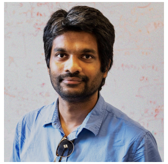
The Out-of-Distribution Problem in Explainability and Search Methods for Finding Feature Importance Explanations





Peter Hase, Harry Xie, Mohit Bansal

{peter, fengyu.xie, mbansal}@cs.unc.edu

NeurIPS 2021

Talk Outline

- Background: Feature Importance Explanations
- Out-of-Distribution Problem in Explainability
  - Why is this a problem?
  - Proposed solution
- Out-of-Distribution Experiments
  - Verify solution effectiveness
  - Select a Replace function
- Search Methods for Feature Importance Explanations
- Explanation Method Evaluation
- Discussion & Conclusions

UNC NLP

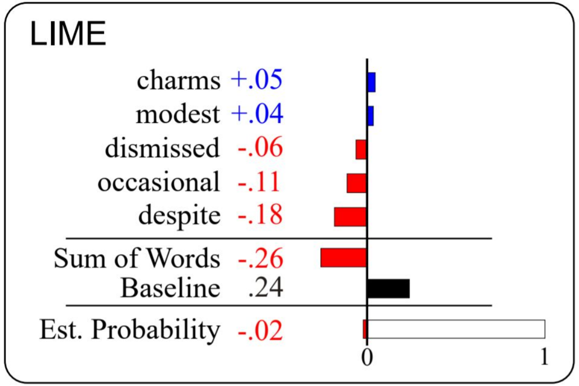
Background: Feature Importance Explanations

- We have feature importance explanations of model decisions
  - (also known as salience, heatmaps, attributions, etc.)
- Example: sentiment analysis

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)

Out-of-Distribution Problem in Explainability

- Are these features actually important to the model?
- Remove important features, check if the model's predicted probability declines
  - We call the edited input a *counterfactual*
- This is the *Comprehensiveness* metric (DeYoung et al., 2020)

$$\mathrm{Comp}(f, x, e) = f(x)_{\hat{y}} - f(\boxed{\texttt{Replace}}(x, e))_{\hat{y}}$$

Typically arbitrarily chosen:

1. Delete words
2. Replace tokens with MASK or UNK token
3. Set embedding to 0
4. Impute words
5. Etc.

These are all out of distribution to a model trained on real data!

Out-of-Distribution Problem in Explainability

- Why can't counterfactuals be OOD to a model?
- At least 15 papers express unease with this situation
- Arguments are generally founded in intuition or basic machine learning principles
  - Models produce "poor" or "insensible" predictions on OOD data (multiple papers)
  - Can produce "sub-optimal attributions" (multiple papers)
  - "It is unclear whether the degradation in model performance comes from the distribution shift or because the features that were removed are truly informative" (Hooker et al., 2019)

$$\mathrm{Comp}(f, x, e) = f(x)_{\hat{y}} - f(\mathtt{Replace}(x, e))_{\hat{y}}$$

"OOD causes degradation" =
If this wasn't OOD, model might reproduce original prediction

- But for a particular trained model, there is no ambiguity regarding the cause of this difference! **"If this wasn't OOD" = if we had a different model**

Out-of-Distribution Problem in Explainability

- We need a stronger argument for not using OOD counterfactuals when explaining a *particular* trained model

We claim that OOD counterfactuals yield **socially-misaligned** explanations and explanation metrics.

(Jacovi and Goldberg, 2021)

- We say both **explanations metrics** and **explanations themselves** are socially misaligned because OOD counterfactuals are regularly used to *obtain* explanations
- *Socially misaligned*: someone's expectation of the kind of information that an explanation will communicate is not fulfilled by what it actually communicates
    - Expectation: explanation = *evidence used to reach a decision*
    - Reality: explanation = *evidence selected after a decision was made*
    - → social misalignment

UNC NLP

Out-of-Distribution Problem in Explainability

- An illustrative example: **classify the sentiment of individual words with BERT**
  - "Good" → positive
  - "Gross" → negative
  - Train BERT on a sentiment dictionary
  - Evaluate on held-out words
- Compute Comprehensiveness by replacing words with the MASK token

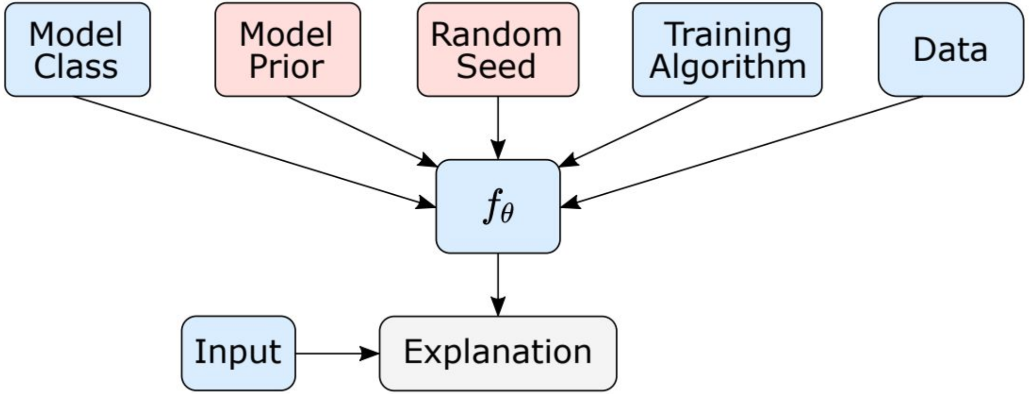
$$\mathrm{Comp}(f, x, e) = f(x)_{\hat{y}} - f(\texttt{Replace}(x, e))_{\hat{y}}$$

$$f_\theta(x = \texttt{MASK} | \mathcal{D})_{\hat{y}}$$

Not learned from data! Depends heavily on the model prior

Out-of-Distribution Problem in Explainability

- People expect a feature importance explanation to reflect how the model **has learned** to interpret features as evidence for a particular decision
- People do not expect FI explanations to be influenced by the **model prior** (the designer's choice) or **random factors** (which are not meaningful)



Out-of-Distribution Problem in Explainability

- Compute Sufficiency by replacing words with the MASK token

$$\text{Comp}(f, x, e) = f(x)_{\hat{y}} - f(\texttt{Replace}(x, e))_{\hat{y}}$$

$$f_\theta(x = \texttt{MASK} | \mathcal{D})_{\hat{y}}$$

If this was learned from data, it would reflect uncertainty in the label, NOT the model prior.

Out-of-Distribution Problem in Explainability

- Solution: **train model on the counterfactuals it will see at explanation-time** (*Counterfactual Training*)
  - Weight equally with the original data
  - Explanations can be expensive to produce (1000+ forward passes)
  - In practice, use random explanations (a good approximation in theory - Jethani et al., 2021)
  - Small hit to accuracy (0.7 points on average across six datasets)

$$\mathrm{Comp}(f, x, e) = f(x)_{\hat{y}} - f(\texttt{Replace}(x, e))_{\hat{y}}$$

$$f_\theta(x = \texttt{MASK} | \mathcal{D})_{\hat{y}}$$

Influenced by model prior, random seed $\longrightarrow$ **Influenced by what the model has learned from data**

Out-of-Distribution Problem in Explainability

- Solution: **train model on the counterfactuals it will see at explanation-time** (*Counterfactual Training*)
  - Weight equally with the original data
  - Explanations can be expensive to produce (1000+ forward passes)
  - In practice, use random explanations (a good approximation in theory - Jethani et al., 2021)
  - Small hit to accuracy (0.7 points on average across six datasets)

Dataset	Standard Acc.	CT Acc.
SNLI	85.84 (0.69)	85.08 (0.71)
BoolQ	74.16 (1.62)	73.76 (1.62)
FEVER	89.66 (0.76)	89.72 (0.76)
Evidence Inference	58.81 (3.12)	57.35 (3.13)
SST-2	92.89 (1.18)	92.43 (1.21)
MultiRC	68.96 (1.30)	67.76 (1.32)

Out-of-Distribution Experiments

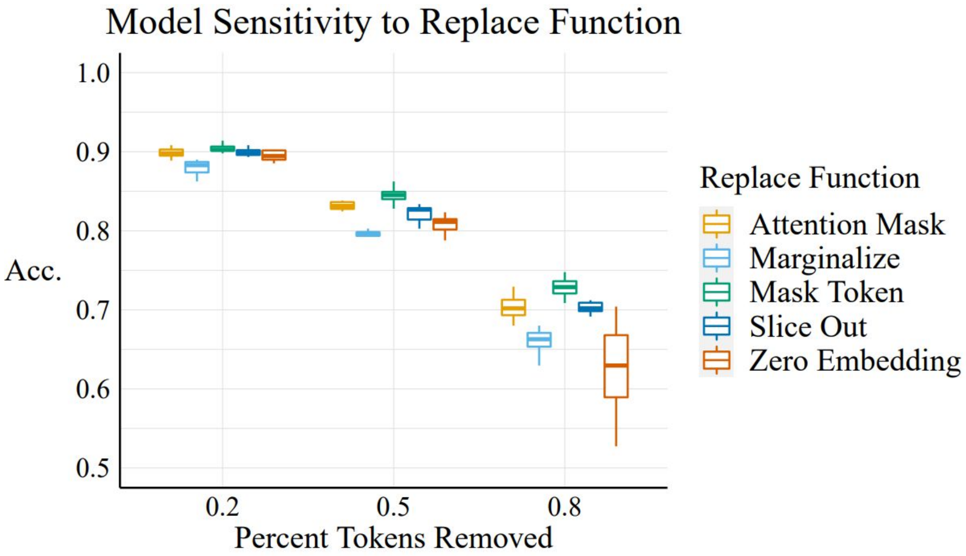
- **RQ1: Which Replace function should you use?**
- **RQ2: Is Counterfactual Training (CT) effective?**

Out-of-Distribution Experiments

- Outcome: degree of distribution shift / robustness
  - Drop in accuracy of a trained model when given Replace(x,e) inputs, using random e
- Measure for multiple Replace functions
  - *Attention Mask*: Set attention weight to 0 for token
  - *Marginalize*: Use an MLM to impute tokens, marginalize predictions over that distribution
  - *MASK Token*: Replace tokens with MASK token
  - *Slice Out*: Delete tokens from the input sequence (affects position embedding)
  - *Zero vector*: Replace token embedding with zero vector
- Measure for Standard vs CT models
  - BERT + RoBERTa
  - Trained on 10k train points from SNLI and SST2
- 10 seeds for everything (10 different models and sets of explanations)

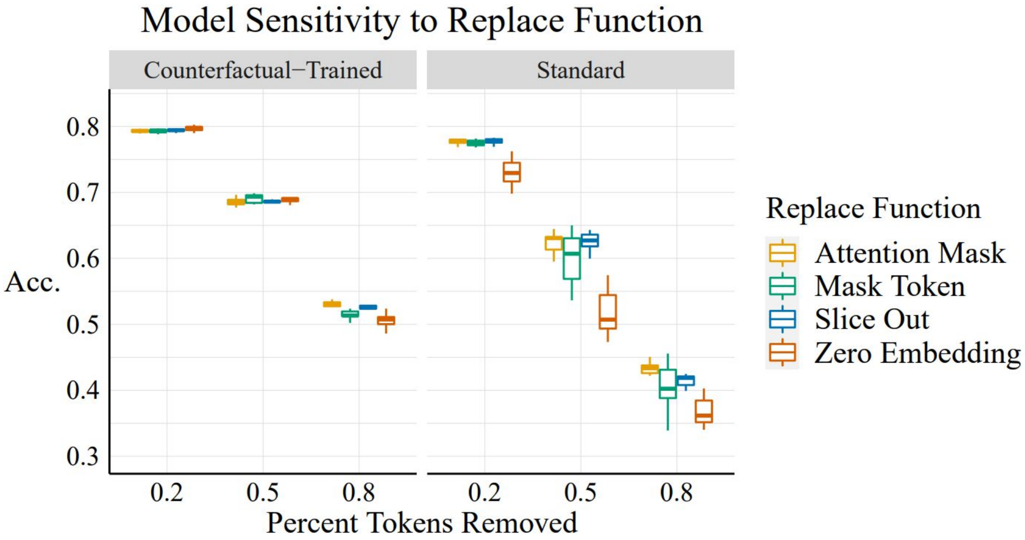
Out-of-Distribution Experiments

- If **not using Counterfactual-Training**, we recommend using Attention Mask or Mask Token



Out-of-Distribution Experiments

- But **Counterfactual-Training is much more important than the choice of Replace function**
- We recommend that you **use CT if you want to explain your model**



UNC NLP

Search Methods for FI Explanations

- Now it's time to assess some explanations (on **both Standard and CT models**)
  - *Sufficiency*: does keeping selected features raise the model confidence?
  - *Comprehensiveness*: does removing selected features lower the model confidence?
- Objective (following DeYoung et al., 2020):

$$\arg\max_{E} \frac{1}{|S|}\sum_{i=1}^{|S|} \text{Suff}(f, x, e_i, s_i) \quad \text{s.t. } e_i \in \{0,1\}^d \text{ and } \sum_{d} e_i^{(d)} \leq \text{ceiling}(s_i \cdot d)$$

Get a **set of explanations** (of varying sparsity)

Indicate features to keep/remove

Limit on # features (sparsity)

- Typically people use salience methods, which output scalar values for each feature
- Search methods are good for combinatorial optimization problems too

Search Methods for FI Explanations

- Salience methods
  - LIME (Ribeiro et al., 2016)
  - Vanilla gradients (Li et al., 2015)
  - Integrated Gradients (Sundararajan et al., 2017)
- Search Methods
  - Anchors (Ribeiro et al., 2018)
  - Random Search
  - Gradient Search (similar to Fong and Vedaldi, 2017)
  - Taylor Search (similar to Ebrahimi et al., 2018)
  - Ordered Search
  - Parallel Local Search
- All methods except Integrated Gradients use the **Attention Mask** Replace function
- **Control for compute budget**: 1000 forward or backward passes per explanation
  - (exact time depends on input size and method; can take only a few seconds)

Search Methods for FI Explanations

- Parallel Local Search:
  (1) Sample a random initial explanation *e* and compute the objective function (Suff or Comp) for that explanation.
  (2) For the remaining budget of *b−1* steps: sample a not-already-seen neighboring explanation *e** and adopt *e** as the new state if it is a new best explanation.
- Neighboring means two elements in *e* flip (from 0 to 1 or vice versa)
- Done in parallel *r=10* threads
- Requires defining Replace(x,e)

Search Methods for FI Explanations

- Parallel Local Search:
  - Using Attention Mask Replace function
  - Find sufficient subset of inputs in ~2 seconds

```
Loading model...
Searching for explanation for point 0...took 2.23 seconds!
Model input:  <s>A dog swims in a pool.</s></s>A puppy is swiming.</s>
Explanation:   __ __ swim __ __ __ __ __ __ puppy __ swim __ __
Model predicts neutral with pred prob: 0.937 | Pred prob for explanation: 0.969 | suff: -3.11 points
Input label: neutral
```

Explanation Method Evaluation

- Outcomes: **Suff and Comp** scores on **Standard and CT** models
- Six benchmark tasks: SNLI, BoolQ, FEVER, SST2, MultiRC, Evidence Inference
- On tasks with input=(question, document), we never Replace the question

Explanation Method Evaluation

Dataset	Method	Sufficiency ↓ Standard Model	Sufficiency ↓ CT Model	Comprehensiveness ↑ Standard Model	Comprehensiveness ↑ CT Model
SNLI	LIME	20.00 (2.02)	27.08 (1.68)	82.18 (2.82)	75.34 (1.93)
	Int-Grad	43.76 (3.27)	32.91 (2.36)	34.01 (2.55)	43.22 (2.28)
	Anchors	11.93 (1.52)	30.96 (1.87)	55.72 (2.62)	48.86 (2.37)
	Gradient Search	17.55 (1.47)	33.98 (1.43)	53.15 (2.53)	49.36 (1.95)
	Taylor Search	6.91 (1.10)	28.00 (1.46)	73.20 (2.57)	66.76 (2.12)
	Ordered Search	-1.45 (0.93)	15.06 (1.37)	87.78 (2.41)	84.67 (1.61)
	Random Search	-1.54 (0.96)	15.38 (1.39)	87.36 (2.47)	84.63 (1.68)
	Parallel Local Search	**-1.65** (1.07)	**14.16** (1.38)	**87.95** (2.55)	**86.18** (1.45)

- **PLS is best in 21 of 24 conditions** (at *p*=.05), **by up to 17.6 points** over next best
- LIME is the best salience method, but it is best overall only once and is outperformed by Random Search on Sufficiency 9/10 times
- Suff and Comp scores are often much worse for CT models than for Standard models, by up to 24 points: **non-CT models have inflated scores!**

UNC NLP

Explanation Method Evaluation

		Sufficiency ↓		Comprehensiveness ↑	
Dataset	Method	Standard Model	CT Model	Standard Model	CT Model
FEVER	LIME	-0.24 (0.50)	0.39 (0.96)	33.86 (3.43)	22.06 (2.36)
	Int-Grad	9.72 (1.80)	4.99 (1.40)	17.81 (2.47)	13.69 (1.71)
	Anchors	6.19 (1.22)	6.36 (1.10)	20.82 (2.58)	11.94 (1.84)
	Gradient Search	0.66 (0.68)	2.63 (1.12)	19.26 (2.68)	11.44 (1.65)
	Taylor Search	4.17 (0.96)	4.20 (1.20)	24.51 (2.78)	15.62 (1.85)
	Ordered Search	-1.26 (0.41)	-0.01 (0.90)	31.79 (3.28)	18.90 (2.46)
	Random Search	-1.51 (0.51)	-1.24 (2.33)	32.47 (3.33)	18.84 (2.11)
	PLS	**-2.04** (0.62)	**-3.66** (0.82)	**37.72** (3.28)	**24.07** (2.46)

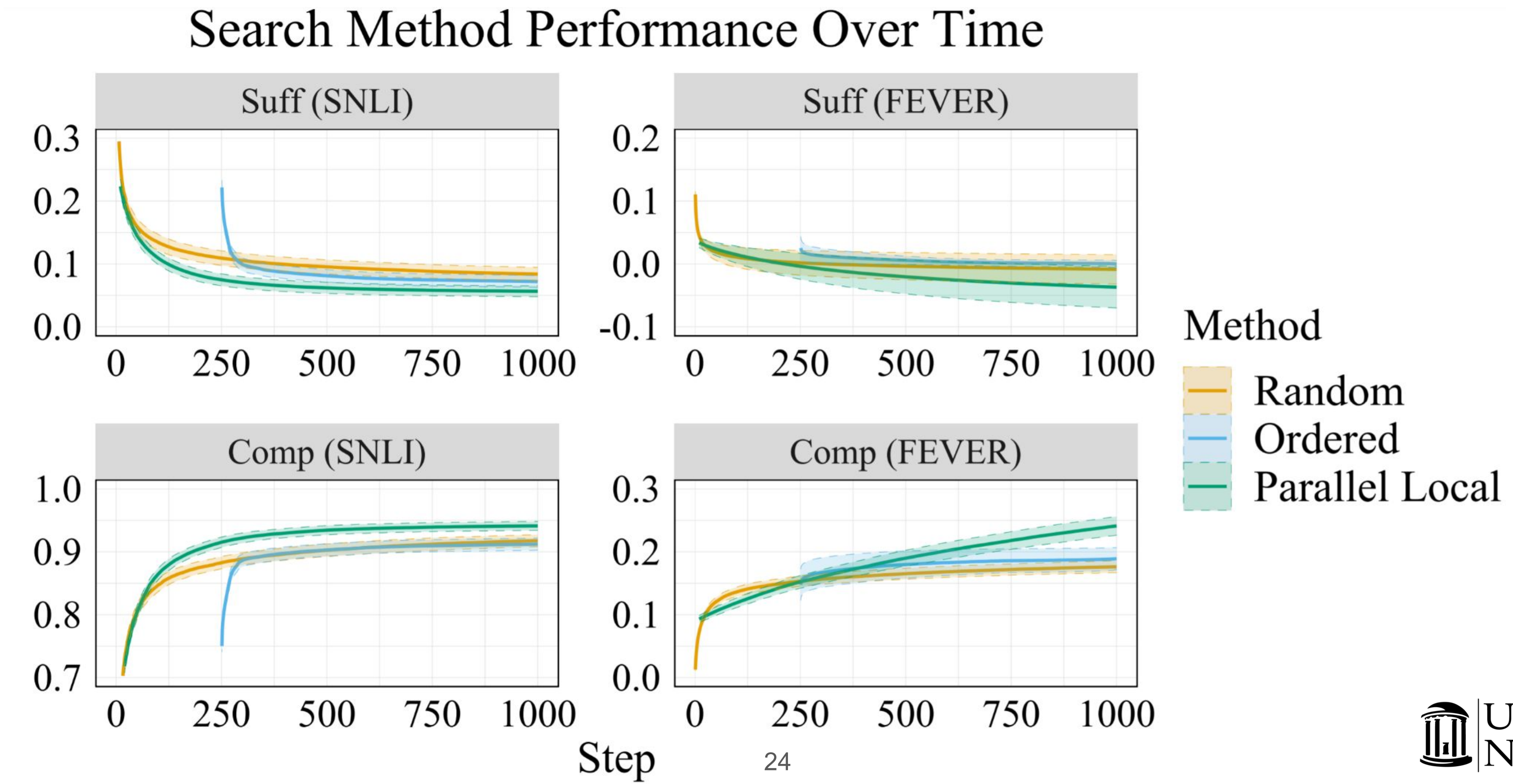
- Results hold for longer sequences too (FEVER avg length: 278 vs 24.4 for SNLI)

Explanation Method Evaluation

		Sufficiency ↓		Comprehensiveness ↑	
Dataset	Method	Standard Model	CT Model	Standard Model	CT Model
SST-2	LIME	1.98 (0.84)	5.92 (0.93)	52.42 (2.92)	45.75 (2.49)
	Anchors	3.44 (0.96)	17.69 (1.64)	30.03 (3.13)	24.19 (2.54)
	Taylor Search	0.09 (0.50)	5.02 (0.79)	45.65 (3.11)	38.91 (2.70)
	Ordered Search	-0.91 (0.47)	2.69 (0.79)	56.24 (2.82)	49.21 (2.48)
	Random Search	-0.91 (0.48)	2.70 (0.79)	56.11 (2.85)	48.98 (2.49)
	PLS	-0.91 (0.51)	2.68 (0.85)	56.28 (2.84)	49.25 (2.53)
	Exhaustive Search	-0.91 (0.51)	2.68 (0.85)	56.29 (2.84)	49.26 (2.53)

- Search is typically optimal on short sequences (typically <=10 tokens)

Explanation Method Evaluation



Discussion & Conclusions

- **If you want to explain your model, train it with Counterfactual-Training**
  - "Should we prefer Counterfactual-Trained models if they are harder to explain?"
  - Explanation metrics are lower for CT models *because* they are socially aligned
  - We want explanations to communicate what a model has learned, rather than the model prior and random seed
  - *Disclaimer*: we can't guarantee that CT *eliminates* the influence of the model prior and random seed
- **Search methods are the new SOTA for Sufficiency and Comprehensiveness**
  - Across six NLP benchmarks
  - Outperforms model-based approximations and gradient-based methods
- It is very important to **control for compute across methods** in experiments
  - It is very rare for papers to discuss the compute budgets used in experiments
  - Results vary substantially with compute budget
  - We see performance benefits from going beyond 1000 samples (i.e. forward passes)
  - How much compute *should* go into explanations?

UNC NLP

Summary of Algorithms

- **Counterfactual Training**
  - Pick a Replace function
  - Augment the training data in equal parts with Replace(x,e) pairs, using random e
  - Train on data
- **Parallel Local Search**
  - Use same Replace function as during training
  - Start with 10 random explanations of a specified sparsity
  - Perform 10 greedy local searches starting from each explanation
  - Return the best explanation according to the Comprehensiveness or Sufficiency objective
  - Repeat for different sparsity levels / objectives as desired

Thank You!

Code: https://github.com/peterbhase/ExplanationSearch (includes a demo of PLS)

```
Loading model...
Searching for explanation for point 0...took 2.23 seconds!
Model input:  <s>A dog swims in a pool.</s></s>A puppy is swiming.</s>
Explanation:   __ __ swim __ __ __ __ __ __ puppy __ swim __ __
Model predicts neutral with pred prob: 0.937 | Pred prob for explanation: 0.969 | suff: -3.11 points
Input label: neutral
```

NeurIPS talk on YouTube: https://www.youtube.com/watch?v=OZ0fSCQ7axw&t=3s

Contact Info:

Peter Hase, UNC Chapel Hill
peter@cs.unc.edu
https://peterbhase.github.io