, in: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 69–78.

[2] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, Neural machine translation by jointly learning to align and translate, in: Yoshua Bengio, Yann LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015, URL http://arxiv.org/abs/1409.0473.

[3] Alexander M. Rush, Sumit Chopra, Jason Weston, A neural attention model for abstractive sentence summarization, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 379–389, http://dx.doi.org/10.18653/v1/D15-1044, URL https://aclanthology.org/D15-1044.

[4] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.

[5] Cynthia Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. (2019).

[6] Stephen H. Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, Rahul Kuchhal, Chris Ré, Rob Malkin, Snorkel DryBell: A case study in deploying weak supervision at industrial scale, in: Proceedings of the 2019 International Conference on Management of Data, 2019, pp. 362–375.

[7] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus, Intriguing properties of neural networks, in: International Conference on Learning Representations, 2014.

[8] Anh Mai Nguyen, Jason Yosinski, Jeff Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, CoRR, 2014, arXiv:1412.1897.

[9] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, Explaining and harnessing adversarial examples, 2014, arXiv preprint arXiv:1412.6572.

[10] Nezihe Merve Gürel, Xiangyu Qi, Luka Rimanic, Ce Zhang, Bo Li, Knowledge enhanced machine learning pipeline against diverse adversarial attacks, in: Marina Meila, Tong Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 3976–3987, URL https://proceedings.mlr.press/v139/gurel21a.html.

[11] Artur S d'Avila Garcez, Krysia Broda, Dov M Gabbay, et al., Neural-Symbolic Learning Systems: Foundations and Applications, Springer Science & Business Media, 2002.

[12] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, Eric Xing, Harnessing deep neural networks with logic rules, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 2410–2420, http://dx.doi.org/10.18653/v1/P16-1228, URL https://www.aclweb.org/anthology/P16-1228.

[13] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, Le Song, Variational reasoning for question answering with knowledge graph, Proc. AAAI Conf. Artif. Intell. 32 (1) (2018) URL https://ojs.aaai.org/index.php/AAAI/article/view/12057.

[14] Edoardo Maria Ponti, Roi Reichart, Anna Korhonen, Ivan Vulić, Isomorphic transfer of syntactic structures in cross-lingual NLP, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1531–1542, http://dx.doi.org/10.18653/v1/P18-1142, URL https://www.aclweb.org/anthology/P18-1142.

[15] Wenya Wang, Sinno Jialin Pan, Variational deep logic network for joint inference of entities and relations, Comput. Linguist. (ISSN: 0891-2017) (2021) 1–38, http://dx.doi.org/10.1162/coli_a_00415.

[16] Kalpesh Krishna, Preethi Jyothi, Mohit Iyyer, Revisiting the importance of encoding logic rules in sentiment classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4743–4751, http://dx.doi.org/10.18653/v1/D18-1505, URL https://www.aclweb.org/anthology/D18-1505.

[17] Da Yin, Tao Meng, Kai-Wei Chang, SentiBERT: A transferable transformer-based architecture for compositional sentiment semantics, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 3695–3706, http://dx.doi.org/10.18653/v1/2020.acl-main.341, URL https://www.aclweb.org/anthology/2020.acl-main.341.

[18] Yoon Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751, http://dx.doi.org/10.3115/v1/D14-1181, URL https://www.aclweb.org/anthology/D14-1181.

[19] Robin Lakoff, If's, and's and but's about conjunction, in: Charles J. Fillmore, D. Terence Langndoen (Eds.), Studies in Linguistic Semantics, Irvington, 1971, pp. 3–114.

[20] Diane Blakemore, Denial and contrast: A relevance theoretic analysis of "but", Linguist. Philos. 12 (1) (1989) 15–37, ISSN:01650157, 15730549 URL http://www.jstor.org/stable/25001330.

[21] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "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, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450342322, 2016, pp. 1135–1144, http://dx.doi.org/10.1145/2939672.2939778, URL https://doi.org/10.1145/2939672.2939778.

[22] Scott M. Lundberg, Su-In Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS '17, Curran Associates Inc., Red Hook, NY, USA, ISBN: 9781510860964, 2017, pp. 4768–4777.

[23] Mukund Sundararajan, Ankur Taly, Qiqi Yan, Axiomatic attribution for deep networks, in: Doina Precup, Yee Whye Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 3319–3328, URL https://proceedings.mlr.press/v70/sundararajan17a.html.

[24] Alec Go, Richa Bhayani, Lei Huang, Twitter sentiment classification using distant supervision, Vol. 1, No. 12, CS224N Project Report, Stanford, 2009, p. 2009.

[25] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, Christopher Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.

[26] Subhabrata Mukherjee, P. Bhattacharyya, Sentiment analysis in Twitter with lightweight discourse analysis, in: COLING, 2012.

[27] Duyu Tang, Sentiment-specific representation learning for document-level sentiment analysis, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450333177, 2015, pp. 447–452, http://dx.doi.org/10.1145/2684822.2697035.

[28] Cäcilia Zirn, Mathias Niepert, Heiner Stuckenschmidt, Michael Strube, Fine-grained sentiment analysis with structural features, in: Proceedings of 5th International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, Chiang Mai, Thailand, 2011, pp. 336–344.

[29] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, Bonnie Webber, The penn discourse TreeBank 2.0., in: Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC'08, European Language Resources Association (ELRA), Marrakech, Morocco, 2008.

[30] Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, Eric Xing, Deep neural networks with massive learned knowledge, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 1670–1679, http://dx.doi.org/10.18653/v1/D16-1173, URL https://aclanthology.org/D16-1173.

[31] Ritesh Agarwal, T.V. Prabhakar, Sugato Chakrabarty, "I know what you feel": Analyzing the role of conjunctions in automatic sentiment analysis, in: Proceedings of the 6th International Conference on Advances in Natural Language Processing, GoTAL '08, Springer-Verlag, Berlin, Heidelberg, ISBN: 9783540852865, 2008, pp. 28–39, http://dx.doi.org/10.1007/978-3-540-85287-2_4.

[32] J. Ross Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.

[33] Didier Dubois, Henri Prade, Modelling uncertainty and inductive inference: A survey of recent non-additive probability systems, Acta Psychol. (ISSN: 0001-6918) 68 (1) (1988) 53–78, http://dx.doi.org/10.1016/0001-6918(88)90045-5, URL https://www.sciencedirect.com/science/article/pii/0001691888900455.

[34] Aidan Ed Feeney, Evan Ed Heit, Inductive reasoning: Experimental, developmental, and computational approaches., in: Fifth International Conference on Thinking, Jul, 2004, University of Leuven, Belgium; Many of the Chapter Authors for this Book Talked At the Aforementioned Symposium, Cambridge University Press, 2007.

[35] Yatin Nandwani, Abhishek Pathak, Mausam, Parag Singla, A primal dual formulation for deep learning with constraints, in: Advances in Neural Information Processing Systems, 32, Curran Associates, Inc., 2019, URL https://proceedings.neurips.cc/paper/2019/file/cf708fc1decf0337aded484f8f4519ae-Paper.pdf.

[36] Manoel V.M. França, Gerson Zaverucha, Artur S. d'Avila Garcez, Fast relational learning using bottom clause propositionalization with artificial neural networks, Mach. Learn. (ISSN: 1573-0565) 94 (1) (2014) 81–104, http://dx.doi.org/10.1007/s10994-013-5392-1.

[37] Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, Luc De Raedt, Neural probabilistic logic programming in DeepProbLog, Artificial Intelligence (ISSN: 0004-3702) 298 (2021) 103504, http://dx.doi.org/10.1016/j.artint.2021.103504, URL https://www.sciencedirect.com/science/article/pii/S0004370221000552.

[38] Geoffrey G. Towell, Jude W. Shavlik, Knowledge-based artificial neural networks, Artificial Intelligence (ISSN: 0004-3702) 70 (1) (1994) 119–165, http://dx.doi.org/10.1016/0004-3702(94)90105-8, URL https://www.sciencedirect.com/science/article/pii/0004370294901058.

[39] Richard Evans, Edward Grefenstette, Learning explanatory rules from noisy data, J. Artificial Intelligence Res. 61 (2018) 1–64.

[40] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, Luc De Raedt, Deepproblog: Neural probabilistic logic programming, Adv. Neural Inf. Process. Syst. 31 (2018).

[41] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, Guy Broeck, A semantic loss function for deep learning with symbolic knowledge, in: International Conference on Machine Learning, PMLR, 2018, pp. 5502–5511.

[42] Qika Lin, Rui Mao, Jun Liu, Fangzhi Xu, Erik Cambria, Fusing topology contexts and logical rules in language models for knowledge graph completion, Inf. Fusion (ISSN: 1566-2535) 90 (2023) 253–264, http://dx.doi.org/10.1016/j.inffus.2022.09.020, URL https://www.sciencedirect.com/science/article/pii/S1566253522001592.

[43] Geoffrey Hinton, Oriol Vinyals, Jeffrey Dean, Distilling the knowledge in a neural network, in: NIPS Deep Learning and Representation Learning Workshop, 2015, URL http://arxiv.org/abs/1503.02531.

[44] Bowen Zhang, Xiaofei Xu, Xutao Li, Xiaojun Chen, Yunming Ye, Zhongjie Wang, Sentiment analysis through critic learning for optimizing convolutional neural networks with rules, Neurocomputing (ISSN: 0925-2312) 356 (2019) 21–30, http://dx.doi.org/10.1016/j.neucom.2019.04.038, URL https://www.sciencedirect.com/science/article/pii/S0925231219306198.

[45] Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, Kenneth Kwok, SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 3829–3839, URL https://aclanthology.org/2022.lrec-1.408.

[46] Bingfeng Chen, Zhifeng Hao, Xiaofeng Cai, Ruichu Cai, Wen Wen, Jian Zhu, Guangqiang Xie, Embedding logic rules into recurrent neural networks, IEEE Access 7 (2019) 14938–14946, http://dx.doi.org/10.1109/ACCESS.2019.2892140.
[47] Wenya Wang, Sinno Jialin Pan, Integrating deep learning with logic fusion for information extraction, Proceedings of the AAAI Conference on Artificial Intelligence 34 (05) (2020) 9225–9232, http://dx.doi.org/10.1609/aaai.v34i05.6460, URL https://ojs.aaai.org/index.php/AAAI/article/view/6460.
[48] Tao Li, Vivek Srikumar, Augmenting neural networks with first-order logic, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 292–302, http://dx.doi.org/10.18653/v1/P19-1028, URL https://aclanthology.org/P19-1028.
[49] Hai Wang, Hoifung Poon, Deep probabilistic logic: A unifying framework for indirect supervision, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1891–1902, http://dx.doi.org/10.18653/v1/D18-1215, URL https://aclanthology.org/D18-1215.
[50] Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, Maosong Sun, Train no evil: Selective masking for task-guided pre-training, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2020, pp. 6966–6974, http://dx.doi.org/10.18653/v1/2020.emnlp-main.566, Online, URL https://aclanthology.org/2020.emnlp-main.566.
[51] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, Deep contextualized word representations, in: Proc. of NAACL, 2018.
[52] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, Francisco Herrera, Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion (ISSN: 1566-2535) 58 (2020) 82–115, http://dx.doi.org/10.1016/j.inffus.2019.12.012, URL https://www.sciencedirect.com/science/article/pii/S1566253519308103.
[53] Haydemar Nunez, Cecilio Angulo, Andreu Catala, Rule-based learning systems for support vector machines, Neural Process. Lett. 24 (2006) 1–18.
[54] Ulf Johansson, Rikard König, Lars Niklasson, The truth is in there - rule extraction from opaque models using genetic programming, 2, 2004, pp. 658–663, URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-10044295085&partnerID=40&md5=1d15ff1434dbae8c8772d137d141fdf5. Cited by: 39,
[55] J.R. Quinlan, Simplifying decision trees, Int. J. Man-Mach. Stud. (ISSN: 0020-7373) 27 (3) (1987) 221–234, http://dx.doi.org/10.1016/S0020-7373(87)80053-6, URL https://www.sciencedirect.com/science/article/pii/S0020737387800536.
[56] Laurent Hyafil, Ronald L. Rivest, Constructing optimal binary decision trees is NP-complete, Inform. Process. Lett. (ISSN: 0020-0190) 5 (1) (1976) 15–17, http://dx.doi.org/10.1016/0020-0190(76)90095-8, URL https://www.sciencedirect.com/science/article/pii/0020019076900958.
[57] Paul E. Utgoff, Incremental induction of decision trees, Mach. Learn. (ISSN: 0885-6125) 4 (2) (1989) 161–186, http://dx.doi.org/10.1023/A:1022699900025, URL https://doi.org/10.1023/A:1022699900025.
[58] Wei Li, Luyao Zhu, Rui Mao, Erik Cambria, SKIER: A symbolic knowledge integrated model for conversational emotion recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023.
[59] H. Tsukimoto, Extracting rules from trained neural networks, IEEE Trans. Neural Netw. 11 (2) (2000) 377–389, http://dx.doi.org/10.1109/72.839008.
[60] Zhengping Che, Sanjay Purushotham, Robinder Khemani, Yan Liu, Interpretable deep models for ICU outcome prediction, in: AMIA Annual Symposium Proceedings, Vol. 2016, American Medical Informatics Association, 2016, p. 371.
[61] Quanshi Zhang, Ying Nian Wu, Song-Chun Zhu, Interpretable convolutional neural networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8827–8836, http://dx.doi.org/10.1109/CVPR.2018.00920.
[62] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017.
[63] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, Wojciech Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One 10 (7) (2015) 1–46, http://dx.doi.org/10.1371/journal.pone.0130140.
[64] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, Klaus-Robert Müller, Explaining nonlinear classification decisions with deep taylor decomposition, Pattern Recognit. (ISSN: 0031-3203) 65 (2017) 211–222, http://dx.doi.org/10.1016/j.patcog.2016.11.008, URL https://www.sciencedirect.com/science/article/pii/S0031320316303582.
[65] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, Anshul Kundaje, Not just a black box: Learning important features through propagating activation differences, 2016, arXiv preprint arXiv:1605.01713.
[66] Sooji Han, Rui Mao, Erik Cambria, Hierarchical attention network for explainable depression detection on Twitter aided by metaphor concept mappings, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 94–104, URL https://aclanthology.org/2022.coling-1.9.
[67] Sandra Wachter, Brent Mittelstadt, Chris Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, Harv. J. Law Technol. 31 (2) (2018) https://ssrn.com/abstract=3063289.
[68] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, Peter Flach, FACE: Feasible and actionable counterfactual explanations, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES '20, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450371100, 2020, pp. 344–350, http://dx.doi.org/10.1145/3375627.3375850.
[69] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, Anchors: High-precision model-agnostic explanations, Proc. AAAI Conf. Artif. Intell. 32 (1) (2018) http://dx.doi.org/10.1609/aaai.v32i1.11491, URL https://ojs.aaai.org/index.php/AAAI/article/view/11491.
[70] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, Dino Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. (ISSN: 0360-0300) 51 (5) (2018) http://dx.doi.org/10.1145/3236009.
[71] McDermid John A., Jia Yan, Porter Zoe, Habli Ibrahim, Artificial intelligence explainability: the technical and ethical dimensions, Phil. Trans. R. Soc. A. (2021) http://dx.doi.org/10.1098/rsta.2020.0363, 3792020036320200363.
[72] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, Prithviraj Sen, A survey of the state of explainable AI for natural language processing, in: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2020, pp. 447–459, URL https://aclanthology.org/2020.aacl-main.46.
[73] Timo Speith, A review of taxonomies of explainable artificial intelligence (XAI) methods, in: 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450393522, 2022, pp. 2239–2250.
[74] Ninghao Liu, Xiao Huang, Jundong Li, Xia Hu, On interpretation of network embedding via taxonomy induction, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450355520, 2018, pp. 1812–1820, http://dx.doi.org/10.1145/3219819.3220001.
[75] Kuzman Ganchev, João Graça, Jennifer Gillenwater, Ben Taskar, Posterior regularization for structured latent variable models, J. Mach. Learn. Res. 11 (67) (2010) 2001–2049, URL http://jmlr.org/papers/v11/ganchev10a.html.
[76] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, Jeff Dean, Distributed representations of words and phrases and their compositionality, in: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol.26, Curran Associates, Inc., 2013.
[77] Jeffrey Pennington, Richard Socher, Christopher D. Manning, GloVe: Global vectors for word representation, in: Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543, URL http://www.aclweb.org/anthology/D14-1162.
[78] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, http://dx.doi.org/10.18653/v1/N19-1423, URL https://www.aclweb.org/anthology/N19-1423.
[79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, Illia Polosukhin, Attention is all you need, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.
[80] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, Improving Language Understanding by Generative Pre-Training, 2018, OpenAI.
[81] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, Noah A. Smith, Don't stop pretraining: Adapt language models to domains and tasks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8342–8360, http://dx.doi.org/10.18653/v1/2020.acl-main.740, URL https://aclanthology.org/2020.acl-main.740.
[82] Iz Beltagy, Kyle Lo, Arman Cohan, SciBERT: A pretrained language model for scientific text, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for

Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620, http://dx.doi.org/10.18653/v1/D19-1371, URL https://aclanthology.org/D19-1371.

[83] Alexander Rietzler, Sebastian Stabinger, Paul Opitz, Stefan Engl, Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification, in: Proceedings of the 12th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 4933–4941, URL https://aclanthology.org/2020.lrec-1.607.

[84] Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, Minlie Huang, SentiLARE: Sentiment-aware language representation learning with linguistic knowledge, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Online, 2020, pp. 6975–6988, http://dx.doi.org/10.18653/v1/2020.emnlp-main.567, URL https://aclanthology.org/2020.emnlp-main.567.

[85] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, Omer Levy, SpanBERT: Improving pre-training by representing and predicting spans, Trans. Assoc. Comput. Linguist. 8 (2020) 64–77, http://dx.doi.org/10.1162/tacl_a_00300, URL https://aclanthology.org/2020.tacl-1.5.

[86] Ye Zhang, Byron Wallace, A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification, in: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Asian Federation of Natural Language Processing, Taipei, Taiwan, 2017, pp. 253–263, URL https://www.aclweb.org/anthology/I17-1026.

[87] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. (ISSN: 0899-7667) 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[88] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, in: Yoshua Bengio, Yann LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015, URL http://arxiv.org/abs/1412.6980.

[89] M.G. Kendall, A new measure of rank correlation, Biometrika (ISSN: 0006-3444) 30 (1–2) (1938) 81–93, http://dx.doi.org/10.1093/biomet/30.1-2.81.

[90] David Alvarez-Melis, Tommi S. Jaakkola, On the robustness of interpretability methods, in: 2018 ICML Workshop on Human Interpretability in Machine Learning, 2018.

[91] Dina Mardaoui, Damien Garreau, An analysis of LIME for text data, in: Arindam Banerjee, Kenji Fukumizu (Eds.), Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 130, PMLR, 2021, pp. 3493–3501, URL https://proceedings.mlr.press/v130/mardaoui21a.html.

[92] Shashank Gupta, Mohamed Reda Bouadjenek, Antonio Robles-Kelly, A mask-based logic rules dissemination method for sentiment classifiers, in: Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, Annalina Caputo (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, ISBN: 978-3-031-28244-7, 2023, pp. 394–408.