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

Peter Hase and Mohit Bansal
peter@cs.unc.edu, mbansal@cs.unc.edu

ACL 2020
Talk Outline

- Motivation
- Proposal
  - Metric
  - Experimental Design
- Explanation Methods
- Results
- Qualitative Analysis
- Concluding Thoughts
- Follow-up Work
Motivation

- We have explanations of model behavior
  - e.g., feature importance estimates
Input, Label, and Model Output

$x = \text{Despite modest aspirations its occasional charms are not to be dismissed.}$

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

(Ribeiro et al., 2016)
Motivation

- We have explanations of model behavior
  - e.g., feature importance estimates
- We want to precisely measure explanation quality
Motivation

● We have explanations of model behavior
  ○ e.g., feature importance estimates

● We want to precisely measure explanation quality

● Quality can mean many things
  ○ Building user trust
  ○ Identifying influence of certain features
  ○ Checking behavior on particular kinds of inputs
  ○ Ensuring models are fair and unbiased
Motivation

- We have explanations of model behavior
  - e.g., feature importance estimates
- We want to precisely measure explanation quality
- We use an operational definition of *simulatability* (Doshi-Velez and Kim, 2017)
  - A model is simulatable when users can predict its outputs
Motivation

- We have explanations of model behavior
  - e.g., feature importance estimates
- We want to precisely measure explanation quality
- We use an operational definition of *simulatability* (Doshi-Velez and Kim, 2017)
  - A model is simulatable when users can predict its outputs
  - Explanations communicate one person’s mental model to another
  - Simulatability could be useful for deployment decisions, model debugging, model design
Proposal: Metric

- Measure the effect of an explanation method on model simulatability
Proposal: Metric

- Measure the effect of an explanation method on model simulatability
  - Compute user accuracy before and after seeing explanations

\[
\text{Post Sim. Accuracy} - \text{Pre Sim. Accuracy} = \text{Explanation Effect}
\]
Proposal: Experimental Design

● Measure the effect of an explanation method on model simulatability
● Important controls:
Proposal: Experimental Design

● Measure the effect of an explanation method on model simulatability

● Important controls:
  ○ Separate explained instances from test instances
Proposal: Experimental Design

- Measure the effect of an explanation method on model simulatability
- Important controls:
  - Separate explained instances from test instances
  - Evaluate the effect of explanations against a baseline of unexplained examples
Proposal: Experimental Design

- Measure the effect of an explanation method on model simulatability
- Important controls:
  - Separate explained instances from test instances
  - Evaluate the effect of explanations against a baseline of unexplained examples
  - Balance data by model correctness and model output
Proposal: Experimental Design

- Measure the effect of an explanation method on model simulatability

- Important controls:
  - Separate explained instances from test instances
  - Evaluate the effect of explanations against a baseline of unexplained examples
  - Balance data by model correctness and model output
  - Force user predictions on all inputs (or penalize abstention)
Proposal: Experimental Design

- Test 1: forward simulation

\[ \{x, y, \hat{y}\}_{dev} \rightarrow \{x\}_{test} \rightarrow \{\hat{y}\} \rightarrow \{\tilde{y}_{pre}\} \rightarrow \{x, y, \hat{y}, e\}_{dev} \rightarrow \{\tilde{y}\} \rightarrow \{\tilde{y}_{post}\} \]

- \( e \): Explanation
- \( \hat{y} \): Model prediction
- \( \tilde{y} \): Human simulation
Proposal: Experimental Design

- Test 2: counterfactual simulation

\[ e \quad : \quad \text{Explanation} \]
\[ \hat{y} \quad : \quad \text{Model prediction} \]
\[ \tilde{y} \quad : \quad \text{Human simulation} \]
\[ x_c \quad : \quad \text{Counterfactual input} \]
\[ \hat{y}_c \quad : \quad \text{Counterfactual model prediction} \]
Explanation Methods

- Feature importance estimates
  - LIME: local linear approximation (Ribeiro et al., 2016)
  - Anchors: if-then probabilistic statements (Ribeiro et al., 2018)
Explanation Methods

- Feature importance estimates
  - LIME: local linear approximation (Ribeiro et al., 2016)
  - Anchors: if-then probabilistic statements (Ribeiro et al., 2018)

- Case-based reasoning
  - Prototype model: identify similar cases
    (Chen et al. 2019; Hase et al. 2019)
Explanation Methods

● Feature importance estimates
  ○ LIME: local linear approximation (Ribeiro et al., 2016)
  ○ Anchors: if-then probabilistic statements (Ribeiro et al., 2018)

● Case-based reasoning
  ○ Prototype model: identify similar cases
    (Chen et al. 2019; Hase et al. 2019)

● Latent space traversal (counterfactual explanations)
  ○ Decision boundary: cross the decision boundary in data space
    (Joshi et al., 2018; Samangouei et al., 2018)
Explanation Methods

- **Feature importance estimates**
  - LIME: local linear approximation (Ribeiro et al., 2016)
  - Anchors: if-then probabilistic statements (Ribeiro et al., 2018)

- **Case-based reasoning**
  - Prototype model: identify similar cases
    (Chen et al. 2019; Hase et al. 2019)

- **Latent space traversal (counterfactual explanations)**
  - Decision boundary: cross the decision boundary in data space
    (Joshi et al., 2018; Samangouei et al., 2018)

- **Composite approach**
  - Combine above methods
Input, Label, and Model Output

\[ x = \text{Despite modest aspirations its occasional charms are not to be dismissed.} \]
\[ y = \text{Positive} \quad \hat{y} = \text{Negative} \]
Explanation Methods

● Feature importance estimates
  ○ LIME, Anchors (Ribeiro et al. 2016; Ribeiro et al. 2018)
  ○ Probabilistic if-then statements
    ■ If P(x) holds, there is a high probability that model will predict y
  ○ Search for Anchors in a multi-armed bandit framework

Anchor

\[ p(\hat{y} = \text{Negative} | \{\text{occasional}\} \subseteq x) \geq .95 \]
Explanation Methods

● Case-based reasoning
  ○ Prototype model: identify similar cases
    (Chen et al. 2019; Hase et al. 2019)
  ○ Keep a **per-class set of prototype vectors**, which are equal to
    vector representations of individual training data points
  ○ Compute class scores as the **highest similarity score** between the
    representation of a new data point and the learned prototypes

---

Prototype

Most similar prototype:
**Routine and rather silly.**
Similarity score: 9.96 out of 10
Important words: (none selected)
Explanation Methods

- Case-based reasoning
  - Prototype model: identify similar cases
    (Chen et al. 2019; Hase et al. 2019)
  - Keep a **per-class set of prototype vectors**, which are equal to vector representations of individual training data points
  - Compute class scores as the **highest similarity score** between the representation of a new data point and the learned prototypes

Prototype

Most similar prototype: *Routine and rather silly.*
Similarity score: 9.96 out of 10
Important words: (none selected)

\[
f(x_i)_c = \max_{p_k \in P_c} a(g(x_i), p_k)
\]
Explanation Methods

- **Latent space traversal**
  - Decision boundary: cross the decision boundary in data space
    (Joshi et al., 2018; Samangouei et al., 2018)
  - **Identify a counterfactual by sampling**, then choosing the closest counterfactual (by edit distance, then Euclidean)
  - **Greedily select one-word edits** that least changes the evidence, until we have the full set of edits.
    - *evidence* defined as difference between the two class scores

```
<table>
<thead>
<tr>
<th>Step</th>
<th>Event</th>
<th>Evidence Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 0</td>
<td></td>
<td>-5.21</td>
</tr>
<tr>
<td>Step 1</td>
<td>occasional → rare</td>
<td>-3.00</td>
</tr>
<tr>
<td>Step 2</td>
<td>modest → impressive</td>
<td>+0.32</td>
</tr>
</tbody>
</table>

x^{(e)} Despite *impressive* aspirations its *rare* charms are not to be dismissed.
```
$x = \text{Despite modest aspirations its occasional charms are not to be dismissed.}$
$y = \text{Positive} \quad \hat{y} = \text{Negative}$

**LIME**
- charms $+.05$
- modest $+.04$
- dismissed $-0.06$
- occasional $-0.11$
- despite $-0.18$

**Prototype**
Most similar prototype:
- Routine and rather silly.

Similarity score: 9.96 out of 10
Important words: (none selected)

**Decision Boundary**
- Step 0 | Evidence Margin: -5.21
- Step 1 | occasional $\rightarrow$ rare
          | Evidence Margin: -3.00
- Step 2 | modest $\rightarrow$ impressive
          | Evidence Margin: +0.32

**Anchor**

$p(\hat{y} = \text{Negative} \mid \{\text{occasional}\} \subseteq x) \geq 0.95$

$x^{(e)} = \text{Despite impressive aspirations its rare charms are not to be dismissed.}$
Experimental Results

- Two binary classification tasks with neural models
  - Textual: sentiment analysis (Pang et al., 2002)
  - Tabular: binary income prediction (Dua and Graff, 2017)
  - Counterfactuals are algorithmically constructed
Experimental Results

- Two binary classification tasks with neural models
  - Textual: sentiment analysis (Pang et al., 2002)
  - Tabular: binary income prediction (Dua and Graff, 2017)
  - Counterfactuals are algorithmically constructed

- 2166 responses from 29 undergraduates (in-person tests)
  - Quantitative backgrounds
  - Passed screening tests (mini task/method lessons with quiz)
Experimental Results

- Two binary classification tasks with neural models
  - Textual: sentiment analysis (Pang et al., 2002)
  - Tabular: binary income prediction (Dua and Graff, 2017)
  - Counterfactuals are algorithmically constructed

- 2166 responses from 29 undergraduates (in-person tests)
  - Quantitative backgrounds
  - Passed screening tests (mini task/method lessons with quiz)

- Hypothesis testing done by block bootstrap
Experimental Results

- Full tables in paper

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

<table>
<thead>
<tr>
<th>Method</th>
<th>n</th>
<th>Pre</th>
<th>Change</th>
<th>CI</th>
<th>p</th>
<th>n</th>
<th>Pre</th>
<th>Change</th>
<th>CI</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Avg.</td>
<td>1103</td>
<td>69.71</td>
<td></td>
<td>6.16</td>
<td>-</td>
<td>1063</td>
<td>63.13</td>
<td></td>
<td>7.87</td>
<td>-</td>
</tr>
<tr>
<td>LIME</td>
<td>190</td>
<td>-</td>
<td>5.70</td>
<td>9.05</td>
<td>.197</td>
<td>179</td>
<td>-</td>
<td>5.25</td>
<td>10.59</td>
<td>.309</td>
</tr>
<tr>
<td>Anchor</td>
<td>199</td>
<td>-</td>
<td>0.86</td>
<td>10.48</td>
<td>.869</td>
<td>197</td>
<td>-</td>
<td>5.66</td>
<td>7.91</td>
<td>.140</td>
</tr>
<tr>
<td>Prototype</td>
<td>223</td>
<td>-</td>
<td>-2.64</td>
<td>9.59</td>
<td>.566</td>
<td>192</td>
<td>-</td>
<td>9.53</td>
<td>8.55</td>
<td>.032</td>
</tr>
<tr>
<td>DB</td>
<td>205</td>
<td>-</td>
<td>-0.92</td>
<td>11.87</td>
<td>.876</td>
<td>207</td>
<td>-</td>
<td>2.48</td>
<td>11.62</td>
<td>.667</td>
</tr>
<tr>
<td>Composite</td>
<td>286</td>
<td>-</td>
<td>-2.07</td>
<td>8.51</td>
<td>.618</td>
<td>288</td>
<td>-</td>
<td>7.36</td>
<td>9.38</td>
<td>.122</td>
</tr>
</tbody>
</table>
Experimental Results

- LIME improves simulatability for tabular data.
  - 70.74% → 81.99% accuracy, +11.25 (± 8.83) ppts, p = .014
  - (across forward and counterfactual tests)
Experimental Results

- LIME improves simulatability for tabular data.
  - 70.74% → 81.99% accuracy, +11.25 (+/- 8.83) ppts, \( p = 0.014 \)
  - (across forward and counterfactual tests)

- Prototype model improves counterfactual simulatability.
  - 63.13% → 72.66% accuracy, +9.53 (+/- 8.55) ppts, \( p = 0.032 \)
  - (across datasets)
Experimental Results

- LIME improves simulatability for tabular data.
  - 70.74% → 81.99% accuracy, +11.25 (+/- 8.83) ppts, $p=0.014$
  - (across forward and counterfactual tests)

- Prototype model improves counterfactual simulatability.
  - 63.13% → 72.66% accuracy, +9.53 (+/- 8.55) ppts, $p=0.032$
  - (across datasets)

- Other estimates do not significantly differ from 0 ($p < 0.05$).
  - Including LIME for text, Prototype for forward sim., Anchor, Decision Boundary, and Composite methods
Experimental Results

- Do user ratings predict explanation effectiveness?
  - Ask users to rate explanations on 1-7 scale
  - “Does this explanation show me why the system thought what it did?”
  - Estimate counterfactual post test correctness from ratings
Experimental Results

● Do user ratings predict explanation effectiveness?
  ○ Ask users to rate explanations on 1-7 scale
  ○ “Does this explanation show me why the system thought what it did?”
  ○ Estimate counterfactual post test correctness from ratings

● Ratings not a significant predictor
  ○ Moving from a rating of 4 to 5 associated with between -2.9 and 5.2 ppt change in expected user accuracy (95% CI for text data)
Qualitative Analysis

- Success: 3 of 6 Pre correct → 5 of 6 Post correct

Original, predicted **positive**:

“Pretty much sucks, but has a funny moment or two.”

Counterfactual, predicted **positive**:

“*Mostly just bothers*, but *looks* a funny moment or two.”
Qualitative Analysis

- Success: 3 of 6 Pre correct → 5 of 6 Post correct

Original, predicted **positive**:
“Pretty much sucks, but has a funny moment or two.”

Counterfactual, predicted **positive**:
“Mostly just bothers, but looks a funny moment or two.”

Activated prototype:
“Murders by Numbers isn’t a great movie, but it’s a perfectly acceptable widget.”
Qualitative Analysis

● Failure: 7 of 13 Post correct (no improvements)

Original, predicted **positive**:
“A bittersweet film, simple in form but rich with human events.”

Counterfactual, predicted **negative**:
“A teary film, simple in form but vibrant with devoid events.”
Qualitative Analysis

- Failure: 7 of 13 Post correct (no improvements)

Original, predicted **positive**:
“A bittersweet film, simple in form but rich with human events.”

Counterfactual, predicted **negative**:
“A teary film, simple in form but vibrant with devoid events.”

- Was “bittersweet” necessary? Is vibrant considered similar to “rich”? If a sentence has the same syntactic structure, will it get the same prediction?
Concluding Thoughts

- With the proper controls, simulation tests provide a general purpose evaluation procedure.

- Explanation methods could be improved:
  - Best tabular Post accuracy: 81.99%
  - Best text Post accuracy: 66.47%
  - (baseline: 50%)
Concluding Thoughts

- With the proper controls, simulation tests provide a general purpose evaluation procedure.

- Explanation methods could be improved:
  - Distinguish between sufficient and necessary factors
  - Clearly point to decision-relevant similarities between new inputs and known cases
  - Use feature spaces appropriate to the problem (individual words probably a suboptimal feature space)
Our follow-up work

- Natural language explanations
  - Leakage-Adjusted Simulatability: Can Models Generate Non-Trivial Explanations of Their Behavior in Natural Language?
Our follow-up work

● Natural language explanations
  ○ Leakage-Adjusted Simulatability: Can Models Generate Non-Trivial Explanations of Their Behavior in Natural Language?

● Explaining models in terms of influential data
  ○ FastIF: Scalable Influence Functions for Efficient Model Interpretation and Debugging
Our follow-up work

- **Natural language explanations**
  - Leakage-Adjusted Simulatability: Can Models Generate Non-Trivial Explanations of Their Behavior in Natural Language?

- **Explaining models in terms of influential data**
  - FastIF: Scalable Influence Functions for Efficient Model Interpretation and Debugging

- **Feature importance explanations**
  - Search Methods for Sufficient, Socially-Aligned Feature Importance Explanations with In-Distribution Counterfactuals
Our follow-up work

- **Natural language explanations**
  - Leakage-Adjusted Simulatability: Can Models Generate Non-Trivial Explanations of Their Behavior in Natural Language?

- **Explaining models in terms of influential data**
  - FastIF: Scalable Influence Functions for Efficient Model Interpretation and Debugging

- **Feature importance explanations**
  - Search Methods for Sufficient, Socially-Aligned Feature Importance Explanations with In-Distribution Counterfactuals

- **Teaching models via explanations**
  - When Can Models Learn From Explanations? A Formal Framework for Understanding the Roles of Explanation Data
Others’ follow-up work

● Explanations in a human-AI team context
  ○ Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance
Others’ follow-up work

- **Explanations in a human-AI team context**
  - Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance

- **More theory: faithfulness, social alignment of explanations**
  - Aligning Faithful Interpretations with their Social Attribution
Others’ follow-up work

- **Explanations in a human-AI team context**
  - Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance

- **More theory: faithfulness, social alignment of explanations**
  - Aligning Faithful Interpretations with their Social Attribution

- **Automating our evaluation (as a model-based evaluation)**
  - Evaluating Explanations: How much do explanations from the teacher aid students?
Others’ follow-up work

- **Explanations in a human-AI team context**
  - Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance

- **More theory: faithfulness, social alignment of explanations**
  - Aligning Faithful Interpretations with their Social Attribution

- **Automating our evaluation (as a model-based evaluation)**
  - Evaluating Explanations: How much do explanations from the teacher aid students?

- **Counterfactual explanations for NLP**
  - Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models
Simulation Tests in RL

- Explainable Reinforcement Learning Through a Causal Lens
  - Ask people to predict what an agent will do next, based on varying kinds of explanations
Simulation Tests in RL

● Explainable Reinforcement Learning Through a Causal Lens
  ○ Ask people to predict what an agent will do next, based on varying kinds of explanations

● More explainable RL work summarized in our blog post:
  ○ Opinions on Interpretable Machine Learning and 70 Summaries of Recent Papers
Thank You!

Code: https://github.com/peterbhhase/interpretableNLP-ACL2020

Contact Info:
Peter Hase
peter@cs.unc.edu
https://peterbhase.github.io