Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?

Peter Hase and Mohit Bansal

Dominika Basaj, 18/05/2020

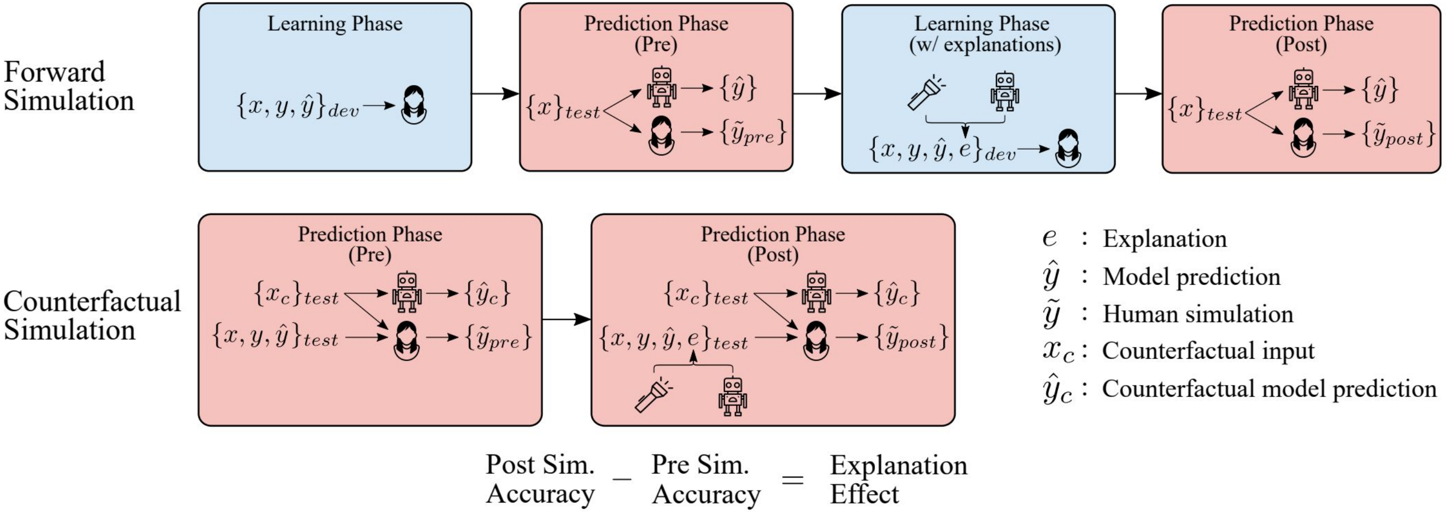
Abstract

- Simulatability - the degree to which something can be *simulated*
- A model is simulatable when a person can predict its behavior on new inputs
- The aim of this paper is to isolate the effect of explanations on simulatbility
- 5 XAI methods: Lime, Anchor, Decision Boundary, a Prototype model, a Composite Approach on text and tabular data

Testing for simulatability

- Doshi-Velez and Kim (2017) describe two human-subject tasks that test for simulatability
- Task 1: **forward simulation**: given:
    - input
    - explanation
    - users must predict what a model **would output for the given input**
- Task 2: **counterfactual simulation**: given
    - Input
    - a model’s output for that input
    - explanation
    - users must predict what the model **will output when given a perturbation of the original input**

Simulation design



Forward Simulation
Learning Phase
$\{x, y, \hat{y}\}_{dev}$
Prediction Phase (Pre)
$\{x\}_{test}$
$\{\hat{y}\}$
$\{\tilde{y}_{pre}\}$
Learning Phase (w/ explanations)
$\{x, y, \hat{y}, e\}_{dev}$
Prediction Phase (Post)
$\{x\}_{test}$
$\{\hat{y}\}$
$\{\tilde{y}_{post}\}$
Counterfactual Simulation
Prediction Phase (Pre)
$\{x_c\}_{test}$
$\{\hat{y}_c\}$
$\{x, y, \hat{y}\}_{test}$
$\{\tilde{y}_{pre}\}$
Prediction Phase (Post)
$\{x_c\}_{test}$
$\{\hat{y}_c\}$
$\{x, y, \hat{y}, e\}_{test}$
$\{\tilde{y}_{post}\}$
e : Explanation
$\hat{y}$: Model prediction
$\tilde{y}$: Human simulation
x_c : Counterfactual input
$\hat{y}_c$: Counterfactual model prediction
Post Sim. Accuracy − Pre Sim. Accuracy = Explanation Effect

- 2100 responses
- The effect of explanations are isolated by first measuring baseline accuracy, then measuring accuracy after users are given access to explanations of model behavior

Interpretability vs explainability

Explanation methods may improve the interpretability of a model, in the sense that **an interpretable model is simulatable**.

XAI methods

Feature Importance Estimation

- Provide information about how the model uses certain features
- LIME, Anchor (Ribeiro)

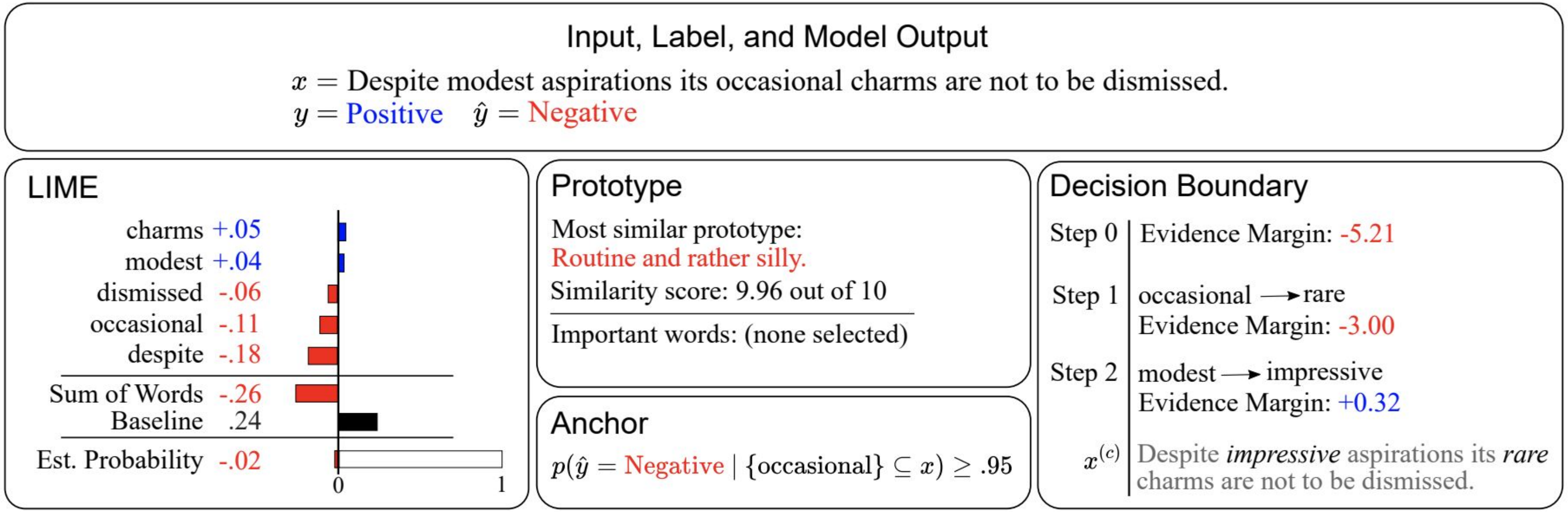
Case-based reasoning

- Prototype models classify new instances based on their similarity to other known cases
- These prototypes are used to produce classifier features that are intended to be directly interpretable

Latent space traversal

- These methods traverse the latent space of a model in order to show how the model behaves as its input changes.
- In a classification setting, crossing the decision boundary may reveal necessary conditions for a model's prediction for the original input.

XAI methods



Input, Label, and Model Output

x = Despite modest aspirations its occasional charms are not to be dismissed.

y = Positive $\hat{y}$ = Negative

LIME

charms +.05

modest +.04

dismissed -.06

occasional -.11

despite -.18

Sum of Words -.26

Baseline .24

Est. Probability -.02

Prototype

Most similar prototype:

Routine and rather silly.

Similarity score: 9.96 out of 10

Important words: (none selected)

Anchor

$p(\hat{y} = \text{Negative} \mid \{\text{occasional}\} \subseteq x) \geq .95$

Decision Boundary

Step 0 Evidence Margin: -5.21

Step 1 occasional → rare

Evidence Margin: -3.00

Step 2 modest → impressive

Evidence Margin: +0.32

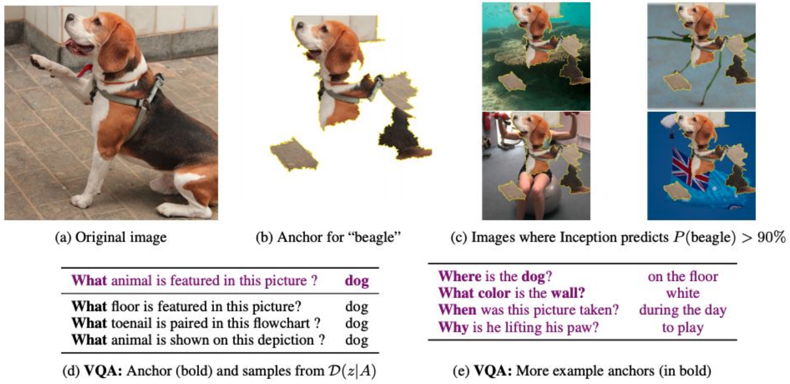
$x^{(c)}$ Despite *impressive* aspirations its *rare* charms are not to be dismissed.

Figure 2: Explanation methods applied to an input from the test set of movie reviews.

Feature importance

- LIME: Local Interpretable Model-Agnostic Explanations
- Anchors: High-Precision Model-Agnostic Explanations

- Let A be a rule (set of predicates) acting on such an interpretable representation, such that A(x) returns 1 if all its feature predicates are true for instance x.
- A is an anchor if A(x) = 1 and A is a sufficient condition for f(x) with high probability — in our running example, if a sample z from D(z|A) is likely predicted as Positive (i.e. f(x) = f(z)). Formally A is an anchor if



(a) Original image

(b) Anchor for "beagle"

(c) Images where Inception predicts $P(\text{beagle}) > 90\%$

What animal is featured in this picture ?	**dog**
What floor is featured in this picture?	dog
What toenail is paired in this flowchart ?	dog
What animal is shown on this depiction ?	dog

(d) **VQA:** Anchor (bold) and samples from $\mathcal{D}(z|A)$

Where is the **dog**?	on the floor
What color is the **wall?**	white
When was this picture taken?	during the day
Why is he lifting his paw?	to play

(e) **VQA:** More example anchors (in bold)

Figure 3: Anchor Explanations for Image Classification and Visual Question Answering (VQA)

Prototype model

- We develop a prototype model for use with text and tabular classification tasks. In our model, a neural network *g* maps inputs to a latent space
- The score of class c is

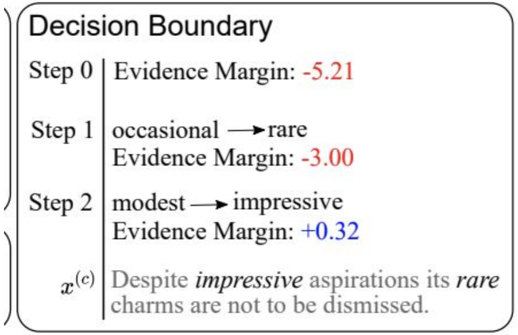
$$f(\mathbf{x}_i)_c = \max_{\mathbf{p_k} \in P_c} a(g(\mathbf{x}_i), \mathbf{p_k})$$

Where a is a similarity function (gaussian kernel) and Pc is a set of prototype vectors for class c.

- The model predicts inputs to belong to the same class as the prototype they’re closest to in the latent space.

Decision boundary

- First samples around the original input to get instances that cross the decision boundary.
- A counterfactual input is chosen from these by taking the instance with the fewest edited features (tokens or variables)
- Lastly, we provide a path between inputs by greedily picking the edit from the remaining edits that least changes the model's evidence margin, which is the difference between positive and negative class scores



Decision Boundary

Step 0	Evidence Margin: -5.21
Step 1	occasional → rare Evidence Margin: -3.00
Step 2	modest → impressive Evidence Margin: +0.32
$x^{(c)}$	Despite *impressive* aspirations its *rare* charms are not to be dismissed.

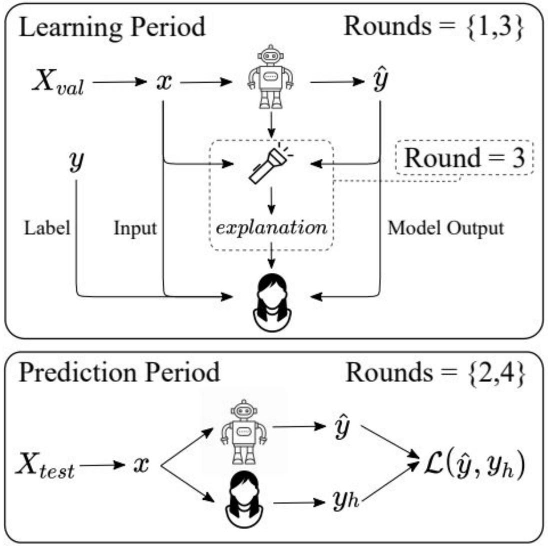
Composite method

- Composite method that combines LIME and Anchor with decision boundary and prototype explanations

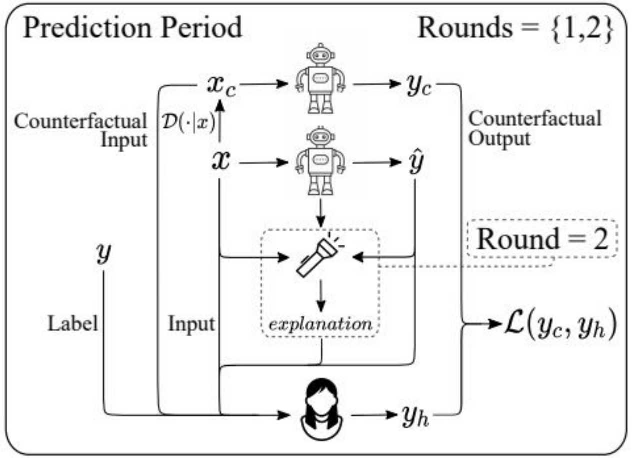
Experiments

- Experiments are conducted for tabular and text data
- Text model: movie review excerpts
  - One sentence
  - Binary sentiment labels
- Tabular model: Adult data from UCI ML repository
  - Records of 15,682 individuals
  - The label is whether their annual income is more than $50,000

Experiments

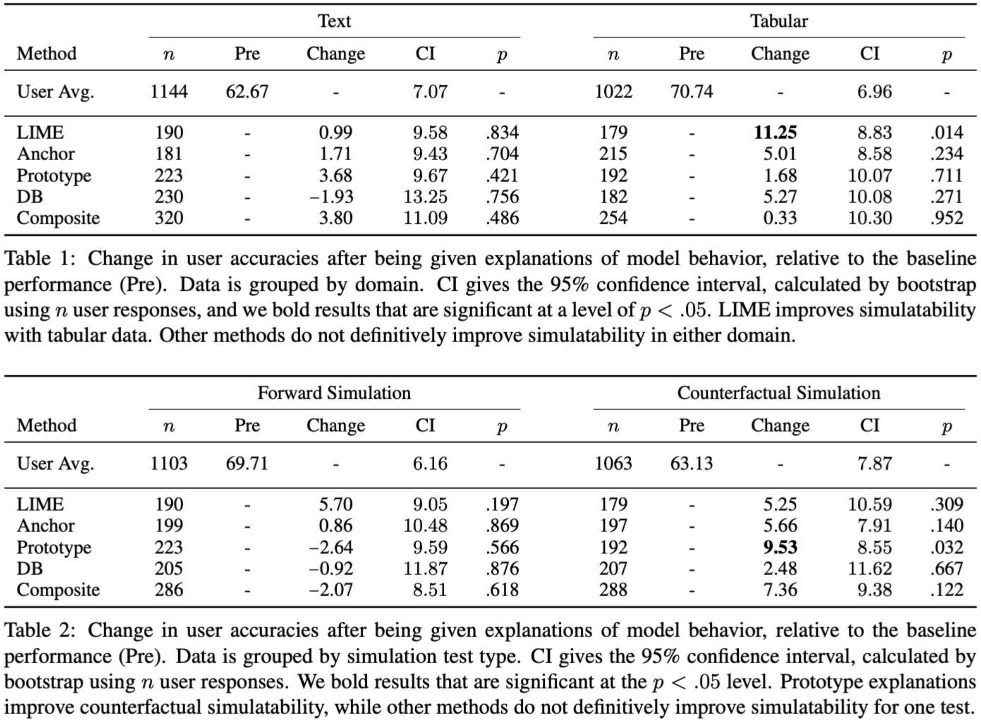


Forward simulation test procedure. We measure human users' ability to predict model behavior. We isolate the effect of explanations by first measuring baseline performance after users are shown examples of model behavior (Rounds 1, 2), and then measuring performance after they are shown explained examples of model behavior (Rounds 3, 4).



Counterfactual simulation test procedure. Users see model behavior for an input, then they predict model behavior on an edited version of the input. We isolate the effect of explanations by measuring user accuracy with and without explanations.

Results: evaluation of XAI methods



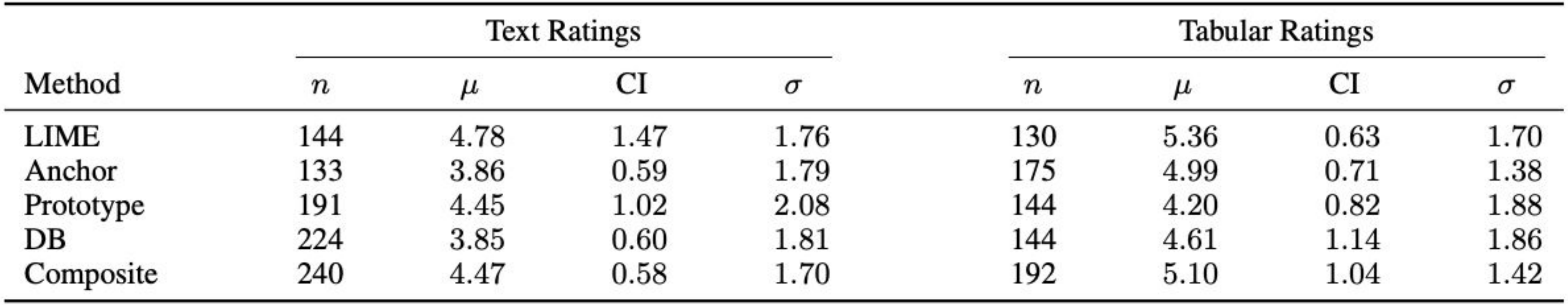
	Text					Tabular				
Method	n	Pre	Change	CI	p	n	Pre	Change	CI	p
User Avg.	1144	62.67	-	7.07	-	1022	70.74	-	6.96	-
LIME	190	-	0.99	9.58	.834	179	-	**11.25**	8.83	.014
Anchor	181	-	1.71	9.43	.704	215	-	5.01	8.58	.234
Prototype	223	-	3.68	9.67	.421	192	-	1.68	10.07	.711
DB	230	-	−1.93	13.25	.756	182	-	5.27	10.08	.271
Composite	320	-	3.80	11.09	.486	254	-	0.33	10.30	.952

Table 1: Change in user accuracies after being given explanations of model behavior, relative to the baseline performance (Pre). Data is grouped by domain. CI gives the 95% confidence interval, calculated by bootstrap using n user responses, and we bold results that are significant at a level of $p < .05$. LIME improves simulatability with tabular data. Other methods do not definitively improve simulatability in either domain.

	Forward Simulation					Counterfactual Simulation				
Method	n	Pre	Change	CI	p	n	Pre	Change	CI	p
User Avg.	1103	69.71	-	6.16	-	1063	63.13	-	7.87	-
LIME	190	-	5.70	9.05	.197	179	-	5.25	10.59	.309
Anchor	199	-	0.86	10.48	.869	197	-	5.66	7.91	.140
Prototype	223	-	−2.64	9.59	.566	192	-	**9.53**	8.55	.032
DB	205	-	−0.92	11.87	.876	207	-	2.48	11.62	.667
Composite	286	-	−2.07	8.51	.618	288	-	7.36	9.38	.122

Table 2: Change in user accuracies after being given explanations of model behavior, relative to the baseline performance (Pre). Data is grouped by simulation test type. CI gives the 95% confidence interval, calculated by bootstrap using n user responses. We bold results that are significant at the $p < .05$ level. Prototype explanations improve counterfactual simulatability, while other methods do not definitively improve simulatability for one test.

Simultability Ratings



	Text Ratings				Tabular Ratings			
Method	n	μ	CI	σ	n	μ	CI	σ
LIME	144	4.78	1.47	1.76	130	5.36	0.63	1.70
Anchor	133	3.86	0.59	1.79	175	4.99	0.71	1.38
Prototype	191	4.45	1.02	2.08	144	4.20	0.82	1.88
DB	224	3.85	0.60	1.81	144	4.61	1.14	1.86
Composite	240	4.47	0.58	1.70	192	5.10	1.04	1.42

User simulatability ratings by data domain, on a scale of 1 to 7. The mean and standard deviation for ratings are given by µ and σ. The 95% confidence interval for the mean is given by CI, as calculated by bootstrap.

Can users predict explanation effectiveness?

- Measuring **how explanation ratings relate to user correctness** in the Post phase of the counterfactual simulation test.
- In this phase, **users rate explanations of model predictions** for an original input and predict model behavior for a perturbation of that input.
- If ratings of explanation quality are a good indicator of their effectiveness, we would expect to see that higher ratings are associated with user correctness
- **We do not find evidence that explanation ratings are predictive of user correctness.**