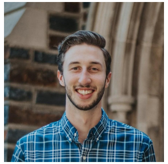
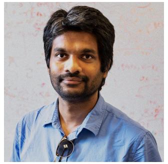
Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?





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

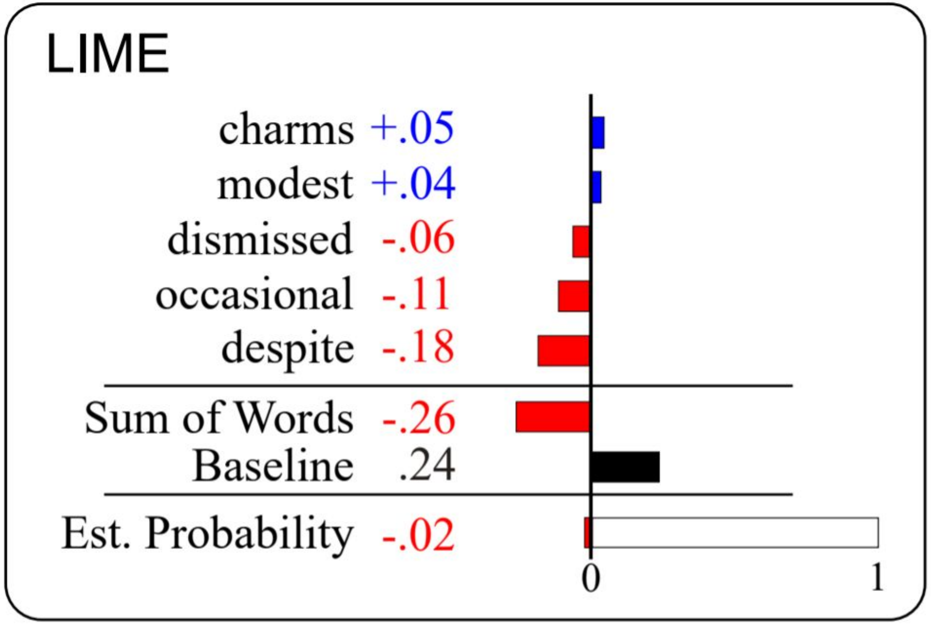
Motivation

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

Input, Label, and Model Output

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

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

UNC NLP

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

$$\begin{array}{c}\text{Post Sim.}\\ \text{Accuracy}\end{array} - \begin{array}{c}\text{Pre Sim.}\\ \text{Accuracy}\end{array} = \begin{array}{c}\text{Explanation}\\ \text{Effect}\end{array}$$

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

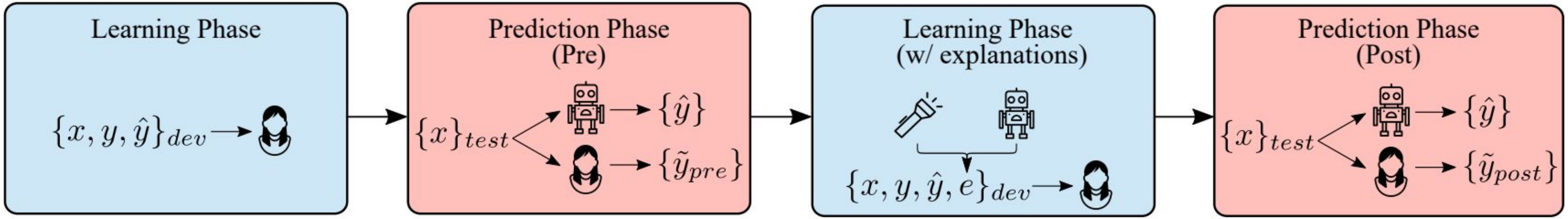
- Measure the effect of an explanation method on model simulatability
- Important controls:
  - Separate explained instances from test instances
  - Evaluate the effect of explanations against a baseline of unexplained examples
  - Balance data by model correctness and model output

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



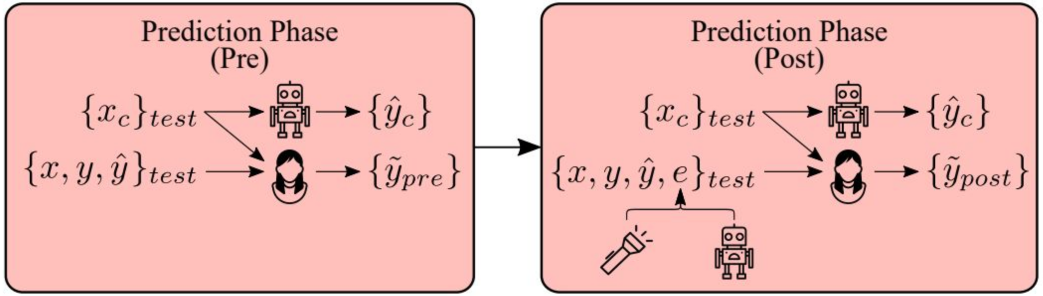
e : Explanation

$\hat{y}$: Model prediction

$\tilde{y}$: Human simulation

Proposal: Experimental Design

- Test 2: counterfactual simulation



e : Explanation

$\hat{y}$: Model prediction

$\tilde{y}$: Human simulation

x_c : Counterfactual input

$\hat{y}_c$: 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

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

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

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

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)

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

UNC NLP

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

Decision Boundary

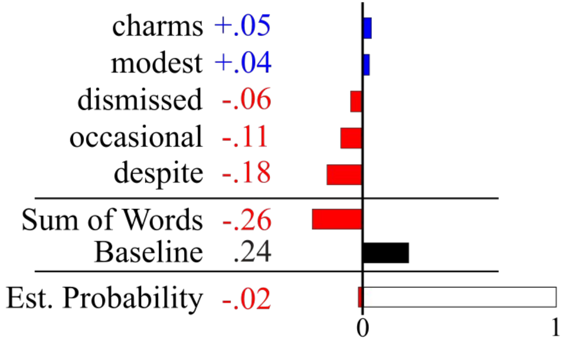
Step 0	Evidence Margin: -5.21
Step 1	occasional → rare Evidence Margin: -3.00
Step 2	modest → impressive Evidence Margin: +0.32
$x^{(c)}$	Despite *impressive* aspirations its *rare* charms are not to be dismissed.

UNC NLP

Input, Label, and Model Output

$x =$ Despite modest aspirations its occasional charms are not to be dismissed.
$y =$ Positive $\quad \hat{y} =$ Negative

LIME



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 $\longrightarrow$ rare
Evidence Margin: -3.00

Step 2 | modest $\longrightarrow$ impressive
Evidence Margin: +0.32

$x^{(c)}$ | 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

	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

	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

UNC NLP

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=.014
  - (across forward and counterfactual tests)
- Prototype model improves counterfactual simulatability.
  - 63.13% → 72.66% accuracy, +9.53 (+/- 8.55) ppts, p=.032
  - (across datasets)

UNC NLP

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)
- Prototype model improves counterfactual simulatability.
  - 63.13% → 72.66% accuracy, +9.53 (+/- 8.55) ppts, p=.032
  - (across datasets)
- Other estimates do not significantly differ from 0 (p <.05).
  - Including LIME for text, Prototype for forward sim., Anchor, Decision Boundary, and Composite methods

UNC NLP

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."

UNC NLP

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/peterbhase/InterpretableNLP-ACL2020

Contact Info:

Peter Hase

peter@cs.unc.edu

https://peterbhase.github.io