Learning to Explain: Generating Stable Explanations Fast

Xuelin Situ[†] **Ingrid Zukerman**[†] **Cecile Paris**[‡]
Sameen Maruf[†] **Gholamreza Haffari**[†]
[†]Dept of Data Science and AI, Faculty of IT, Monash University, Australia
[‡]CSIRO Data61, Australia
[†]`{firstname.lastname}@monash.edu`
[‡]`{firstname.lastname}@data61.csiro.au`

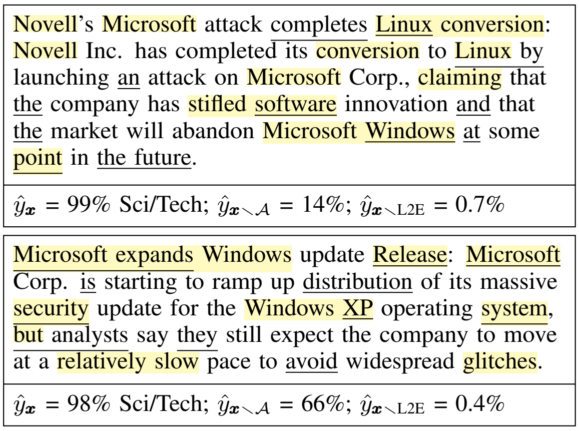
Abstract

The importance of explaining the outcome of a machine learning model, especially a black-box model, is widely acknowledged. Recent approaches explain an outcome by identifying the contributions of input features to this outcome. In environments involving large black-box models or complex inputs, this leads to computationally demanding algorithms. Further, these algorithms often suffer from low stability, with explanations varying significantly across similar examples. In this paper, we propose a Learning to Explain (L2E) approach that learns the behaviour of an underlying explanation algorithm simultaneously from all training examples. Once the explanation algorithm is distilled into an explainer network, it can be used to explain new instances. Our experiments on three classification tasks, which compare our approach to six explanation algorithms, show that L2E is between 5 and 7.5×10^4 times faster than these algorithms, while generating more stable explanations, and having comparable faithfulness to the black-box model.

1 Introduction

Explaining the mechanisms and reasoning behind the outcome of complex machine learning models, such as deep neural networks (DNNs), is crucial. Such explanations can shed light on the potential flaws and biases within these powerful and widely applicable models, e.g., in medical diagnosis (Caruana et al., 2015) and judicial systems (Rich, 2016).

Existing explainability methods mostly produce explanations, or rationales (DeYoung et al., 2020), which identify the attributions of features in an input example, e.g., are they contributing positively or negatively to the prediction of an outcome. For text classifiers, this means identifying words or phrases in an input document that account for a prediction. Current approaches are typically computationally demanding, requiring expensive operations, such as consulting a black-box model multiple times (Zeiler and Fergus, 2014), or generating samples to learn an approximate but explainable transparent model (Ribeiro et al., 2016). This computational demand reduces the utility of these explanation algorithms, especially for large black-box models, long documents and real-time scenarios (Kim et al., 2018). Further, these algorithms generate explanations for different examples independently. This may lead to the generation of different explanations for similar examples, which is undesirable. For example, a black-box predicts with similar confidence (99% and 98%) that the topic of the two semantically similar documents in Figure 1 is Sci/Tech. However, even though the words 'Microsoft' and 'Windows' appear in both documents, the baseline explainer $\mathcal{A}$ deems 'Windows' to be important for the top document, and 'Microsoft' for the bottom document (that is, mask-



Novell's Microsoft attack completes Linux conversion: Novell Inc. has completed its conversion to Linux by launching an attack on Microsoft Corp., claiming that the company has stifled software innovation and that the market will abandon Microsoft Windows at some point in the future.

$\hat{y}_{\boldsymbol{x}}$ = 99% Sci/Tech; $\hat{y}_{\boldsymbol{x}\setminus\mathcal{A}}$ = 14%; $\hat{y}_{\boldsymbol{x}\setminus\mathrm{L2E}}$ = 0.7%

Microsoft expands Windows update Release: Microsoft Corp. is starting to ramp up distribution of its massive security update for the Windows XP operating system, but analysts say they still expect the company to move at a relatively slow pace to avoid widespread glitches.

$\hat{y}_{\boldsymbol{x}}$ = 98% Sci/Tech; $\hat{y}_{\boldsymbol{x}\setminus\mathcal{A}}$ = 66%; $\hat{y}_{\boldsymbol{x}\setminus\mathrm{L2E}}$ = 0.4%

Figure 1: Two similar examples from the News dataset. The most important words (top 30%) found by our method L2E are yellow-highlighted, and those from a baseline $\mathcal{A}$ are underlined. L2E considers words like 'Microsoft' and 'Windows' important in both examples. $\hat{y}_{\boldsymbol{x}}$ is the model's prediction, and $\hat{y}_{\boldsymbol{x}\setminus\cdot}$ is the model's prediction after removing important words in $\boldsymbol{x}$.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing, pages 5340–5355
August 1–6, 2021. ©2021 Association for Computational Linguistics

ing these words results in a significant drop in the black-box's confidence).

In this paper, we present a learning to explain (L2E) approach that efficiently learns the commonalities of the explanation process across different examples. This, in turn, leads to explanations that exhibit stability, i.e., important words are chosen consistently, without loss of faithfulness to the underlying black-box.[1] Given a set of examples paired with their explanations produced by an existing method, e.g., LIME (Ribeiro et al., 2016), our approach uses a DNN to learn the explanation algorithm. DNNs are Turing complete (Pérez et al., 2019; Montufar et al., 2014); therefore, given enough training data and learning capacity, they should be able to learn the existing explanation algorithms. This is akin to Knowledge Distillation (Hinton et al., 2015), where a teacher, or in our case a teacher algorithm, distils knowledge into a student network.

Our contributions are: (i) the L2E framework, which is general, and can successfully learn to produce explanations from several teacher explainers; (ii) two learning formulations, i.e., Ranking and Sequence Labelling, to enable L2E to circumvent the high variance of non-discrete teacher explanations via discretization; (iii) an experimental setup to compare L2E against six popular explanation algorithms, and a comprehensive evaluation to investigate the stability and faithfulness of L2E on three text classification tasks; (iv) a methodology that employs *human rationales* as proxies for the ground-truth explanations of a black-box model. The core of this method is a modified training protocol whereby the model makes neutral predictions if human rationales are absent.

2 Related Work

We consider two main approaches to explanation generation: algorithmic and model-based.

Algorithmic Approaches. These approaches can be broadly categorized into gradient-based, attention-based and perturbation-based methods.

Gradient-based methods (Simonyan et al., 2013; Sundararajan et al., 2017; Shrikumar et al., 2017; Erion et al., 2019) or backpropagation-based methods (Bach et al., 2015) require access to the black-box, and are mostly applied to models with differentiable functions. Further, they may be sensitive to randomized model initializations or permuted data labels (Adebayo et al., 2018), which is undesirable. These methods can be computationally heavy in the case of complex black-box models (Wu and Ong, 2021), e.g., BERT (Devlin et al., 2018).

Attention-based methods (Wiegreffe and Pinter, 2019) can only be applied to Transformer-based models (Vaswani et al., 2017), and their effectiveness is questionable (Jain and Wallace, 2019; Serrano and Smith, 2019).

Perturbation-based methods approximate feature importance by observing changes in a model's outcome after a feature is changed. They either consider changes in performance as an indicator of feature importance directly (Martens and Provost, 2014; Zeiler and Fergus, 2014; Schwab and Karlen, 2019), or they employ a higher-order approximation of the decision boundary (Ribeiro et al., 2016; Lundberg and Lee, 2017). Perturbation-based methods are typically computationally inefficient for explaining high-dimensional data, and they suffer from high variance due to perturbation randomness (Slack et al., 2020; Chen et al., 2019).

Model-based Approaches. These approaches train the explainer with an objective function to improve efficiency at test time. The closest work to ours is by Schwab and Karlen (2019), who train an explainer using a causality-based explanation algorithm. However, these approaches do not learn from arbitrary algorithms or discretize feature weights — the high variation of continuous weights may impair the ability to capture the commonalities in an explanation algorithm. Jain et al. (2020) discretize the weights produced by an existing method, but they use these weights to build a faithful classifier for an underlying black-box model, rather than using them to explain the model directly.

Other works train a classifier and an explainer jointly in order to incorporate explainability directly into the classifier (Lei et al., 2016; Camburu et al., 2018). Unlike these approaches, we do not change the classifier or require an expensive process to collect human rationales, as done in (Camburu et al., 2018). Lastly, a few works use information-theoretic objectives to train an explainer directly from the underlying classifier (Chen et al., 2018; Bang et al., 2019). These explainers require careful training to select a low number of important features (Paranjape et al., 2020); hence, some input features do not have attributions.

[1] This approach does not aim to improve the transparency (Lipton, 2018) of the black-box model.

Goodness of Explanations. Researchers have quantified the goodness of an explanation in different ways, such as brevity, alignment to human rationales, contrastiveness and stability.

Minimal (brief) explanations are generated in (Martens and Provost, 2014; Ribeiro et al., 2018; Alvarez-Melis et al., 2019; Bang et al., 2019). Explanations aligned with human rationales are produced in (Sen et al., 2020; Atanasova et al., 2020), and contrastive explanations are generated in (Miller, 2018; Alvarez-Melis et al., 2019).

According to Atanasova et al. (2020), only a few algorithmic explanation methods produce stable explanations (Robnik-Šikonja and Bohanec, 2018), e.g., LIME (Ribeiro et al., 2016). To the best of our knowledge, we are the first to explore the stability of explanations in model-based approaches.

3 Learning to Explain (L2E)

L2E can be applied to any Natural Language Processing task to which an underlying feature-based explanation algorithm can be applied, such as Natural Language Inference and Question Answering (Wang et al., 2020). In this paper, we focus on explaining text classification models.

Our setup requires two inputs: (i) a black-box text classification model $\hat{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, which assigns document $\boldsymbol{x}$ to a label $\hat{y} \in Y$, where Y is the label set; and (ii) an explanation algorithm $\mathcal{A}(\boldsymbol{x}, \hat{y}, f_{\boldsymbol{\theta}}) \rightarrow \boldsymbol{w}$, which generates explanation $\boldsymbol{w} \in \mathbb{R}^{|\boldsymbol{x}|}$ for the class of document $\boldsymbol{x}$ obtained by the black-box $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. $\mathcal{A}$ can be any off-the-shelf explanation algorithm; and w_i can be thought as the importance weight of x_i – the i^{th} token of a document.

The main idea of L2E is to train a separate explanation model $g_{\boldsymbol{\phi}}(\boldsymbol{x})$ to predict the explanation generated by $\mathcal{A}(.)$ for $f_{\boldsymbol{\theta}}(.)$ (Figures 2a and 2b). Intuitively, our approach distils the explanation algorithm $\mathcal{A}$ into the explanation model $g_{\boldsymbol{\phi}}$. As confirmed by our experiments (§4.5), this has several benefits. Firstly, it leads to stable explanations, as $g_{\boldsymbol{\phi}}$ can capture $\mathcal{A}$'s common patterns when generating explanations for different documents. Secondly, it speeds up the explanation generation process compared to many existing explanation algorithms, which rely on computationally heavy operations, such as consulting the black-box model multiple times, e.g., Occlusion (Zeiler and Fergus, 2014), or sampling, e.g., LIME (Ribeiro et al., 2016). Our approach, which learns a model with explanations of all training data, takes advantage of the computations done by $\mathcal{A}$, and generates more stable explanations faster.

Algorithm 1 Learning to Explain (L2E)

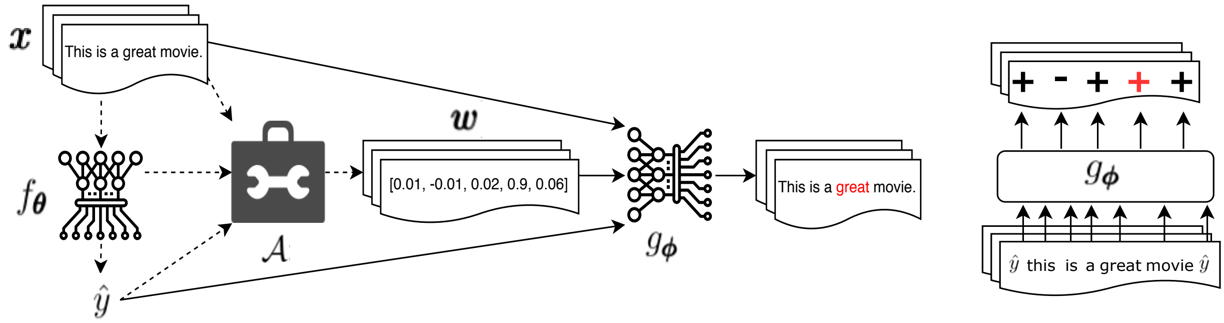
```
 1: $D$: a training set of documents
 2: $f_{\boldsymbol{\theta}}$: the original deep NN model
 3: $g_{\boldsymbol{\phi}}$: the explainer deep NN model
 4: $\mathcal{A}$: the underlying explanation method
 5: procedure TrainExplainer($D$, $f_{\boldsymbol{\theta}}$)
 6:     $Z \leftarrow \emptyset$
 7:     for each input $\boldsymbol{x} \in D$ do
 8:         $\hat{y} \leftarrow f_{\boldsymbol{\theta}}(\boldsymbol{x})$
 9:         $\boldsymbol{w} \leftarrow \mathcal{A}(\boldsymbol{x}, \hat{y}, f_{\boldsymbol{\theta}})$
10:         $Z \leftarrow Z \cup (\boldsymbol{x}, \hat{y}, \boldsymbol{w})$
11:     end for
12:     initialize $\boldsymbol{\phi}$ randomly
13:     $t \leftarrow 0$
14:     while a stopping condition is not met do
15:         Randomly pick $(\boldsymbol{x}_t, \hat{y}_t, \boldsymbol{w}_t) \in Z$
16:         $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \eta_t \nabla_{\boldsymbol{\phi}} \mathcal{L}(g_{\boldsymbol{\phi}}(\boldsymbol{x}_t, \hat{y}_t), \boldsymbol{w}_t)$
17:         $t \leftarrow t + 1$
18:     end while
19:     return the explanation model $g_{\boldsymbol{\phi}}$
20: end procedure
```

Our approach to train the explanation model $g_{\boldsymbol{\phi}}$ is summarized in Algorithm 1. First, the algorithm generates training data in the form of triplets $(\boldsymbol{x}, \hat{y}, \boldsymbol{w})$ (lines 7–11), and then it trains the explanation model using supervised learning (lines 14–18). At test time, the trained model is deployed to generate explanations for unseen documents.

A crucial component in training the explanation model under supervised learning is the loss function $\mathcal{L}(g_{\boldsymbol{\phi}}(\boldsymbol{x}_t, \hat{y}), \boldsymbol{w})$. It penalizes a deviation of the predicted explanation $g_{\boldsymbol{\phi}}(\boldsymbol{x}_t, \hat{y})$ from the ground truth explanation $\boldsymbol{w}$. This loss function is determined by our supervised learning formulation.

Given that $\boldsymbol{w}$ is a continuous-valued vector, learning the model $g_{\boldsymbol{\theta}}$ may be cast as a multivariate regression problem. However, the continuous feature attributions generated by existing explanation algorithms could be sensitive to initializations (Slack et al., 2020). Further, manually annotated rationales (highlighting important words in a document) are sufficient for people to understand/perform a classification task (Zaidan et al., 2007). So, instead of a regression formulation, we consider two supervised learning formulations for discretized outputs: Ranking and Sequence Labeling.

Ranking Formulation. In this formulation, the explanation model aims to learn the ranking of the document tokens from their importance weights. That is, we consider the ordering of the token weights induced by $\boldsymbol{w}$, and train the explanation



(a)

(b)

Figure 2: (a) Pipeline of our L2E method; dashed arrows represent offline processes. (b) Detailed input and output for the sequence labeling formulation of our explanation model; red + label indicates that $g_{\boldsymbol{\theta}}$ considers 'great' to be more important than other words in the prediction $\hat{y}$.

model $g_{\boldsymbol{\phi}}$ such that it induces the same ordering. Specifically, the loss function is as follows:

$$\mathcal{L}(g_{\boldsymbol{\phi}}(\boldsymbol{x},\hat{y}),\boldsymbol{w}) = -\sum_{i=1}^{|\boldsymbol{x}|-1}\sum_{j=i+1}^{|\boldsymbol{x}|}\log\frac{e^{v_k}}{e^{v_i}+e^{v_j}}$$

where v_i (v_j) is the i^{th} (j^{th}) component of the importance vector $\boldsymbol{v} = g_{\boldsymbol{\phi}}(\boldsymbol{x},\hat{y})$ predicted by the explanation model, and $k = \arg\max_{k'\in\{i,j\}}|w_{k'}|$. In other words, each pair of token weights is compared, and the parameters are learnt such that a token with a high importance weight under $\mathcal{A}$ also gets a high score under $g_{\boldsymbol{\phi}}$.

Sequence Labeling Formulation. Here, explanation generation is treated as a sequence labeling problem, where the continuous importance weights are discretized according to the heuristic h, whereby the importance weights are partitioned along two dimensions, high/low and positive/neutral/negative, according to the mean value of the positive/negative weights from the baseline explanation method $\mathcal{A}$. Thus, the labels are recoded to {high negative, low negative, neutral, low positive, high positive}. The explanation model $g_{\boldsymbol{\phi}}$ is then trained to predict the label of the tokens according to the following loss function:

$$\mathcal{L}(g_{\boldsymbol{\phi}}(\boldsymbol{x},\hat{y}),\boldsymbol{w}) = -\sum_{i=1}^{|\boldsymbol{x}|}\log\Pr(h(w_i)|g_{\boldsymbol{\phi},i}(\boldsymbol{x},\hat{y}))$$

where $g_{\boldsymbol{\phi},i}(\boldsymbol{x},\hat{y})$ is the predicted distribution over the labels of the i^{th} token of the document, and $h(w_i)$ is the discrete label produced using the discretization heuristic h.

Owing to the quadratic complexity of the Ranking formulation, compared to the linear complexity of Sequence Labeling, we recommend using Ranking when the input is short, and a fine-grained order of feature attributions is required. Otherwise, the Sequence Labeling formulation is a better option.

4 Experiments

4.1 Tasks and Black-Box Models ($f_{\boldsymbol{\theta}}$)

We conduct experiments on three classification tasks; each task has a different black-box classifier chosen based on the best accuracy on the selected dataset as reported in the literature.[2] Dataset statistics are reported in Appendix A.

- **Topic Classification.** The AG corpus (Zhang et al., 2015) comprises news articles on multiple topics. We separate 10% of the training documents for the dev set. The black-box classifier is a fine-tuned BERT model (Devlin et al., 2018) with 12 hidden layers and 12 attention heads. It achieves a 92.6% test accuracy.

- **Sentiment Analysis.** The SST dataset (Socher et al., 2013) comprises movie reviews with positive and negative sentiments. The black-box classifier is a distilled BERT model (Sanh et al., 2019) with 6 layers and 12 attention heads from Hugging Face (Wolf et al., 2019). It achieves 90% test accuracy.

- **Linguistic Acceptability.** The CoLA dataset (Warstadt et al., 2019) contains sentences that are deemed acceptable or unacceptable in terms of their grammatical correctness. The black-box classifier is a fine-tuned ALBERT model (Lan et al., 2020) with 12 attention heads and 12 layers. It achieves a 74% test accuracy.

[2] All black-box models are open-sourced by TextAttack (Morris et al., 2020) unless otherwise stated.

4.2 Baseline Explanation Methods ($\mathcal{A}$)

We use six baselines for our experimental setup: Occlusion (Zeiler and Fergus, 2014; Schwab and Karlen, 2019), Gradient (Simonyan et al., 2013), LRP (Bach et al., 2015), LIME (Ribeiro et al., 2016), Kernel SHAP (Lundberg and Lee, 2017) and Deep SHAP (Shrikumar et al., 2017; Lundberg and Lee, 2017). The detailed setup of these baselines is provided in Appendix B.

4.3 Explanation Models ($g_{\boldsymbol{\phi}}$)

We use a Transformer encoder (Vaswani et al., 2017) with 4 blocks and 4 attention heads as $g_{\boldsymbol{\phi}}$.[3] All models are trained with a Stochastic Gradient Descent optimizer and a fixed learning rate ($1e^{-4}$) until convergence. To balance the different statuses of model convergence, we train all models with three random parameter initializations and report the average values of their performance metrics.

We condition the explainer model $g_{\boldsymbol{\phi}}$ on the label $\hat{y}$ predicted by the underlying black-box model $f_{\boldsymbol{\theta}}$ by appending $\hat{y}$ to the start and the end of the input document before passing it to $g_{\boldsymbol{\phi}}$ (Figure 2a). Thus, $g_{\boldsymbol{\phi}}$ can leverage the predicted label in the attention computation. For the sequence labeling formulation, we also introduce a softmax layer on top to produce the labeling distribution over the discrete labels for each token, as detailed in Figure 2b.

4.4 Performance Metrics

Faithfulness. A standard approach to evaluate the faithfulness of an explanation to a black-box classification model is to measure the degree of agreement between the prediction given the full document and the prediction given the explanation (Ribeiro et al., 2016). However, the aim of L2E is to approximate an existing explanation method $\mathcal{A}$, which constitutes a layer of separation from the original black-box $f_{\boldsymbol{\theta}}$. Hence, we provide two faithfulness evaluations for our approach when the ground-truth explanation is unavailable:

- *Prediction based.* We measure the agreement between: (a) the predictions of the black-box model $f_{\boldsymbol{\theta}}$ when the explanations generated by $g_{\boldsymbol{\phi}}$ are given as input, and (b) $f_{\boldsymbol{\theta}}$'s predictions when $\mathcal{A}$'s explanations are given as input (instead of using the full document);[4]
- *Confidence based.* We adopt the Δlog-odds$(\boldsymbol{x})$ metric used by Schwab and Karlen (2019), which measures the difference in the confidence of the $f_{\boldsymbol{\theta}}$ black-box model in a prediction before and after masking the words in an explanation.

  $$\text{log-odds}(\Pr(\hat{y}|f_{\boldsymbol{\theta}}(\boldsymbol{x}))) - \text{log-odds}(\Pr(\hat{y}|f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})))$$

  where $\hat{y}$ is the predicted output of $f_{\boldsymbol{\theta}}(\boldsymbol{x})$, $\text{log-odds}(\Pr) = \log \frac{\Pr}{1-\Pr}$, and $\tilde{\boldsymbol{x}}$ is a version of input $\boldsymbol{x}$ where the tokens in the explanation are masked out. We expect a high Δlog-odds value if we mask positive important words in $\tilde{\boldsymbol{x}}$, and a low value if we mask unimportant or negative important words.

We report the average of each of these metrics across the test documents.

Stability. We employ *Intersection over Union* (*IoU*) to measure explanation stability across similar instances. Specifically, for each test instance $\boldsymbol{x}$, we select its nearest neighbors $\mathcal{N}(\boldsymbol{x})$ according to one of two pairwise document similarity metrics: semantic similarity – cosine of their BERT representations; and lexical similarity – ratio of overlapping n-grams. Details appear in Appendix C. $\text{IoU}(\boldsymbol{x}, \mathcal{N}(\boldsymbol{x}))$ then measures the consistency of explanations of $\boldsymbol{x}$ and those of its neighbours,

$$\frac{1}{|\mathcal{N}(\boldsymbol{x})|} \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x})} \frac{\sum_{\substack{\ell \in L \\ \ell \neq \text{neutral}}} |\boldsymbol{v}_{\boldsymbol{x}}^{\ell} \cap \boldsymbol{v}_{\boldsymbol{x}'}^{\ell}|}{\sum_{\substack{\ell \in L \\ \ell \neq \text{neutral}}} |\boldsymbol{v}_{\boldsymbol{x}}^{\ell} \cup \boldsymbol{v}_{\boldsymbol{x}'}^{\ell}|} \tag{1}$$

where L is the discretized label set in the Sequence Labeling formulation or the top K words in the Ranking formulation, and $\boldsymbol{v}_{\boldsymbol{x}}^{\ell}$ is the set of tokens with label ℓ in the predicted explanation $g_{\boldsymbol{\phi}}(\boldsymbol{x}, \hat{y})$. We report the average of $\text{IoU}(\boldsymbol{x}, \mathcal{N}(\boldsymbol{x}))$ across documents in the test set.

4.5 Results and Discussion

We start by investigating the faithfulness of an explanation model to the black-box model $f_{\boldsymbol{\theta}}$. Once faithfulness has been established, we investigate stability and speed compared to the underlying explanation methods $\mathcal{A}$. We also include a *Random* baseline, which displays the performance obtained by randomly selecting the same K number of words as we select from explanations produced by L2E and $\mathcal{A}$ in each row of the table, and averaging it over the six comparisons.

[3] We use the fairseq framework (Ott et al., 2019) for all our implementations of $g_{\boldsymbol{\phi}}$. Our source code is available at https://github.com/situsnow/L2E.

[4] We do not evaluate the faithfulness of L2E to $\mathcal{A}$ in terms of token importance, because $\mathcal{A}$ is not always faithful to the black-box model.

		Ranking			Seq. Labeling		
		News	SST	CoLA	News	SST	CoLA
Random		60	37	65	43.6	45.8	70
Occ.	$\mathcal{A}$	94	88	67	81	89	76
	L2E	89	83	65	84	90	80
	Both	89	85	98	82	85	94
Grad.	$\mathcal{A}$	89	73	65	87	71	68
	L2E	91	66	65	94	70	70
	Both	85	73	100	89	77	92
LRP	$\mathcal{A}$	97	95	67	85	83	62
	L2E	90	83	67	86	85	52
	Both	93	84	100	90	83	90
LIME	$\mathcal{A}$	100	86	67	99	84	71
	L2E	96	81	65	98	82	80
	Both	96	81	98	97	80	89
K' SHAP	$\mathcal{A}$	79	34	63	70	23	72
	L2E	89	**70**	63	**90**	**70**	72
	Both	80	58	100	74	51	100
D' SHAP	$\mathcal{A}$	82	68	65	81	70	59
	L2E	77	70	65	82	73	59
	Both	76	64	100	78	69	100

Table 1: Percentage agreement of the black-box model with the baseline explanation algorithm $\mathcal{A}$ and L2E; "Both" shows the agreement between L2E and $\mathcal{A}$; **bold** indicates statistical significance.

		Positive Δlog-odds $\uparrow$		
	Models	News	SST	CoLA
Random		0.57±0.11	1.93±0.17	1.92±0.1
Occ.	$\mathcal{A}$	3.69±1.2	5.04±1.55	———
	L2E	**6.82±1.06**	4.79±1.16	———
Grad.	$\mathcal{A}$	2.69±1.2	5.29±0.83	1.87±0.24
	L2E	**6.59±1.31**	6.47±0.74	1.8±0.24
LRP	$\mathcal{A}$	1.87±0.67	4.48±0.89	1.18±0.46
	L2E	2.09±0.7	3.68±0.86	0.92±0.37
LIME	$\mathcal{A}$	11.06±0.86	5.7±1.51	1.41±0.44
	L2E	11.31±0.53	5.26±0.91	1.41±0.44
K' SHAP	$\mathcal{A}$	4.33±1.21	0.22±0.37	1.96±0.28
	L2E	3.24±0.92	**2.81±0.65**	1.97±0.24
D' SHAP	$\mathcal{A}$	1.16±0.54	4.82±2.65	1.22±0.58
	L2E	2.25±0.72	7.61±1.86	1.02±0.48

Table 2: Positive Δlog-odds when employing the Sequence Labeling formulation; **bold** indicates statistical significance; the experimental results also show that L2E never performs significantly worse than $\mathcal{A}$; missing entries are due to all words being considered as positively important.

Faithfulness. For the Ranking formulation of L2E, we select the top 30% of the important words in each test sample.[5] For the Sequence Labeling formulation, we select the same number of positive/negative words identified by L2E and $\mathcal{A}$.

Table 1 shows the Prediction-based agreement between the black-box model $f_{\boldsymbol{\theta}}$ and our method L2E, between $f_{\boldsymbol{\theta}}$ and the underlying explainer $\mathcal{A}$, and between L2E and $\mathcal{A}$. We see that the explanations generated by L2E are equally predictive of the output class as those generated by $\mathcal{A}$ in both the Ranking and the Sequence Labeling formulations. We also note that the L2E version that learns with the Ranking formulation is often less faithful, though not significantly, to the black-box model $f_{\boldsymbol{\theta}}$ than $\mathcal{A}$ compared to the version that learns with the Sequence Labeling formulation.[6] For example, the percentage agreement of L2E-Ranking is lower than that of Occlusion for the three datasets, while the agreement of L2E-SequenceLabeling is higher than that of Occlusion for these datasets. Interestingly, when the baseline explanation algorithm does not perform well, e.g., Kernel SHAP on SST, L2E is still able to find words that are predictive of the output of $f_{\boldsymbol{\theta}}$. In such circumstances, the agreement between L2E and $\mathcal{A}$ is quite low ("Both" is 58% and 51% for Ranking and Sequence Labeling respectively). The low performance of Kernel SHAP may be attributed to insufficient samples (10^3 in this case) in the kernel computation for SST, while L2E could still utilize all the samples during training.

Table 2 presents the Δlog-odds results for positive explanation words in the Sequence Labeling formulation. Similar results are observed for negative explanation words in the same formulation, and top important words in the Ranking formulation. These results appear in Appendix D. They are obtained by randomly selecting 100 documents in the test set, and masking the same number of important words in each document based on the explanations generated by L2E and by $\mathcal{A}$.

We observe that some baselines have inconsistent faithfulness for different datasets. For example, LRP and Deep SHAP perform worse than Kernel SHAP for the News dataset, but better for SST. We also note that, when one baseline performs worse than the other baselines, e.g., Kernel SHAP for SST, our method L2E still performs significantly better than that baseline. This result demonstrates that our model can learn important words that yield more faithful explanations than those learned by the teacher explainer. Interestingly, none of the results for the CoLA dataset, from the baseline $\mathcal{A}$ or L2E, significantly outperforms the *Random* baseline. This flags a drawback of evaluating explanation faithfulness on short documents.

[5] We select 30% to ensure sufficient important words are selected in each dataset given their average document length. We use the same percentage in the Stability evaluation.

[6] Statistical significance ($\alpha < 0.05$) was measured by performing the Wilcoxon Signed-Rank Test (Woolson, 2007) followed by a sequential Holm-Bonferroni correction (Holm, 1979; Abdi, 2010) for all pairs of comparisons in a table.

	Models	Ranking News	 SST	 CoLA	Sequence Labeling News	 SST	 CoLA
Random		6.19±0.38	3.74±0.39	9.29±0.9	11.0±0.6	6.75±0.36	19.25±1.45
Occlusion	$\mathcal{A}$	7.18±1.35	4.58±1.19	15.59±3.88	10.82±1.34	7.04±0.87	23.67±3.76
	L2E	**8.96±1.67**	**8.52±1.48**	**21.26±3.72**	**13.94±1.49**	**8.38±0.75**	**26.47±3.84**
Gradient	$\mathcal{A}$	7.17±1.17	3.87±0.97	8.63±1.89	9.02±1.06	6.33±0.74	10.4±1.84
	L2E	**10.36±1.77**	**7.41±1.2**	**20.75±4.11**	**14.38±1.42**	**7.27±0.67**	**22.09±3.61**
LRP	$\mathcal{A}$	10.2±1.74	1.13±0.46	10.92±2.25	13.84±1.45	6.28±0.79	22.53±3.37
	L2E	11.2±1.95	**7.75±1.22**	**19.45±3.8**	14.67±1.48	7.45±0.79	26.45±3.84
LIME	$\mathcal{A}$	8.24±1.3	3.01±0.87	13.96±2.96	8.61±1.11	6.92±0.86	17.48±3.49
	L2E	**12.44±1.94**	**5.01±0.88**	16.78±3.82	**13.89±1.47**	**8.24±0.73**	**25.35±3.82**
K' SHAP	$\mathcal{A}$	7.47±1.52	2.49±0.72	19.12±3.91	5.75±1.2	1.47±0.52	22.39±4.8
	L2E	**12.84±1.97**	2.82±0.83	18.78±3.96	**9.15±1.63**	1.73±0.53	21.82±4.7
D' SHAP	$\mathcal{A}$	6.68±1.08	**2.87±0.71**	16.8±4.18	7.22±1.02	6.66±0.88	13.05±3.23
	L2E	**8.67±1.3**	1.43±0.68	20.82±3.71	**10.05±1.38**	**8.41±0.81**	**21.88±3.89**

Table 3: Intersection over Union (IoU) using semantic similarity; **bold** indicates statistical significance. Since LRP considers all words to be positively important for the prediction, we only consider the IoU of high positive words in the Labeling formulation.

Stability. For each test document, we consider the top-3 similar documents in the test set, and report the average IoU as explained in §4.4. Table 3 shows the results obtained using semantic similarity for the baseline $\mathcal{A}$ and L2E. Similar results with lexical similarity appear in Appendix C. From Table 3, we see that, in most cases, our method statistically significantly outperforms the baseline for all three datasets. For both formulations, Ranking and Sequence Labeling, L2E achieves a higher stability than the baseline $\mathcal{A}$, even in cases where $\mathcal{A}$'s IoU is comparable to that of the *Random* baseline, e.g., Gradient for SST and CoLA. These results show that learning the explanation process across different examples, as done by L2E, can capture more commonalities (higher stability) than generating explanation individually (baselines).

Overall, the LIME baseline performs consistently better than most baselines in terms of faithfulness and stability across the three datasets. Therefore, L2E also performs better when it learns from LIME than when it learns from other baselines.

Computational Efficiency. We now compare the efficiency of L2E against that of the baseline explanation algorithms $\mathcal{A}$ when generating explanations for test documents. In our experiments, the black-box is a transformer-based model comprising L layers, H attention-heads and D embedding dimensions. The complexity of this model when predicting a document of size N is then $\mathcal{O}(L \times N \times D \times (D + N + H))$ (Gu et al., 2020). Various factors contribute to the computational demands of existing explanation algorithms (details in Appendix B), and make the complexity of these algorithms grow with the size of the black-box model. These factors include the size of the input document (Occlusion), the sample size (LIME, Kernel SHAP and Deep SHAP) etc. In contrast, L2E is a distillation of any explanation algorithm, employing a smaller architecture than the black-box, e.g., fewer layers and attention heads, and lower embedding dimensions.

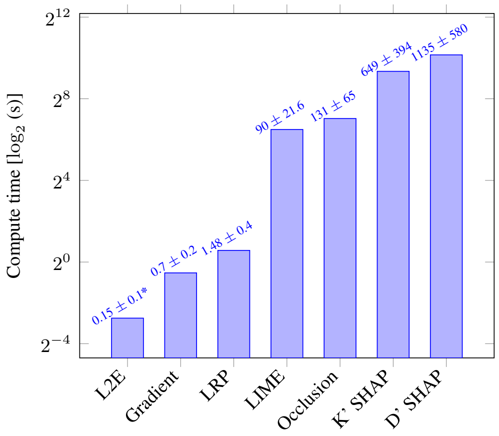


Figure 3: Average inference time for the baseline explanation algorithms and L2E-SequenceLabeling for 100 documents from IMDB-R; lower is better; y-axis is in log-scale; * indicates statistical significance.

Figure 3 shows the inference time of L2E-SequenceLabeling compared to that of the baseline explainers for the IMDB-R dataset.[7] We only show the results obtained with Sequence Labeling, since the inference time of L2E models is independent of the learning formulation. As seen in Figure 3, L2E requires statistically significantly less time than any of the six baseline explanation algorithms for IMDB-R. Similar patterns were observed for the

[7] All timing information is collected with the same hardware configuration: Intel Xeon E5-2680 v3, NVIDIA Tesla K80, 32 GB RAM.

other three datasets (Appendix E).

Finally, L2E only needs a forward pass through the explainer DNN. Comparing with Gradient and LRP, which require only one backpropagation through the black-box DNN, L2E is respectively 5 and 10 times faster for all datasets (all black-box sizes appear in §4.1 and Appendix F).

5 Evaluation with Human Rationales

Evaluation of explanation methods for DNNs is challenging, as ground-truth explanations are often unavailable. In this section, we propose to address this issue using the IMDB-R dataset (Zaidan et al., 2007), which contains movie reviews $\boldsymbol{x}$ together with their sentiment y, as well as rationales $\boldsymbol{r}$ annotated by people for the sentiment label. Our use of rationales for evaluating explanations is related to that in (Osman et al., 2020), where synthetic data are generated from apriori fixed rationales. Specifically, we generate new data by assigning a "neutral" label to an example where the human rationales are masked. We then use both the original data (without masking) and the new data to train the black-box model, where the training protocol forces the classifier to make a "neutral" prediction when the human rationales are removed from the review. More formally, we maximize the following training objective,

$$\sum_{(\boldsymbol{x},\boldsymbol{r},y)\in\mathcal{D}} \log \Pr(y = f_{\boldsymbol{\theta}}(\boldsymbol{x})) + \log \Pr(\textsc{neutral} = f_{\boldsymbol{\theta}}(\boldsymbol{x} - \boldsymbol{r}))$$

where $\boldsymbol{x} - \boldsymbol{r}$ denotes the input $\boldsymbol{x}$ with the rationale words $\boldsymbol{r}$ masked out, NEUTRAL is an extra label,[8] and $\mathcal{D}$ is the training data.

Our classifier achieves an accuracy of 83.83% on the training set, 79.68% on the validation set and 74.5% on the test set. Due to the large document sizes (Table 6 in Appendix A) and the quadratic time complexity of the Ranking formulation as a function of document size, we only train L2E with the Sequence Labeling formulation; we use lexical similarity to measure IoU, due to the time-consuming computation of semantic similarity with BERT. Details about the dataset, the classifier and the explainer's architecture appear in Appendix F.

The faithfulness and stability of the explanation methods are evaluated as follows.

		Positive Reviews			Negative Reviews		
		Pre.	**Re.**	**F1**	**Pre.**	**Re.**	**F1**
Occ.	$\mathcal{A}$	9	42	14	12	41	16
	L2E	**10**	**92**	**18**	12	**82**	**19**
Grad.	$\mathcal{A}$	9	21	12	13	26	15
	L2E	**11**	**49**	**17**	15	**47**	**18**
LRP	$\mathcal{A}$	7	5	5	12	18	14
	L2E	**12**	**12**	**11**	12	**28**	**16**
LIME	$\mathcal{A}$	**12**	39	17	**15**	42	21
	L2E	11	**45**	17	13	**49**	20
K'SHAP	$\mathcal{A}$	11	2	3	11	2	3
	L2E	10	2	2	14	2	3
D'SHAP	$\mathcal{A}$	9	22	12	12	28	16
	L2E	**11**	**50**	**17**	13	**54**	**20**

Table 4: Percentage precision, recall and F1 of explanations from L2E and corresponding baselines for dataset IMDB-R; **bold** indicates statistical significance. Detailed precision and recall values of positive reviews appear in Appendix G.

Faithfulness. We select the top-K important words generated by an explanation method and compute the precision, recall and F1 against the human-annotated rationales. It is worth noting that our L2E explainer is not supervised by human rationales directly. Instead, we use the same experimental setup as in Section 4.5 to ensure the L2E explainer is learning from the baseline algorithms rather than the human rationales.

Table 4 displays the average values over all test instances. As noted by Carton et al. (2020), the rationales in the original dataset are not exhaustively identified by human annotators. For a particular event, we expect to observe a lower precision than recall, since the black-box model might still be able to utilize the words not being annotated in addition to the words annotated by a human. The results in Table 4 align with this hypothesis. For instance, besides LRP for the positive reviews and Kernel SHAP for both reviews, all baselines and the corresponding L2E have higher recall than precision. Furthermore, L2E outperforms the corresponding baseline $\mathcal{A}$ significantly in most cases for both positive and negative reviews, except when comparing with LIME's precision. This observation indicates that learning the explanations of multiple examples together, as done by L2E, achieves high faithfulness to human rationales, as well as to the black-box model.

Stability. Table 5 displays stability computed in three ways: (1) no filtering (which extracts important words only, Table 3), (2) filtering non-annotated words, and (3) filtering stop-words. For the two filtering measures, prior to filtering, we

[8] It simulates abstaining from predicting any label from the original label set.

		no filter	filter non-annotated words	filter stop-words
Human		5.83±0.27	5.83±0.27	3.06 ±0.27
Occ.	$\mathcal{A}$	4.53±0.38	5.8±0.28	2.46±0.19
	L2E	**4.57±0.38**	**5.86±0.27**	**2.48±0.19**
Grad.	$\mathcal{A}$	4.06±0.25	3.64±0.31	1.65±0.15
	L2E	**4.41±0.33**	**4.7±0.34**	**2.33±0.19**
LRP	$\mathcal{A}$	1.89±0.15	1.08±0.27	1.7±0.13
	L2E	**4.01±0.24**	**5.66±0.43**	**2.45±0.17**
LIME	$\mathcal{A}$	1.36±0.08	1.05±0.16	0.93±0.07
	L2E	**2.19±0.17**	**1.93±0.25**	**1.97±0.15**
K'SHAP	$\mathcal{A}$	0.03±0.04	0.06±0.09	0.03±0.04
	L2E	**0.45±0.17**	0.69±0.59	0.45±0.17
D'SHAP	$\mathcal{A}$	4.29±0.18	3.33±0.29	1.61±0.13
	L2E	4.39±0.25	**4.02±0.3**	**2.37±0.17**

Table 5: Intersection over Union (IoU) using lexical similarity for the IMDB-R test set; **bold** indicates statistical significance.

ensure the same number of important words is selected from the explanation produced by baseline $\mathcal{A}$ and L2E. Equation 1 is then used to compute the IoU value. To ensure a fair comparison, we select the same number of words in L2E and a comparable baseline $\mathcal{A}$ before filtering.

Similarly to the results in §4.5, as seen in Table 5, L2E yields more stable explanations than the corresponding baselines. The best stability, obtained with L2E (58.6 ± 0.27) by filtering non-annotated words when learning from Occlusion, is comparable to that of the human rationales. This is due to the high recall (92 and 82 for positive and negative reviews respectively in Table 4) in the explanations produced by L2E, which indicates they have high overlap with human rationales. Further, when measuring the IoU values, the L2E explanations of similar examples have the same intersection with the human rationales, but a lower union. This result indicates that people favour stable rationales in similar documents, and reinforces our findings regarding the greater consistency of the explanations produced by L2E compared to the baselines.

LRP has been proven to have explanation continuity (Montavon et al., 2018), where the explanations of two nearly equivalent instances are also equivalent. However, we do not observe such a pattern in our experiments. We hypothesize that using perturbed instances as neighbours, as done by Montavon et al. (2018), does not necessarily follow the same distribution of the data. Instead, we posit that finding similar examples within a dataset, as done in our experiments, is a better proxy for stability evaluation.

6 Conclusions and Future Work

We have presented a Learning to Explain (L2E) approach to learn the commonalities of the explanation generation processes across different examples. We have further proposed Ranking and Sequence Labeling formulations to effectively learn the explainer model by discretizing feature weights produced by existing explanation algorithms.

Our experimental results show that our method can generate more stable explanations (i.e., not vary much across similar documents) than those generated by the explainer baselines, while maintaining the same level of faithfulness to the underlying black-box model as the baseline algorithms. Moreover, our L2E approach produces explanations between 5 and 7.5 × 10^4 times faster than the six baselines, making it suitable for long documents and very large black-box models.

Our L2E approach trains an explainer, a black-box, to mimic the behaviour of an explanation method for an existing black-box model. A key challenge lies in the variation in the convergence status of such an explainer for different initializations. In order to mitigate this problem, we evaluate the performance of our explainer by averaging three different initializations.

The L2E approach opens up the possibility of distilling multiple explanation algorithms into one model. Although we focused on the stability, faithfulness and efficiency aspects of explanation generation, there are further desirable properties, e.g., transparency, comprehensibility and novelty (Robnik-Šikonja and Bohanec, 2018). Devising model-based explanation methods and their evaluation with these desiderata are interesting directions for future research.

Acknowledgements

This research was supported in part by grant DP190100006 from the Australian Research Council. The first author was partly supported by CSIRO/Data61. The computational resources for this work were supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) (www.massive.org.au). We would like to thank the anonymous reviewers for their insightful comments.

References

Hervé Abdi. 2010. Holm's sequential bonferroni procedure. *Encyclopedia of research design*, 1(8):1–8.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515.

David Alvarez-Melis, Hal Daumé III, Jennifer Wortman Vaughan, and Hanna Wallach. 2019. Weight of evidence as a basis for human-oriented explanations. In *NeurIPS Workshop on HCML*.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A Diagnostic Study of Explainability Techniques for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274.

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

Seojin Bang, Pengtao Xie, Heewook Lee, Wei Wu, and Eric Xing. 2019. Explaining a black-box using deep variational information bottleneck approach.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems 31*, pages 9539–9549.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and characterizing human rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9294–9307.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1721–1730.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 883–892.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2019. L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458.

G Erion, JD Janizek, P Sturmfels, S Lundberg, and SI Lee. 2019. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *arXiv preprint arXiv:1906.10670*.

Xiaotao Gu, Liyuan Liu, Hongkun Yu, Jing Li, Chen Chen, and Jiawei Han. 2020. On the transformer growth for progressive bert training. *arXiv preprint arXiv:2010.12562*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

S. Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. 2020. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. 2018. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578.

Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. 2016. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.

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

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

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

David Martens and Foster Provost. 2014. Explaining data-driven document classifications. *Mis Quarterly*, 38(1):73–100.

Tim Miller. 2018. Contrastive explanation: A structural-model approach. *arXiv preprint arXiv:1811.03163*.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. 2014. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Ahmed Osman, Leila Arras, and Wojciech Samek. 2020. Towards ground truth evaluation of visual explanations. ArXiv:2003.07258.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, pages 48–53.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of EMNLP*, pages 1938–1952.

Jorge Pérez, Javier Marinković, and Pablo Barceló. 2019. On the turing completeness of modern neural network architectures. In *International Conference on Learning Representations*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI*, volume 18, pages 1527–1535.

Michael L Rich. 2016. Machine learning, automated suspicion algorithms, and the fourth amendment. *University of Pennsylvania Law Review*, pages 871–929.

Marko Robnik-Šikonja and Marko Bohanec. 2018. Perturbation-based explanations of prediction models. In *Human and machine learning*, pages 159–175. Springer.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS Workshop on EMC^2*.

Patrick Schwab and Walter Karlen. 2019. Cxplain: Causal explanations for model interpretation under uncertainty. In *Advances in Neural Information Processing Systems*, pages 10220–10230.

Cansu Sen, Thomas Hartvigsen, Biao Yin, Xiangnan Kong, and Elke Rundensteiner. 2020. Human attention maps for text classification: Do humans and neural networks focus on the same words? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4596–4608.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

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

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of Machine Learning Research*, volume 70, International Convention Centre, Sydney, Australia. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Dylan Slack, Sophie Hilgard, Sameer Singh, and Himabindu Lakkaraju. 2020. How much should i trust you? modeling uncertainty of black box explanations. *arXiv preprint arXiv:2008.05030*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 3319–3328. JMLR.org.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of NLP models is manipulable. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

RF Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3.

Zhengxuan Wu and Desmond C Ong. 2021. On explaining your explanations of bert: An empirical study with sequence classification. *arXiv preprint arXiv:2101.00196*.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Appendix A Datasets Statistics

Datasets	Train/Dev/Test	Avg. len
News	108000/12000/7600	38
SST	6920/872/1821	17
CoLA	8551/527/516	8
IMDB-R	2864/320/200	666

Table 6: Dataset statistics used in the experiments.

Appendix B Baseline Explanation Methods ($\mathcal{A}$)

In this section, we describe our experimental setups for the six baselines.

- **Occlusion** (Zeiler and Fergus, 2014; Schwab and Karlen, 2019). The occlusion method converts $\boldsymbol{x}$ into $\tilde{\boldsymbol{x}}$ by masking token x_i with a predefined token. The weight of x_i is then determined by the difference of the output or loss from $f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$ and $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. In our experiments, we use the mask token from the corresponding black-box tokenizer, and measure the feature weight based on the changes between the loss functions before and after the masking. The time complexity for this baseline is $\mathcal{O}(|\boldsymbol{x}|)$ at test time.
- **Gradient** (Simonyan et al., 2013). The weight of token x_i is given by the accumulated gradients of the highest probable prediction with regards to each dimension of the token in the embedding layer. We also multiply the corresponding embedding value before accumulation (Kindermans et al., 2016). Gathering the weights of all features in $\boldsymbol{x}$ requires one pass of backward propagation, and the time complexity for this baseline is dependent on the size of black-box model at test time.
- **LRP** (Bach et al., 2015). Layer-wise Relevance Propagation decomposes a model's outcome from the output layer into relevance scores of neurons in each intermediate layer until it reaches features in the input layer. The rule of decomposition is subject to the type of kernel and connectivity between neurons in adjacent layers, such as linear or attention layers. We follow the implementation by Wu and Ong (2021) to measure relevance score in variations of BERT model (Devlin et al., 2018).
- **LIME** (Ribeiro et al., 2016). LIME samples the neighbors of $\boldsymbol{x}$ by perturbing different x_i, and uses these samples to learn a linear separator which approximates the local behavior of the black-box $f_{\boldsymbol{\theta}}$. The weight of each x_i is then given by the coefficients of the separator.
- **Kernel SHAP** (Lundberg and Lee, 2017). The Shapley value (Shapley, 1953) is a concept from cooperative game theory which calculates the weight of feature x_i by considering its interaction with all the other subsets of features. Kernel SHAP approximates the Shapley value by weighted sampling (kernel). The kernel is determined by the number of permutations of features. According to Lundberg and Lee (2017), LIME and Kernel SHAP only differ in the choice of kernel. We use cosine similarity in LIME and the size of subset permutations in Kernel SHAP for the kernel computation.
- **Deep SHAP** (Shrikumar et al., 2017). This is another method to approximate the Shapley value. It computes the weight of x_i as the effect on the output when x_i is set to a reference value. Such an effect is achieved by linearizing the black-box model through back-propagation. Hence, the complexity of Deep SHAP is dependent on both the size of reference samples and the black-box, which makes it the most computationally expensive method among all our baselines. In our experiments, we use the API provided by Lundberg and Lee (2017) and set the reference value of x_i to the corresponding value in each of the randomly selected samples. A sample size of 500/1000/1000/1000 respectively for datasets IMDB-R, AGNews, SST and CoLA is used in the baselines requiring sampling – LIME, Kernel SHAP and Deep SHAP.

Appendix C Document Similarity for Intersection over Union

The first approach for computing IoU uses semantic similarity between two documents. This is measured by summing the token representations along hidden dimensions from a pre-trained BERT base model with uncased English, open-sourced by Hugging Face (Wolf et al., 2019).

Our second approach is to compute the intersection over union for overlapping n-grams between two documents, referred to as lexical similarity. In our experiment, we sum this value up to 4-grams in two documents as the score of similarity. The results from this approach are reported in Table 7.

		Ranking			Sequence Labeling		
	Models	**News**	**SST**	**CoLA**	**News**	**SST**	**CoLA**
Random		7.36±0.34	6.34±0.54	11.94±0.92	13.11±0.57	13.1±0.51	23.35±1.45
Occlusion	$\mathcal{A}$	7.5±1.23	8.15±1.35	16.83±3.37	12.36±1.26	13.5±1.19	28.78±3.51
	L2E	**10.51±1.63**	**14.11±1.86**	**23.37±3.71**	**16.8±1.26**	**15.56±0.89**	**31.78±3.56**
Gradient	$\mathcal{A}$	7.05±1.11	5.14±1.05	11.95±2.33	10.32±0.96	11.5±0.87	13.36±1.95
	L2E	**11.92±1.79**	**13.8±1.7**	**22.52±3.81**	**17.35±1.16**	**13.77±0.8**	**25.77±3.38**
LRP	$\mathcal{A}$	10.34±1.64	2.67±0.58	12.37±2.27	16.33±1.23	12.76±0.84	26.69±3.16
	L2E	11.87±1.92	**12.52±1.61**	**21.87±3.85**	17.6±1.21	14.84±0.73	31.8±3.55
LIME	$\mathcal{A}$	8.65±1.22	6.42±1.17	16.8±3.19	9.83±0.97	13.92±1.13	20.56±3.38
	L2E	**13.89±2.02**	**11.23±1.61**	18.9±3.85	**16.71±1.25**	**16.24±0.74**	**30.48±3.53**
K' SHAP	$\mathcal{A}$	8.85±1.44	5.74±1.31	22.17±4.02	6.96±1.13	4.8±1.0	25.9±4.78
	L2E	**14.59±1.77**	**8.79±1.52**	21.92±4.03	**10.79±1.39**	5.67±1.01	25.12±4.69
D' SHAP	$\mathcal{A}$	7.2±1.05	5.31±1.13	19.74±3.91	8.35±0.89	11.88±1.04	14.38±3.38
	L2E	**10.98±1.27**	4.93±1.14	23.77±3.62	**12.53±1.16**	**16.24±0.71**	**25.38±3.65**

Table 7: Intersection over union (IoU) using lexical similarity (measured according to overlapping n-grams); **bold** indicates statistical significance.

Appendix D Faithfulness

We present the negative explanation words of the Sequence Labeling formulation in Table 8 and the top important words of Ranking formulation in Table 9.

		Negative Δlog-odds ↓		
	Models	**News**	**SST**	**CoLA**
Random		0.57±0.11	1.93±0.17	1.92±0.1
Occ.	$\mathcal{A}$	-0.69±0.36	1.47±1.1	———
	L2E	-0.39±0.4	1.57±1.15	———
Grad.	$\mathcal{A}$	0.12±0.21	0.99±0.55	1.53±0.36
	L2E	0.05±0.29	0.87±0.41	1.48±0.33
LIME	$\mathcal{A}$	-0.4±0.26	1.51±1.24	1.05±0.71
	L2E	-0.41±0.27	1.03±0.95	1.04±0.76
K' SHAP	$\mathcal{A}$	1.57±0.7	3.78±0.91	———
	L2E	0.46±0.33	**1.8±0.76**	———
D' SHAP	$\mathcal{A}$	0.83±0.44	0.37±0.52	1.67±0.29
	L2E	1.19±0.6	0.5±0.78	1.93±0.22

Table 8: Negative Δlog-odds when employing the Sequence Labeling formulation; **bold** indicates statistical significance; the experimental results show that L2E never performs significantly worse than $\mathcal{A}$; LRP is not included because all words were considered to be positively important.

Methods	**Models**	**News**	**SST**	**CoLA**
Random		0.57±0.11	1.93±0.17	1.92±0.1
Occ.	$\mathcal{A}$	3.03±0.79	**4.84±0.86**	1.87±0.23
	L2E	1.78±0.67	3.08±0.75	1.9±0.23
Grad.	$\mathcal{A}$	1.13±0.56	2.17±0.66	1.95±0.22
	L2E	0.91±0.54	1.74±0.56	1.88±0.23
LRP	$\mathcal{A}$	2.89±0.78	**4.41±0.86**	1.93±0.22
	L2E	2.43±0.65	2.64±0.68	1.87±0.23
LIME	$\mathcal{A}$	**6.96±1.02**	**5.01±0.89**	1.86±0.22
	L2E	5.03±0.95	2.93±0.62	1.89±0.23
K' SHAP	$\mathcal{A}$	**4.51±1.19**	0.2±0.32	1.97±0.23
	L2E	1.8±0.77	**2.25±0.67**	2.03±0.23
D' SHAP	$\mathcal{A}$	0.22±0.29	2.74±0.68	1.92±0.23
	L2E	0.57±0.33	2.98±0.77	1.85±0.23

Table 9: Δlog-odds when employing the Ranking formulation; **bold** indicates statistical significance.

Appendix E Computational efficiency

Figures 4, 5 and 6 show that L2E is more efficient than all baselines for AGNews, SST and CoLA datasets.

Appendix F IMDB dataset with human-annotated rationales

There are 900 positive and 900 negative movie reviews with rationales annotated by human in the original dataset from Zaidan et al. (2007). We randomly assign 160 and 200 examples to the validation and test set respectively, with each set having an even distribution of positive and negative reviews. We also remove 8 very long documents from the training set for the sake of CUDA memory. For each example in the training and validation sets, we construct a new example by masking the rationales, i.e., we replace each words in the rationale with a mask token, and assign this new example to a third label, e.g., neutral, so as to ensure the classifier 'pays attention' to the rationale. The final dataset split appears in Table 6.

The classifier is trained by fine-tuning the last layer of a pre-trained Longformer (Beltagy et al., 2020) with 12 layers and 12 attention heads from Hugging Face (Wolf et al., 2019). It achieves 83.83%/79.68%/74.5% accuracy for the training/validation/test sets respectively after 40 epochs. The statistics of our experiment are measured on test examples that are predicted correctly by the classifier. For each L2E explainer that learns from a baseline explanation method, we use a Longformer with 4 layers, 4 attention heads.

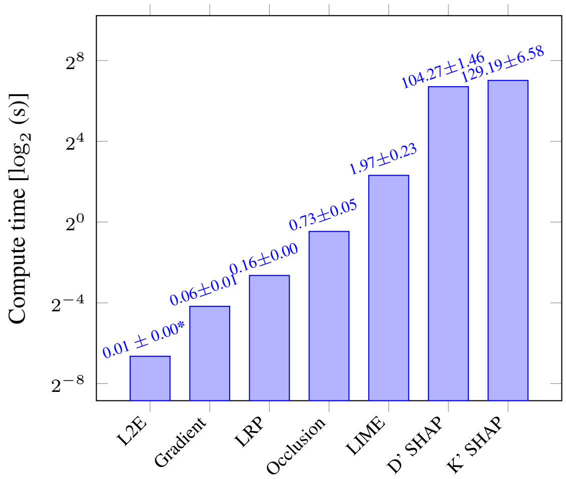


Figure 4: Average inference time for the six baseline explanation algorithms and ours (L2E) for the same 100 documents on the News dataset; lower is better; * indicates statistical significance.

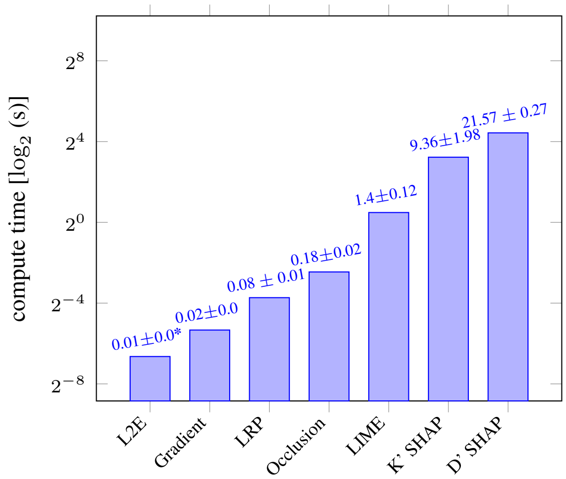


Figure 5: Average inference time for the six baseline explanation algorithms and ours (L2E) for the same 100 documents on the SST dataset; lower is better; * indicates statistical significance.

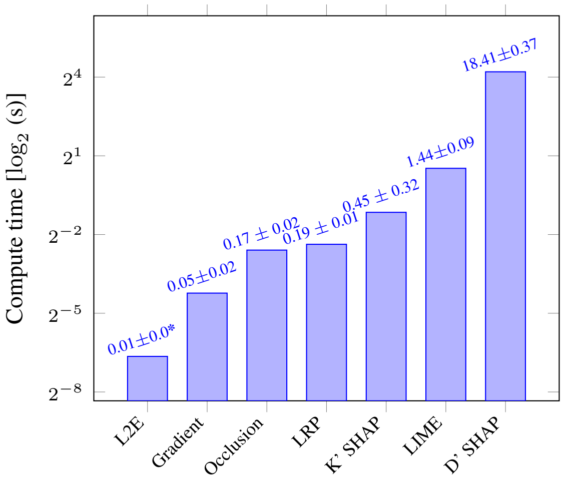


Figure 6: Average inference time for the six baseline explanation algorithms and ours (L2E) for the same 100 documents on the CoLA dataset; lower is better; * indicates statistical significance.

Appendix G Precision and Recall on Positive Reviews

We plot the precision versus recall from all the L2E-$\mathcal{A}$ pairs in dataset IMDB-R in Figure 7. The results show that, in most case, L2E performs better than $\mathcal{A}$ in terms of faithfulness to the underlying black-box and alignment with the human rationales.

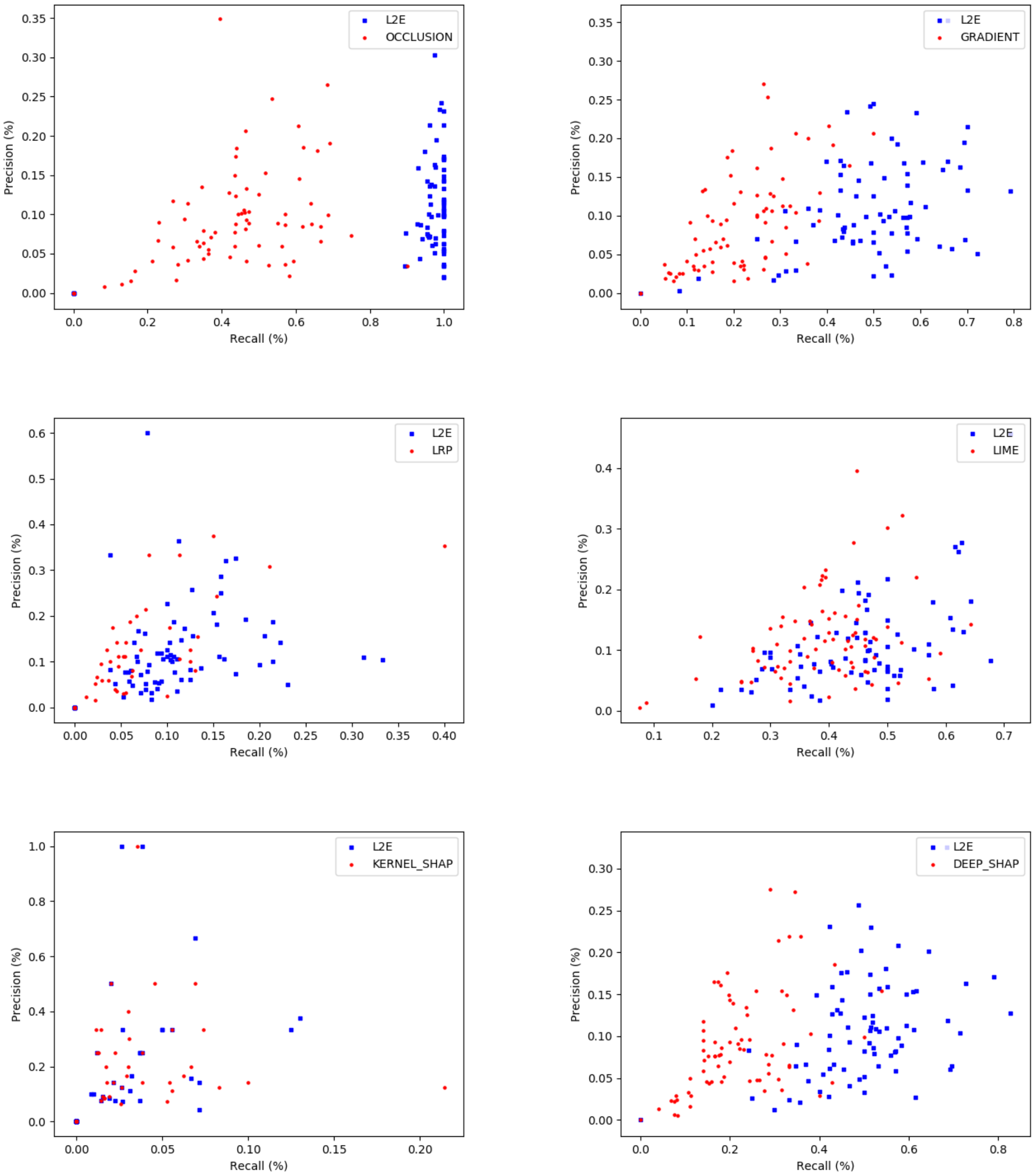


Figure 7: Precision and recall of L2E versus each of the six baselines for all correctly predicted positive reviews from IMDB-R test.