Interpreting Predictions of NLP Models

Eric Wallace, Matt Gardner, Sameer Singh

EMNLP 2020

UC Berkeley

Allen Institute for AI

UC Irvine

Eric Wallace
UC Berkeley

Matt Gardner
Allen Institute for AI

Sameer Singh
UC Irvine

Slides and Video ericswallace.com/interpretability

Tutorial Outline

(1) Overview of Interpretability (Matt, 25 min)

(2) What Parts of An Input Led to a Prediction? (Sameer, 45 min)

(3) What Decision Rules Led to a Prediction? (Eric, 20 min)

Open QA (30 min)

Break (30 min)

(4) Which Training Examples Caused a Prediction? (Sameer, 20 min)

(5) Implementing Interpretations (Matt, 15 min)

(6) Open Problems (Eric, 25 min)

Open QA + Discussion (30 min)

Tutorial Outline

(1) Overview of Interpretability

(2) What Parts of An Input Led to a Prediction?

(3) What Decision Rules Led to a Prediction?

(4) Which Training Examples Caused a Prediction?

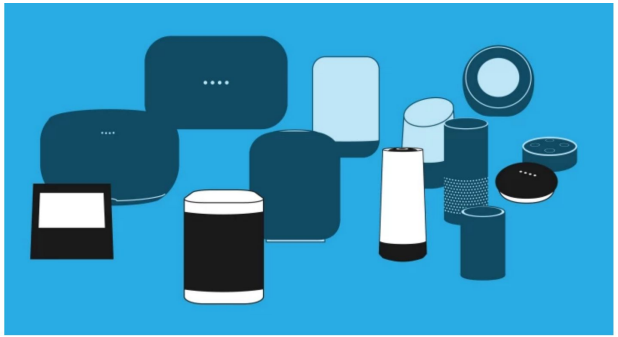
(5) Implementing Interpretations

(6) Open Problems

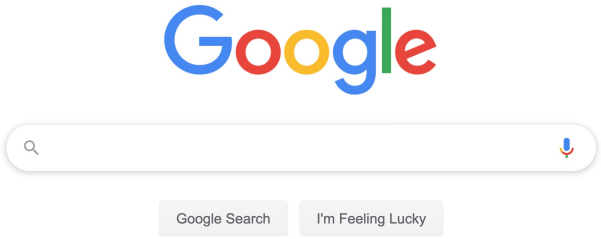
Notable Successes of Neural NLP Models

Speech Recognition

Machine Translation



Smart Assistants

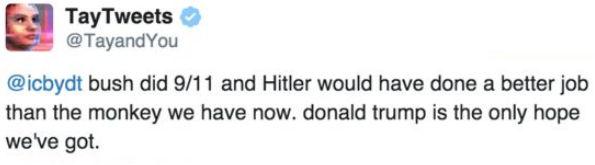


Information Retrieval

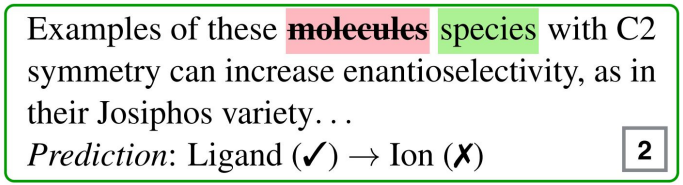
Story Generation

.....

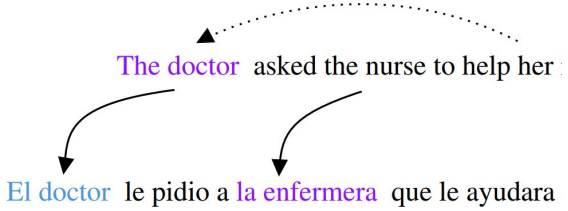
Notable **Failures** of Neural NLP Models



Generate Hate Speech



Fragile to Small Edits



Reflect Gender Biases

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Rely on pattern matching

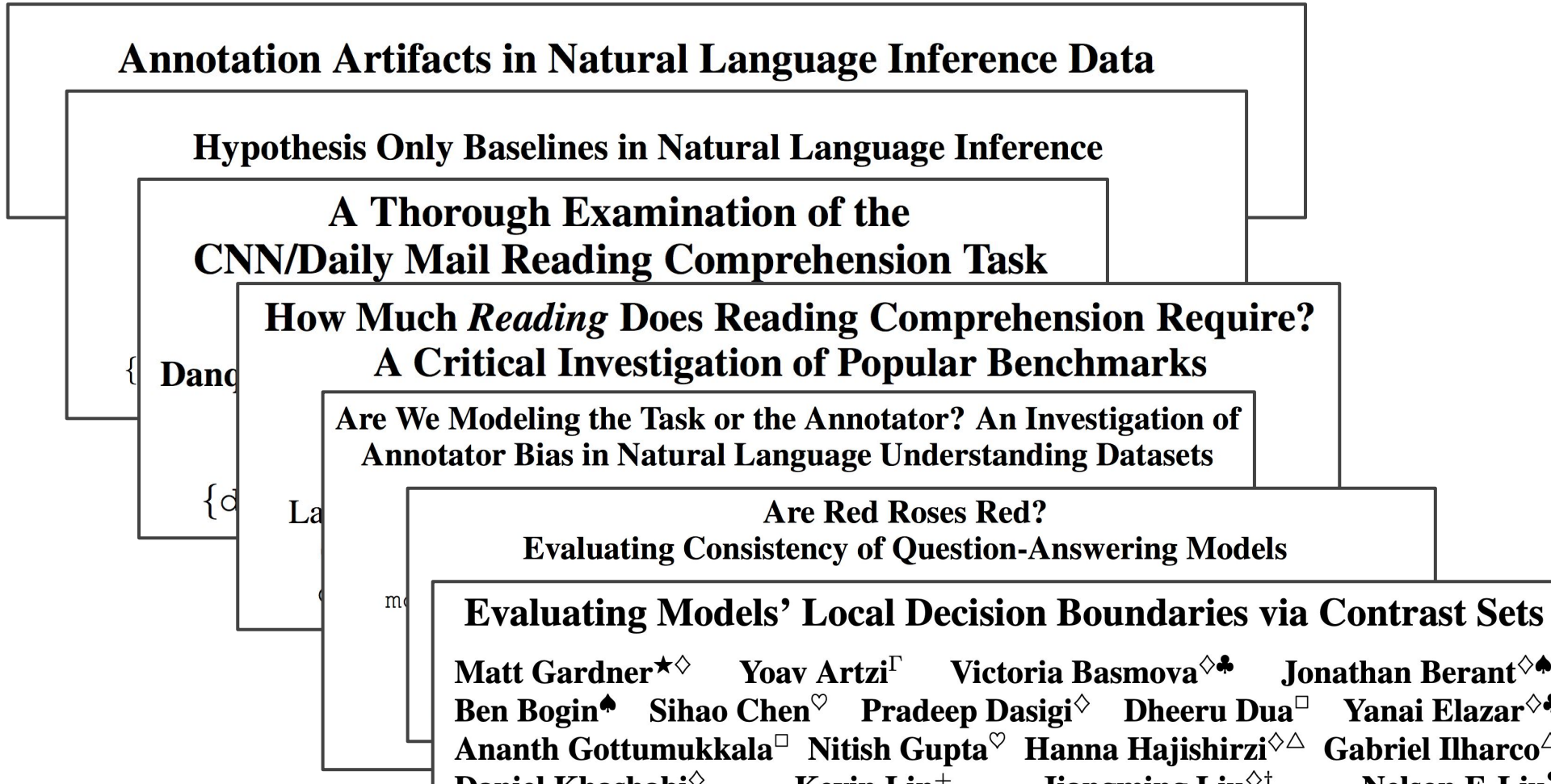
SQuAD

Context	In 1899, John Jacob Astor IV invested $100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his Colorado Springs experiments.
Original	What did Tesla spend Astor's money on ?
Reduced	**did**
Confidence	0.78 → 0.91

Behave Counterintuitively

. . . . . .

Notable Failures of Neural NLP Models



Looking at metrics
is **not** enough!

Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models

The [MASK] ran to the emergency room to see his patient.

Mask 1 Predictions:

36.5%	**doctor**
12.7%	**man**
2.8%	**boy**
2.7%	**nurse**
2.0%	**patient**

The [MASK] ran to the emergency room to see her patient.

Mask 1 Predictions:

44.9%	**nurse**
19.3%	**woman**
7.4%	**doctor**
5.3%	**girl**
3.6%	**mother**

[CLS] The [MASK] ran to the emergency room to see her patient . [SEP]

[Ribeiro et al. 2016, Belinkov and Bisk 2018, Wallace et al. 2019]

Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models
- Find errors and bugs in data

Test Example

Polar Bear ✘

Important Training Example

Polar Bear ✘

[Yeh et al. 2018, Koh and Liang 2017]

Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models
- Find errors and bugs in data
- Understand how models work so we can improve them

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

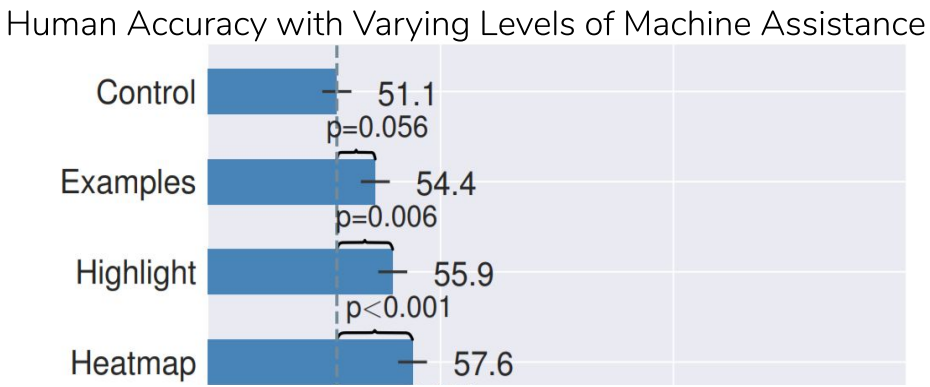
SQuAD

Context	In 1899, John Jacob Astor IV invested $100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his Colorado Springs experiments.
Original	What did Tesla spend Astor's money on ?
Reduced	**did**
Confidence	0.78 → 0.91

[Jia and Liang 2017, Ribeiro et al. 2018, Feng et al. 2018]

Beyond validation metrics

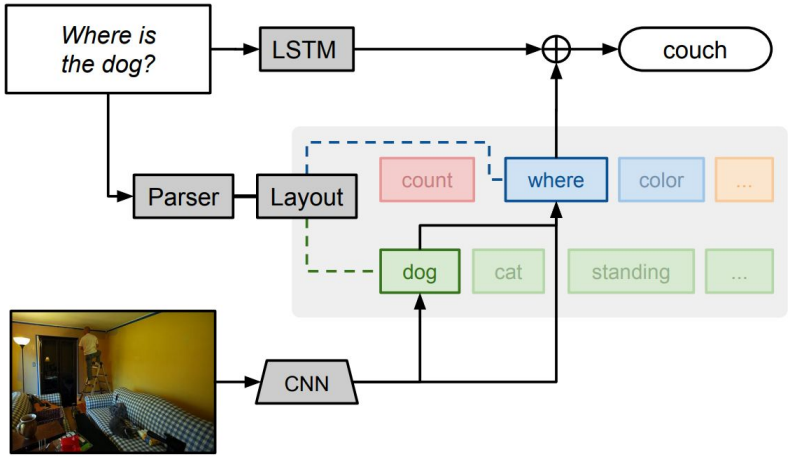
- Find errors, bugs, and undesirable behavior in models
- Find errors and bugs in data
- Understand how models work so we can improve them
- Understand how models work so they are trusted (or not trusted)

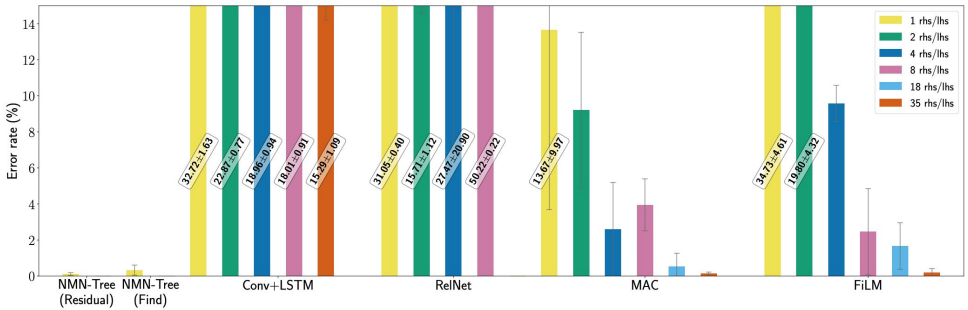


[Ribeiro et al. 2016, Lai and Tan 2019, Feng and Boyd-Graber 2019, Narayanan et al. 2018]

Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models
- Find errors and bugs in data
- Understand how models work so we can improve them
- Understand how models work so they are trusted (or not trusted)
- Have understandable models so they work better

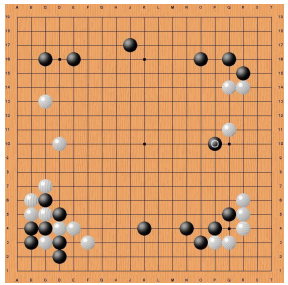


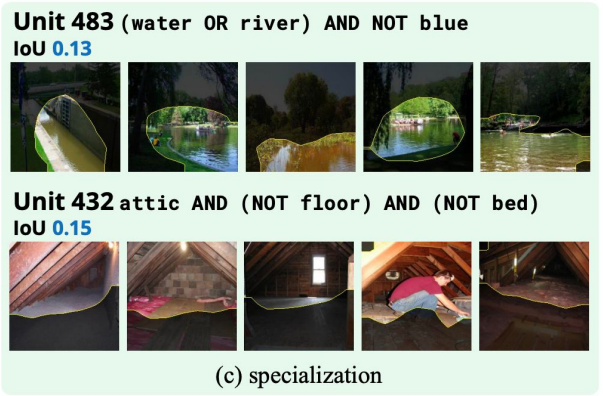


[Andreas et al. 2016, Bahdanau et al. 2018, Bogin et al. 2020]

Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models
- Find errors and bugs in data
- Understand how models work so we can improve them
- Understand how models work so they are trusted (or not trusted)
- Have understandable models so they work better
- Uncover the patterns that models find, for scientific discovery





(c) specialization

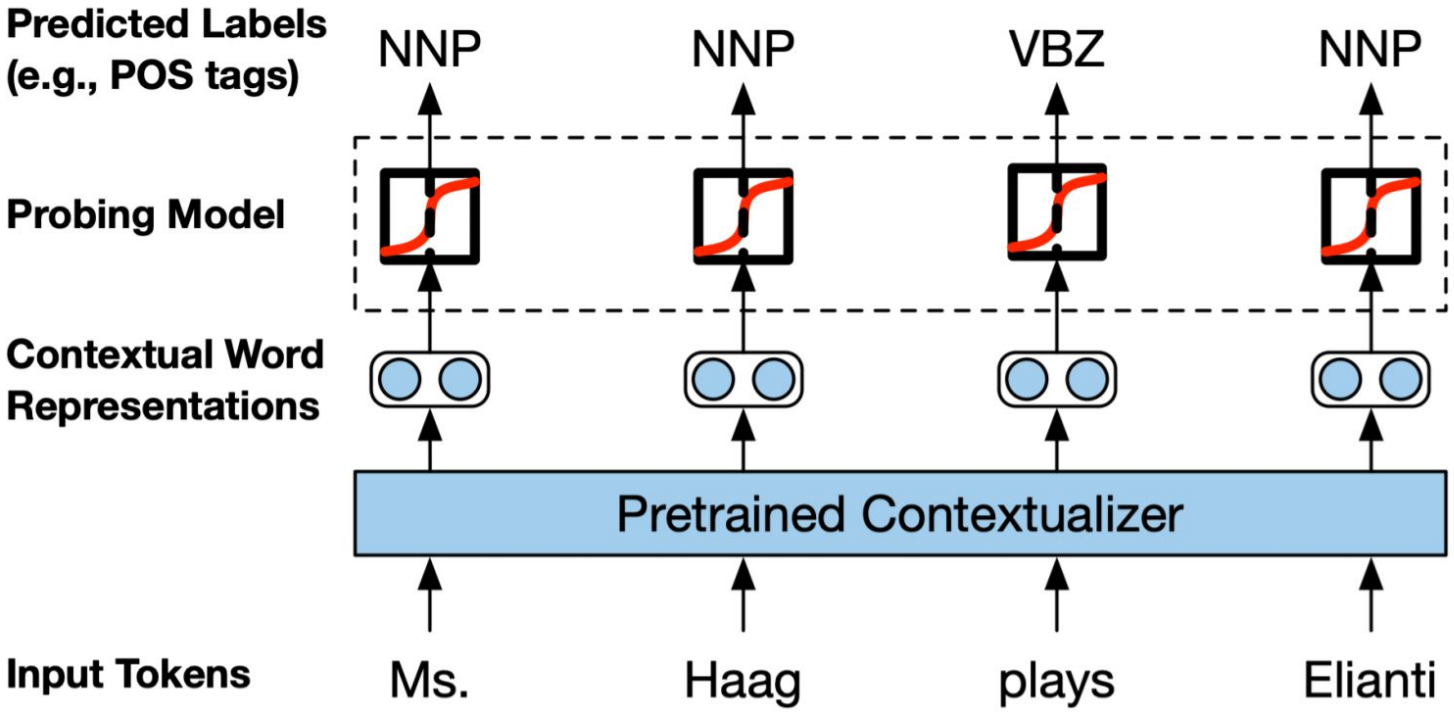
(d) polysemanticity

[Silver et al. 2016, Mu and Andreas 2020]

Ok, but how?

Interpretation methods

- Probing internal representations



[Shi et al. 2016, Adi et al. 2017, Conneau et al. 2018, ...]

Interpretation methods

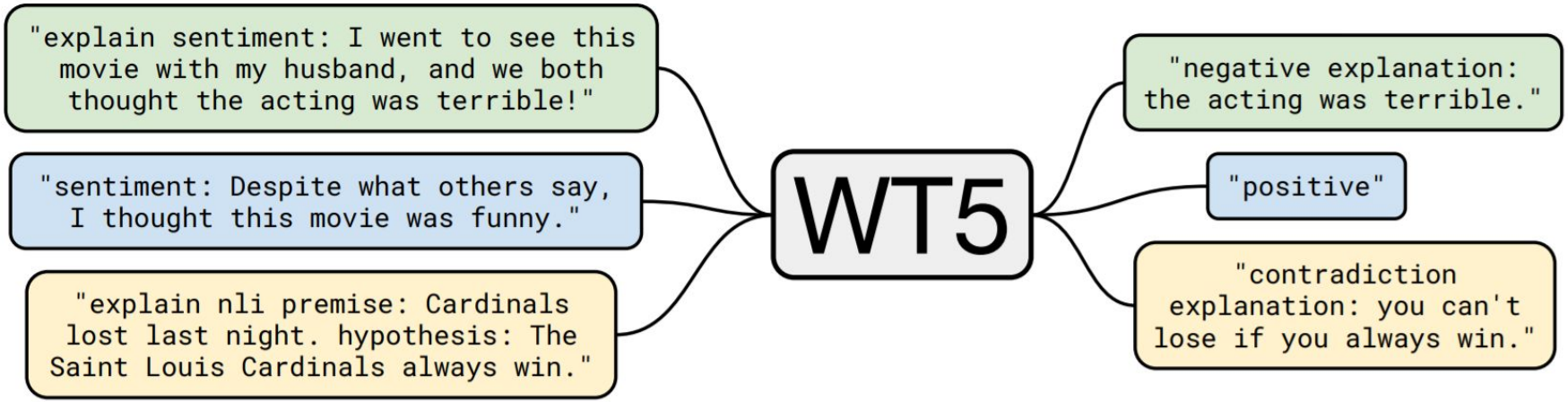
- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.

Coarse-Grained Categories	Fine-Grained Categories
Lexical Semantics	Lexical Entailment, Morphological Negation, Factivity, Symmetry/Collectivity, Redundancy, Named Entities, Quantifiers
Predicate-Argument Structure	Core Arguments, Prepositional Phrases, Ellipsis/Implicits, Anaphora/Coreference Active/Passive, Nominalization, Genitives/Partitives, Datives, Relative Clauses, Coordination Scope, Intersectivity, Restrictivity
Logic	Negation, Double Negation, Intervals/Numbers, Conjunction, Disjunction, Conditionals, Universal, Existential, Temporal, Upward Monotone, Downward Monotone, Non-Monotone
Knowledge	Common Sense, World Knowledge

[Levesque et al. 2012, Ettinger et al. 2017, Wang et al. 2018, McCoy et al. 2019, Gardner et al. 2020, Ribeiro et al. 2020, ...]

Interpretation methods

- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.
- Baking interpretability into the model

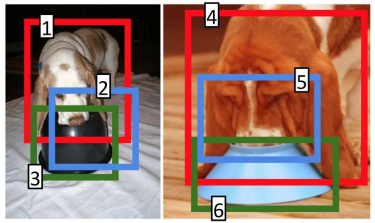


[Lei et al. 2016, Narang et al. 2020, DeYoung et al. 2020, ...]

Interpretation methods

- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.
- Baking interpretability into the model

two dogs are touching a food dish with their face



Program	Output
equal	True
count	2
with-relation [is touching]	[2, 5]
relocate [face]	[2, 5]
find [dog]	[1, 4]
find [food dish]	[3, 6]
number [two]	2

Who threw the longest touchdown pass in the second half?

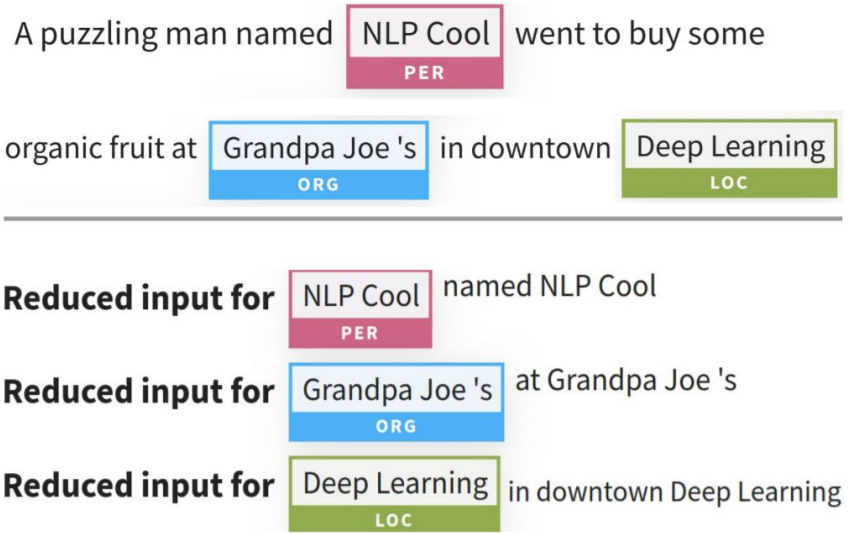
In the first quarter, the Texans trailed early after QB Kerry Collins threw a 19-yard TD pass [1] to WR Nate Washington. Second quarter started with kicker Neil Rackers made a 37-yard field goal, and the quarter closed with kicker Rob Bironas hitting a 30-yard field goal. The Texans tried to cut the lead with QB Matt Schaub getting a 8-yard TD pass [2] to WR Andre Johnson, but the Titans would pull away with RB Javon Ringer throwing a 7-yard TD pass [3]. The Texans tried to come back into the game in the fourth quarter, but only came away with Schaub [4] throwing a 12-yard TD pass [5] to WR Kevin Walter.

Program	Output
relocate[who threw]	Schaub [4]
find-max-num	[5]
filter [the second half]	[2, 3, 5]
find [touchdown pass]	[1, 2, 3, 5]

[Andreas et al. 2016, Gupta et al. 2020, Subramanian et al. 2020, ...]

Interpretation methods

- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.
- Baking interpretability into the model
- Looking at input features



Method	Saliency Map
Gradient	an intelligent fiction about learning through cultural clash.

[Ribeiro et al. 2016, Feng et al. 2018, Wallace et al. 2019]

Interpretation methods

- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.
- Baking interpretability into the model
- Looking at input features
- Looking for global decision rules

Instance	If	Predict
I want to play(V) ball.	previous word is PARTICLE	play is VERB.
I went to a play(N) yesterday.	previous word is DETERMINER	play is NOUN.
I play(V) ball on Mondays.	previous word is PRONOUN	play is VERB.

Contradiction Trigger Words

nobody
nothing
sleeps
None
sleeping
aliens

[Ribeiro et al. 2018, Wallace et al. 2019]

Interpretation methods

- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.
- Baking interpretability into the model
- Looking at input features
- Looking for global decision rules
- Looking at training examples

Test Example

Polar Bear

✘

Important Training Example

Polar Bear

✘

[Koh and Liang 2017, Yeh et al. 2018]

Interpretation methods

- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.
- Baking interpretability into the model
- Looking at input features
- Looking for global decision rules
- Looking at training examples

Interpretation methods

- **Probing internal representations**
- **Testing model behavior using challenge sets, diagnostic sets, etc.**
- Baking interpretability into the model
- Looking at input features
- Looking for global decision rules
- Looking at training examples

ACL 2020 Tutorial

Interpretation methods

- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.
- **Baking interpretability into the model** **No comprehensive resources** 😞
- Looking at input features
- Looking for global decision rules
- Looking at training examples

Interpretation methods

- Probing internal representations
- Testing model behavior using challenge sets, diagnostic sets, etc.
- Baking interpretability into the model
- **Looking at input features**
- **Looking for global decision rules**
- **Looking at training examples**

This tutorial

Why focus on these three methods?

- Interpret *specific* model predictions. Answer fine-grained questions like:
  - why did my model fail on this particular input?
  - what is the impact of this particular training point?

- Methods have desirable properties like:
  - model-agnostic
  - fast, easy-to-compute
  - faithful to underlying model*

*more on this at the end of tutorial

Tutorial Outline

~~(1) Overview of Interpretability~~

(2) What Parts of An Input Led to a Prediction?

(3) What Decision Rules Led to a Prediction?

(4) Which Training Examples Caused a Prediction?

(5) Implementing Interpretations

(6) Open Problems

Why did my model make this prediction?

Which parts of the input are responsible for this prediction?

Two classes of methods

- Saliency maps
  - generated using gradients
  - generated using perturbations

- Perturbation themselves as explanations
  - input reduction
  - adversarial perturbations

Two classes of methods

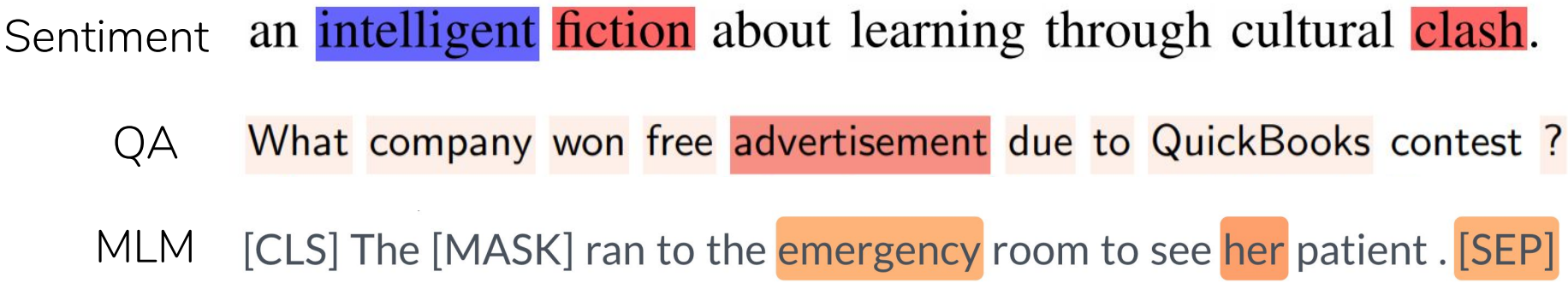
- **Saliency maps**
  - **generated using gradients**
  - **generated using perturbations**

- Perturbation themselves as explanations
  - input reduction
  - adversarial perturbations

Saliency Map Techniques in General

- Compute the relative *importance* of each token in the input
- Importance is, loosely:
  if you change or remove the token, how much is the prediction affected?

Examples of Saliency Maps:



[Ribeiro et al. 2016, Murdoch et al. 2018, Wallace et al. 2019]

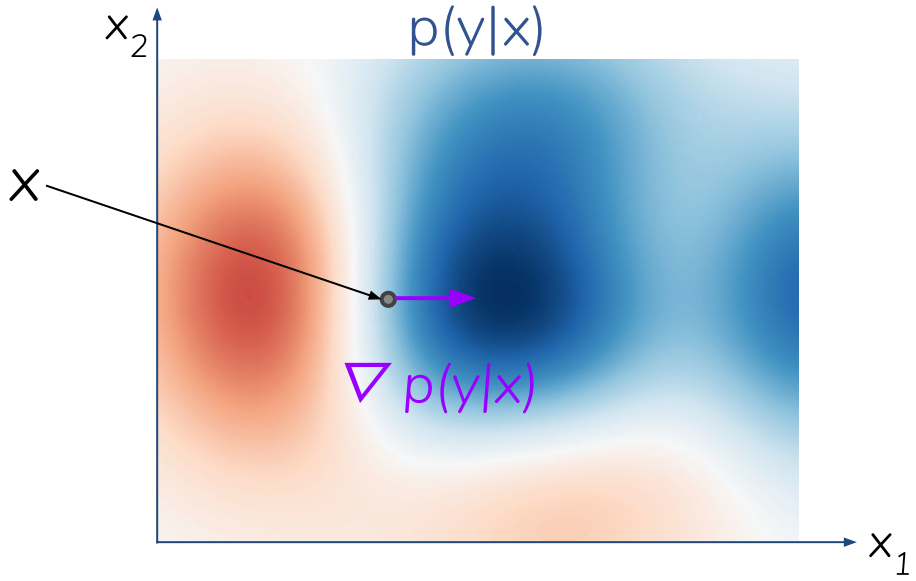
Two classes of methods

- **Saliency maps**
  - **generated using gradients**
  - generated using perturbations

- Perturbation themselves as explanations
  - input reduction
  - adversarial perturbations

Saliency Maps via Input Gradients

- Estimate importance of a feature using derivative of output w.r.t that feature
- i.e., with a “tiny change” to the feature, what happens to the prediction?



- We then visualize the importance values of each feature in a heatmap

[Simonyan et al. 2014]

Gradient-based Saliency Maps for NLP

For NLP, derivative of output w.r.t a feature

=

derivative of **output** w.r.t an **input token**

What to use as the output?

- Top prediction probability
- Top prediction logits
- Loss (with the top prediction as the ground-truth class)

What happens with multiple outputs?

- Text generation
- Tagging

Token is actually an embedding. How to turn gradient w.r.t embedding into a scalar score?

- Sum it?
- Take an L_p norm?
- Dot product with embedding itself?

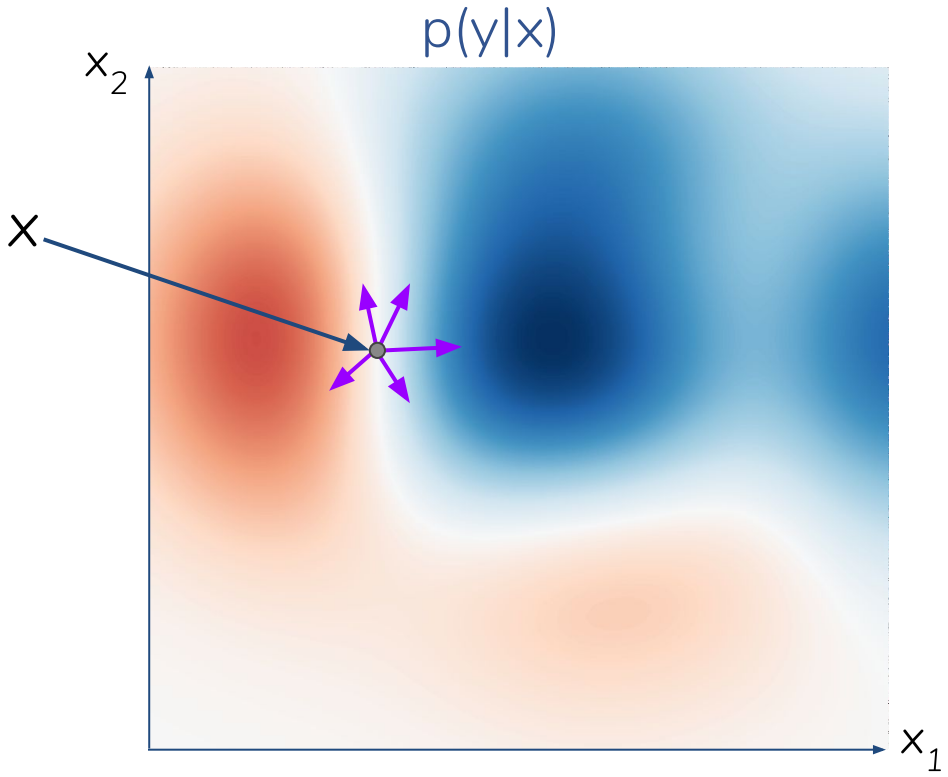
Do we normalize values across sentence?

$$-\nabla_{e(t)}\mathcal{L}_{\hat{y}} \cdot e(t)$$

Eqn from [Han et al. 2020]

Problems with Using Gradient for Saliency Maps

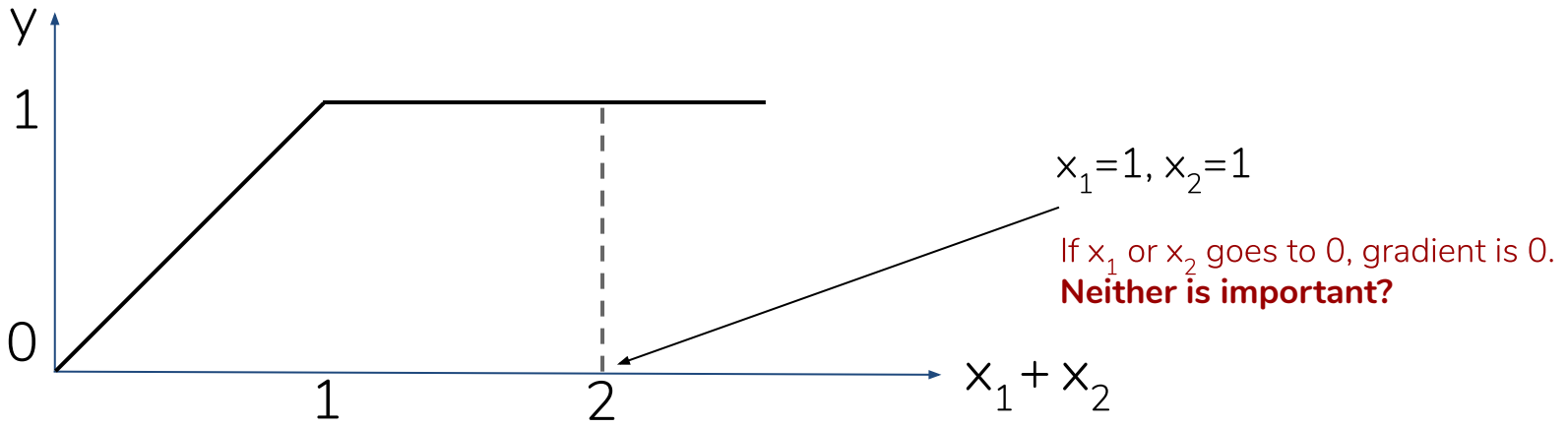
- Too “local” and thus sensitive to slight perturbations



Problems with Using Gradient for Saliency Maps

- too "local" and thus sensitive to slight perturbations
- "saturated outputs" lead to unintuitive gradients

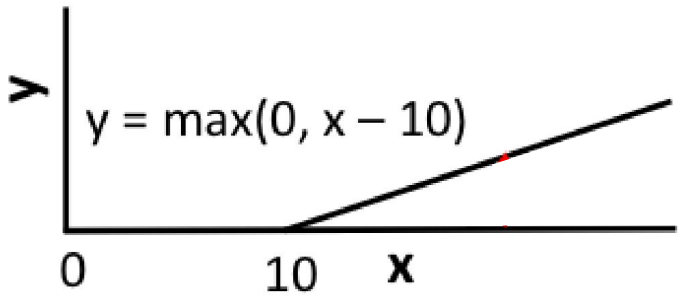
$$y = x_1 + x_2 \quad \text{when } (x_1 + x_2) < 1$$
$$1 \quad \text{when } (x_1 + x_2) >= 1$$

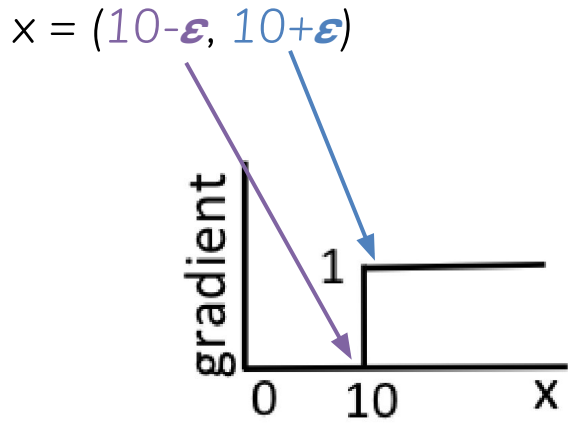


[Shrikumar et al. 2017]

Problems with Using Gradient for Saliency Maps

- too “local” and thus sensitive to slight perturbations
- “saturated outputs” lead to unintuitive gradients
- discontinuous gradients (e.g., thresholding) are problematic





[Shrikumar et al. 2017]

Extensions of Vanilla Gradient

- too “local” and thus sensitive to slight perturbations
- “saturated outputs” lead to unintuitive gradients
- discontinuous gradients (e.g., thresholding) are problematic

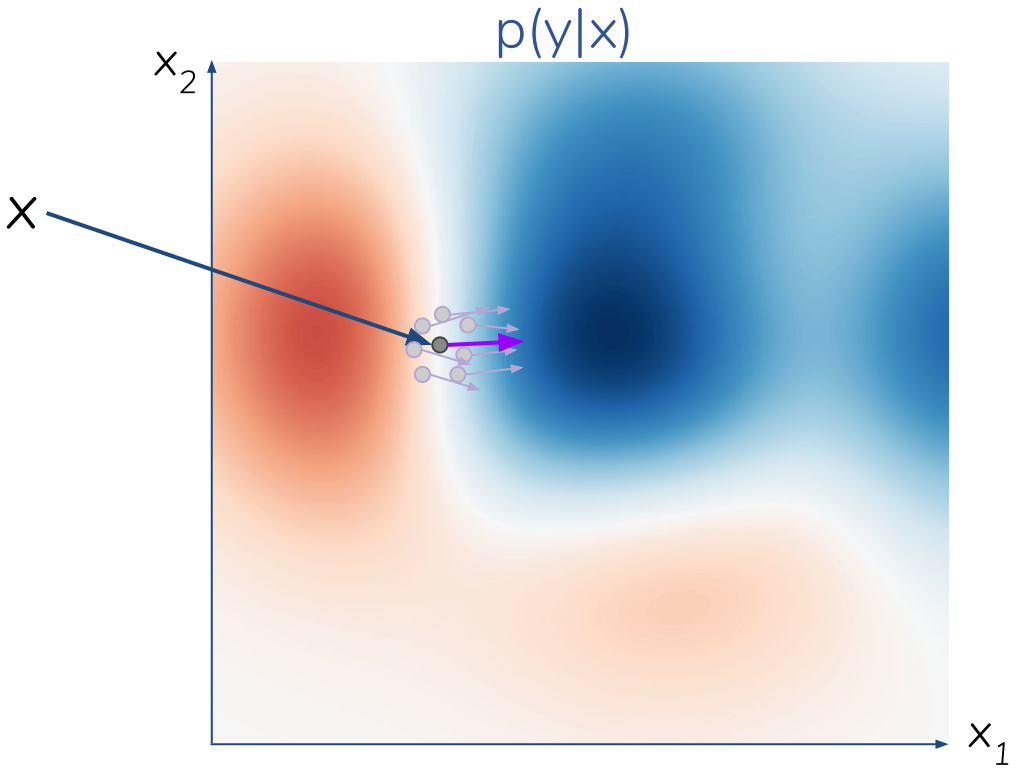
How to mitigate these issues? Don’t rely on a single gradient calculation:

- SmoothGrad
- Integrated Gradients

Other approaches, e.g., LRP, DeepLIFT, GradCAM. Not covered here.

Extensions of Vanilla Gradient

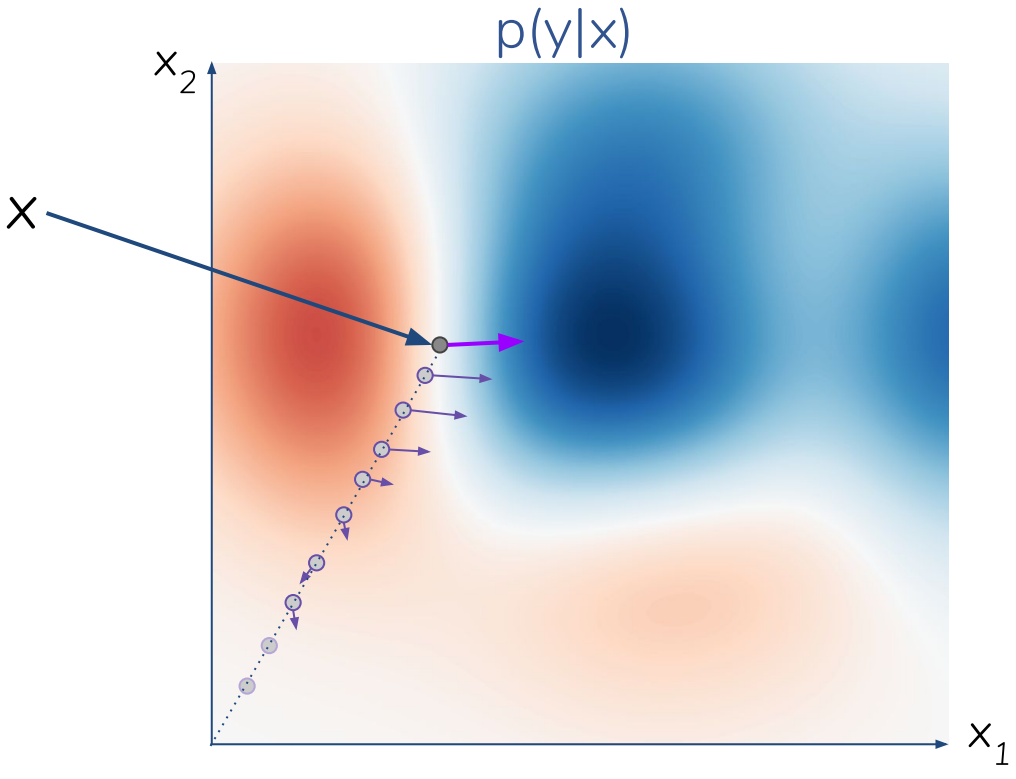
SmoothGrad: add gaussian noise to input and average the gradient



[Smilkov et al. 2017]

Extensions of Vanilla Gradient

Integrated Gradients: average gradients along path from zero to input



[Sundararajan et al. 2017]

Summary of Gradient-based Saliency Methods

Positives:

- Fast to compute: single (or a few) calls to backward()
- Visually appealing: spectrum of importance values

Negatives:

- Needs white-box (gradient) access to the model
- Not “customizable”
  - small changes in a individual “token” are not necessarily meaningful
  - distance is implicitly Euclidean (L_2)
- Gradients can be unintuitive with saturated or thresholded values
- Difficult to apply to non-classification tasks

Two classes of methods

- Saliency maps
  - generated using gradients
  - **generated using perturbations**

- Perturbation themselves as explanations
  - input reduction
  - adversarial perturbations

Alternative: Generating Saliency Maps Using Input Perturbations

Goal is the same: saliency map over the input

However, these methods:

- are black-box (completely model-agnostic)
- allow input perturbations/neighborhoods to be defined
  i.e., we can use different “units of explanations”
  - words, phrases, sentences
  - for multimodal inputs: image, environment, etc.
  - what perturbations are valid? do we enforce grammaticality?

[Ribeiro et al. 2016]

Leave-one-out: remove tokens and look at confidence

- Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

Question	Confidence	Highlight
What did Tesla spend Astor's money on ?	**0.78**	

[Li et al. 2017]

Leave-one-out: remove tokens and look at confidence

- Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

Question	Confidence	Highlight
What did Tesla spend Astor's money on ?	**0.78**	
~~What~~ did Tesla spend Astor's money on ?	0.67	What

[Li et al. 2017]

Leave-one-out: remove tokens and look at confidence

- Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

Question	Confidence	Highlight
What did Tesla spend Astor's money on ?	**0.78**	
~~What~~ did Tesla spend Astor's money on ?	0.67	What
What ~~did~~ Tesla spend Astor's money on ?	0.72	did

[Li et al. 2017]

Leave-one-out: remove tokens and look at confidence

- Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

Question	Confidence	Highlight
What did Tesla spend Astor's money on ?	**0.78**	
~~What~~ did Tesla spend Astor's money on ?	0.67	What
What ~~did~~ Tesla spend Astor's money on ?	0.72	did
What did ~~Tesla~~ spend Astor's money on ?	0.66	Tesla
What did Tesla ~~spend~~ Astor's money on ?	0.74	spend
What did Tesla spend ~~Astor's~~ money on ?	0.76	Astor's
What did Tesla spend Astor's ~~money~~ on ?	**0.48**	money
What did Tesla spend Astor's money ~~on~~ ?	0.72	on
What did Tesla spend Astor's money on ~~?~~	0.73	?

[Li et al. 2017]

Leave-one-out: remove tokens and look at confidence

- Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

Question	Confidence	Highlight
What did Tesla spend Astor's money on ?	**0.78**	
~~What~~ did Tesla spend Astor's money on ?	0.67	What
What ~~did~~ Tesla spend Astor's money on ?	0.72	did
What did ~~Tesla~~ spend Astor's money on ?	0.66	Tesla
What did Tesla ~~spend~~ Astor's money on ?	0.74	spend
What did Tesla spend ~~Astor's~~ money on ?	0.76	Astor's
What did Tesla spend Astor's ~~money~~ on ?	**0.48**	money
What did Tesla spend Astor's money ~~on~~ ?	0.72	on
What did Tesla spend Astor's money on ~~?~~	0.73	?

What did Tesla spend Astor's money on ?

[Li et al. 2017]

Concern with Leave-one-out

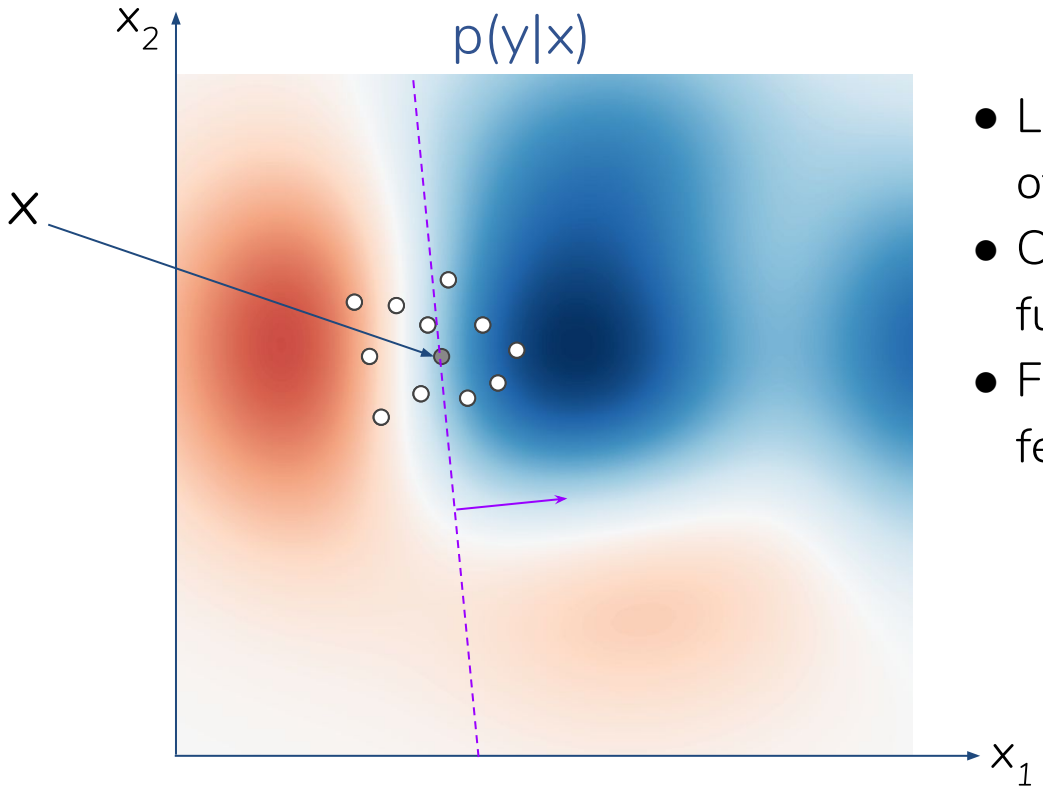
Problem: Leave-**ONE**-out

The movie is mediocre, maybe even bad.	**Negative** 99.8%
The movie is mediocre, maybe even ~~bad~~.	**Negative** 98.0%
The movie is ~~mediocre~~, maybe even bad.	**Negative** 98.7%

What we really need:

The movie is ~~mediocre~~, maybe even ~~bad~~.	**Positive** 63.4%

LIME: Intuition Behind the Approach



- Look at model's predictions for a bunch of nearby inputs
- Closer points are more important than further points
- Fit a linear model. Its weights are the feature importances

[Ribeiro et al. 2016]

LIME Sentiment Analysis Example

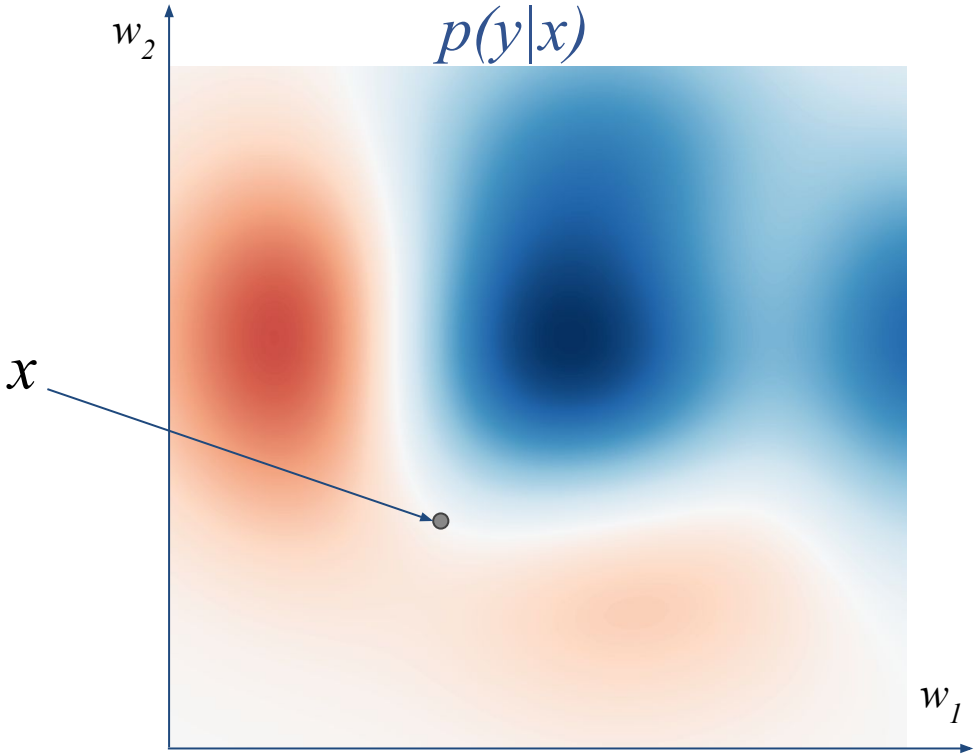
The movie is mediocre, maybe even bad.	**Negative** 99.8%
The movie is mediocre, maybe even ~~bad~~.	**Negative** 98.0%
The movie is ~~mediocre~~, maybe even bad.	**Negative** 98.7%
The movie is ~~mediocre~~, maybe even ~~bad~~.	**Positive** 63.4%
The movie is ~~mediocre~~, ~~maybe~~ even ~~bad~~.	**Positive** 74.5%
The ~~movie~~ is mediocre, maybe even ~~bad~~.	**Negative** 97.9%

The movie is mediocre, maybe even bad.

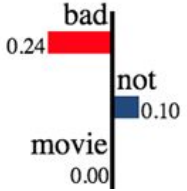
Problems with Leave-one-out and LIME

- Customizing perturbations and distances is difficult to define intuitively
- More importantly, difficult for users to understand
    - different explanations with different perturbations?
- How do we define “distance” between sentences? Lexical similarity?
- Generating interpretations is expensive (many calls to underlying model)
- Difficult to apply to non-classification tasks

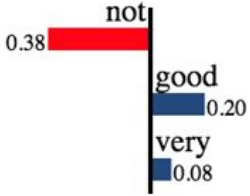
Problem with Saliency Maps:
(linear) representations can be quite limited



+ This movie is not bad.



− This movie is not very good.



Two classes of methods

- Saliency maps
  - generated using gradients
  - generated using perturbations

- **Perturbation themselves as explanations**
  - **input reduction**
  - **adversarial perturbations**

Perturbation-Based Explanations

Goal: Which parts of the input are important?

Instead of “soft” attributions, let’s talk in terms of changes to the input

Examples of perturbation-based explanations:

- here’s how much you can remove without changing predictions
- here’s a similar example with a different prediction

Two classes of methods

- Saliency maps
  - generated using gradients
  - generated using perturbations

- Perturbation themselves as explanations
  - **input reduction**
  - adversarial perturbations

Input Reduction

- What if we remove the **least** important words?
- But, keep the model prediction the **same**

[Feng et al. 2018]

Input Reduction

- What if we remove the **least** important words?
- But, keep the model prediction the **same**

Question	Confidence
What did Tesla ~~spend~~ Astor's money on ?	0.78

[Feng et al. 2018]

Input Reduction

- What if we remove the **least** important words?
- But, keep the model prediction the **same**

Question	Confidence
What did Tesla ~~spend~~ Astor's money on ?	0.78
What did Tesla Astor's ~~money~~ on ?	0.74

[Feng et al. 2018]

Input Reduction

- What if we remove the **least** important words?
- But, keep the model prediction the **same**

Question								Confidence
What	did	Tesla	~~spend~~	Astor's	money	on	?	0.78
What	did	Tesla		Astor's	~~money~~	on	?	0.74
What	did	Tesla		Astor's		~~on~~	?	0.76
~~What~~	did	Tesla		Astor's			?	0.80
	did	Tesla		Astor's			~~?~~	0.87

[Feng et al. 2018]

Input Reduction

- What if we remove the **least** important words?
- But, keep the model prediction the **same**

Question								Confidence
What	did	Tesla	~~spend~~	Astor's	money	on	?	0.78
What	did	Tesla		Astor's	~~money~~	on	?	0.74
What	did	Tesla		Astor's		~~on~~	?	0.76
~~What~~	did	Tesla		Astor's			?	0.80
	did	Tesla		Astor's			~~?~~	0.87
	did	~~Tesla~~		Astor's				0.82
	did			~~Astor's~~				0.89
	did							0.91

[Feng et al. 2018]

Input Reduction

- What if we remove the **least** important words?
- But, keep the model prediction the **same**

Question								Confidence
What	did	Tesla	~~spend~~	Astor's	money	on	?	0.78
What	did	Tesla		Astor's	~~money~~	on	?	0.74
What	did	Tesla		Astor's		~~on~~	?	0.76
~~What~~	did	Tesla		Astor's			?	0.80
	did	Tesla		Astor's			~~?~~	0.87
	did	~~Tesla~~		Astor's				0.82
	did			~~Astor's~~				0.89
	did							0.91

Prediction remains the same.

[Feng et al. 2018]

Input Reduction Examples

SQuAD	
Context	In 1899, John Jacob Astor IV invested $100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his Colorado Springs experiments.
Original	What did Tesla spend Astor's money on ?
Reduced	**did**
Confidence	0.78 → 0.91

[Feng et al. 2018]

Input Reduction Examples

SQuAD

Context	In 1899, John Jacob Astor IV invested $100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his Colorado Springs experiments.
Original	What did Tesla spend Astor's money on ?
Reduced	**did**
Confidence	0.78 → 0.91

VQA

Original	What color is the flower ?
Answer	yellow
Reduced	**flower ?**
Confidence	0.827 → 0.819

[Feng et al. 2018]

Input Reduction Examples

SQuAD

Context	In 1899, John Jacob Astor IV invested $100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his Colorado Springs experiments.
Original	What did Tesla spend Astor's money on ?
Reduced	**did**
Confidence	0.78 → 0.91

VQA

Original	What color is the flower ?
Answer	yellow
Reduced	**flower ?**
Confidence	0.827 → 0.819

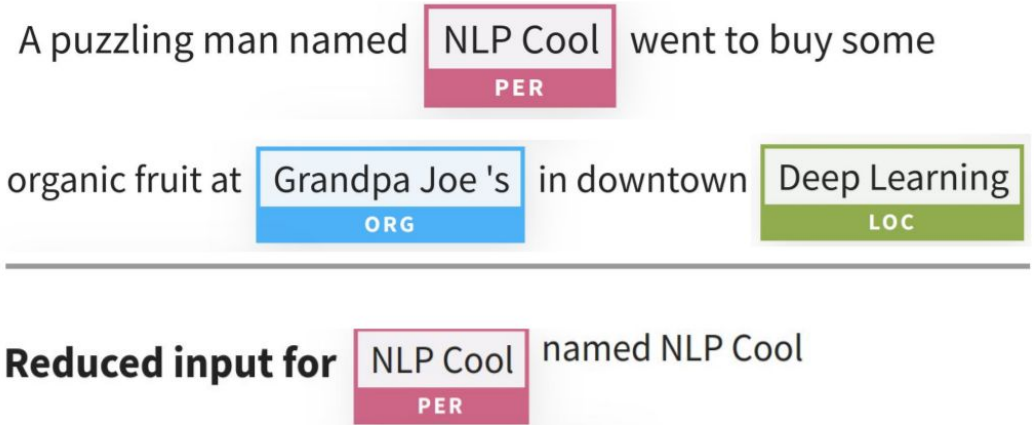
SNLI

Premise	Well dressed man and woman dancing in the street
Original	Two man is dancing on the street
Answer	Contradiction
Reduced	**dancing**
Confidence	0.977 → 0.706

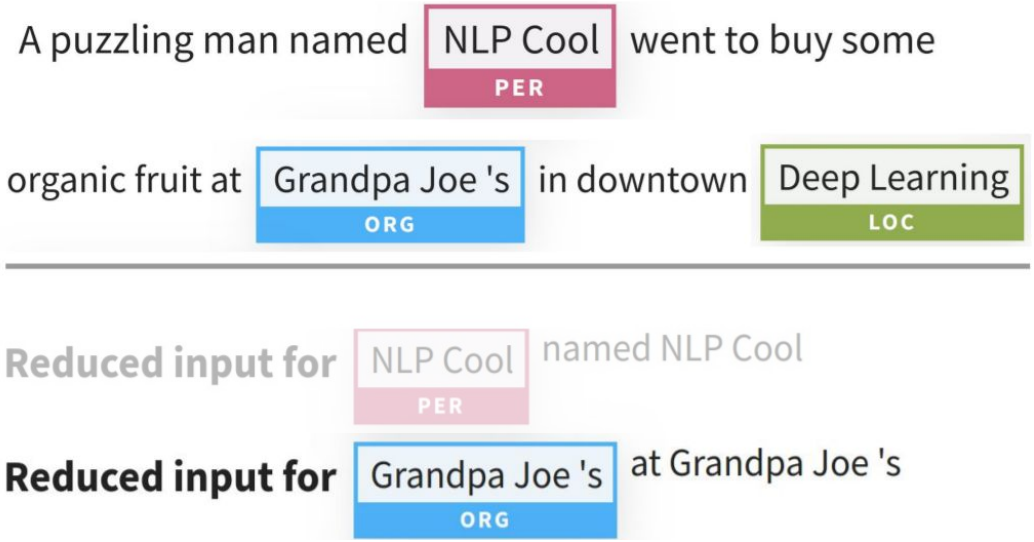
[Feng et al. 2018]

Input Reduction as Explanations

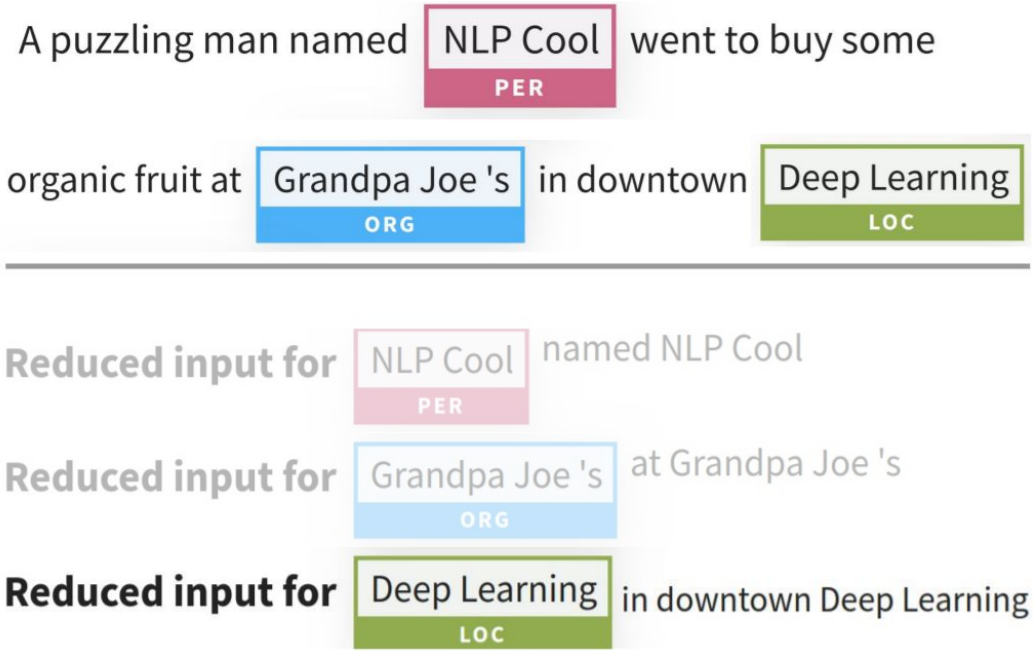
Input Reduction as Explanations



Input Reduction as Explanations



Input Reduction as Explanations



Debugging NLVR2 Models With Input Reduction

Label: True

Original Instance: "[CLS] the left and right image contains no more than three bottles of lot ##ion. [SEP]"

[Subramanian et al. 2020]

Debugging NLVR2 Models With Input Reduction

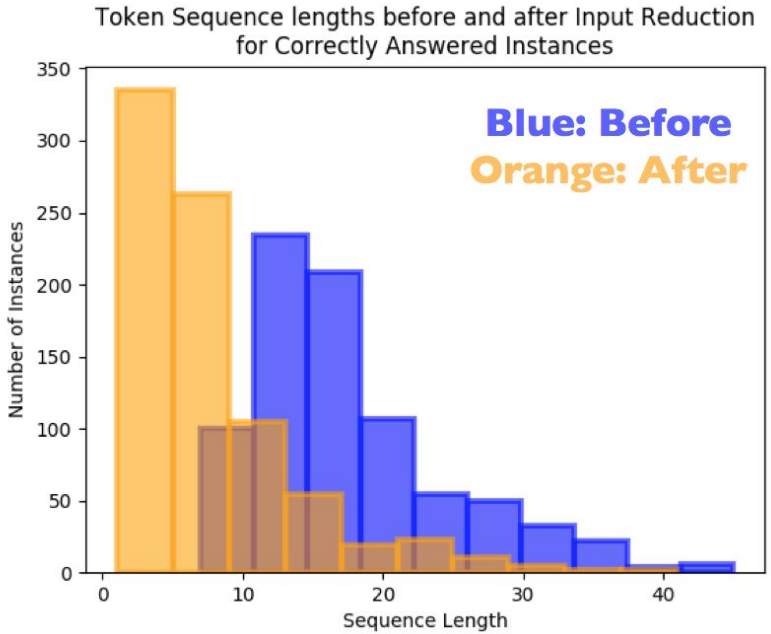
Label: True

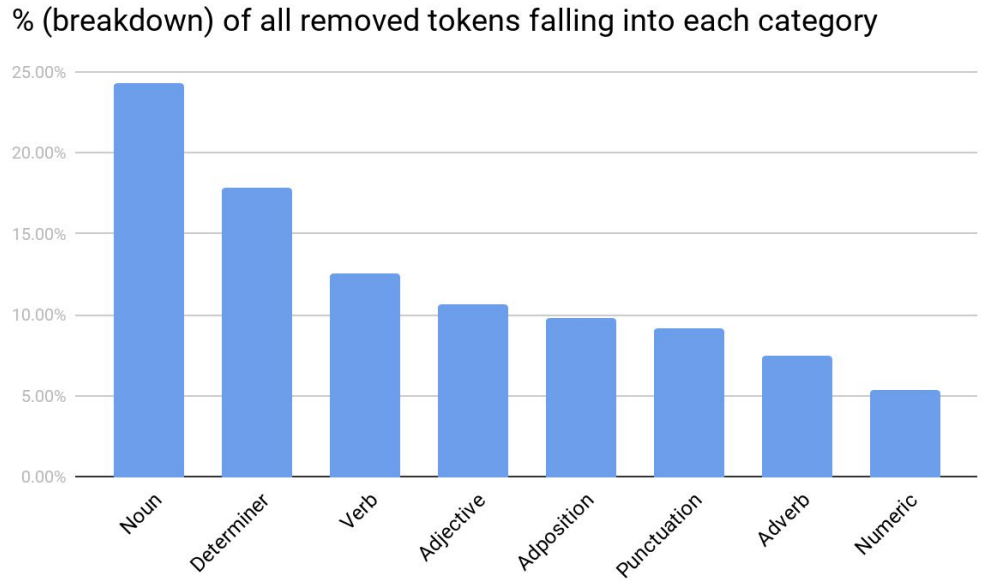
Original Instance: "[CLS] the left and right image contains no more than three bottles of lot ##ion. [SEP]"

Reduced Instance: "[CLS] ~~the~~ ~~left~~ and right ~~image~~ ~~contains~~ no more ~~than~~ ~~three~~ bottles ~~of~~ lot ~~##ion~~. [SEP]"

[Subramanian et al. 2020]

Debugging NLVR2 Models With Input Reduction





[Subramanian et al. 2020]

Debugging NLVR2 Models With Input Reduction

Before: an image shows a man sitting in front of a computer screen
After: **man sitting screen**

Before: at least one human is wearing eye glasses
After: **eye**

Before: exactly three white ducks are standing in a row on dry ground
After: **exactly three white ducks row**

[Subramanian et al. 2020]

Problems with Input Reduction

- The input gets heavily modified, is it still an explanation of the original input?
- Humans often do not understand results (and thus cannot improve model)
- Search is hard: the gradient changes considerably as the input changes
    - many different “reduced inputs” exist

Pose a different problem: find the nearest example with a different prediction

(sounds a lot like **adversarial examples**!)

Two classes of methods

- Saliency maps
  - generated using gradients
  - generated using perturbations

- Perturbation themselves as explanations
  - input reduction
  - **adversarial perturbations**

Adversarial Examples in NLP

In the United States especially, several high-profile cases such as Debra LaFave, Pamela Rogers, and Mary Kay Letourneau have caused increased scrutiny on teacher misconduct.

Original	What has been the result of this publicity?	increased scrutiny on teacher misconduct
HotFlip [Ebrahimi et al. 2018]	What haL been the result of this publicity?	teacher misconduct
SEARs [Ribeiro et al. 2018]	What's been the result of this publicity?	teacher misconduct
SCPN (Iyyer et al. 2018)	The result of this publicity is what?	teacher misconduct

Adversarial Examples in NLP

In the United States especially, several high-profile cases such as Debra LaFave, Pamela Rogers, and Mary Kay Letourneau have caused increased scrutiny on teacher misconduct. *Hurricane Harvey resulted in at least 107 deaths.*

Original	What has been the result of this publicity?	increased scrutiny on teacher misconduct
HotFlip [Ebrahimi et al. 2018]	What haL been the result of this publicity?	teacher misconduct
SEARs [Ribeiro et al. 2018]	What's been the result of this publicity?	teacher misconduct
SCPN [Iyyer et al. 2018]	The result of this publicity is what?	teacher misconduct
AddSent [Jia and Liang 2017]	What has been the result of this publicity?	at least 107 deaths

Adversarial Examples as Explanations?

The doctor ran to the emergency room to see [MASK] patient.

Mask 1 Predictions:
38.3% his
36.9% the
8.1% another

The **woman** ran to the emergency room to see [MASK] patient.

Mask 1 Predictions:
44.9% her
30.4% the
12.6% a

all the amped up tony hawk style stunts and thrashing rap-metal can't disguise the fact that, really, we've been here, done that.

Positive. (69.2%)

all the amped up **cookie-cutter** hawk style stunts and thrashing rap-metal can't disguise the fact that, really, we've been here, done that.

Negative. (67.9%)

Problems with Perturbation Explanations

- Perturbations often lead to **nonsensical inputs**
  - humans can't understand results (and thus cannot improve model)
  - does accuracy matter on these inputs?

- Perturbations are often **overly specific** to the particular example
  - does the explanation/perturbation apply more generally?
  - (next section describes universally-applicable perturbations)

- Practically, need to intuitively **"explain" the interpretation**
  - users have to imagine how explanation was generated
  - e.g., how did the search work? what was the perturbation function?

Which Part of the Input? Two Types of Methods

- Saliency maps
- Perturbation as explanations

These are explanations that focus on how **the input** affects the output:

- Efficient, flexible and extensible
- The interpretation is intuitive and natural

Biggest concerns:

- Very specific in its focus
  - what's the bigger picture? What can you take away from these?
  - if there's a problem, is it that one example or a bigger issue?
- Insights are rarely actionable
  - if I find a problem, how do I fix it?
  - data processing? errors in training data? overfitting?

Tutorial Outline

~~(1) Overview of Interpretability~~

~~(2) What Parts of An Input Led to a Prediction?~~

(3) What Decision Rules Led to a Prediction?

(4) Which Training Examples Caused a Prediction?

(5) Implementing Interpretations

(6) Open Problems

Why did my model make this prediction?

What decision rules led to this prediction?

What Decision Rule Led to a Prediction?

- Past interpretations are “local”
  - How can we move towards more “global” explanations?
- Can we extract **decision rules** that approximate a model’s predictions?
- Loose definition: “if pattern x holds, model typically makes prediction y.”
  - e.g., if NLI hypothesis contains “nobody”, then predict Contradiction 98% of the time.
- A simplification: don’t identify a complete set of rules that you can “run”
  - Instead, find rules that only cover a tiny part of the input space

[Lakkaraju et al. 2017, Guidotti et al. 2018, Tan et al. 2018]

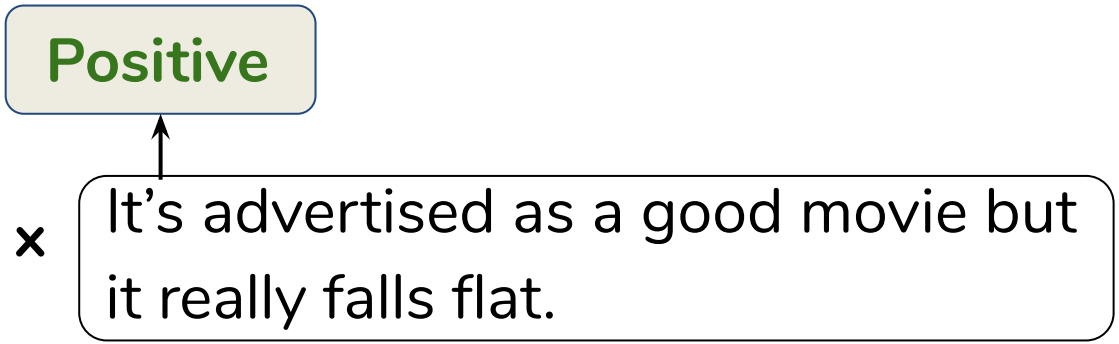
Finding Decision Rules

We will cover two methods for global explanations:

- Anchors [Ribeiro et al. 2018]
- Universal Adversarial Triggers [Wallace et al. 2019]

Also see [Sushil et al. 2018, Ribeiro et al. 2018, Li et al. 2019]

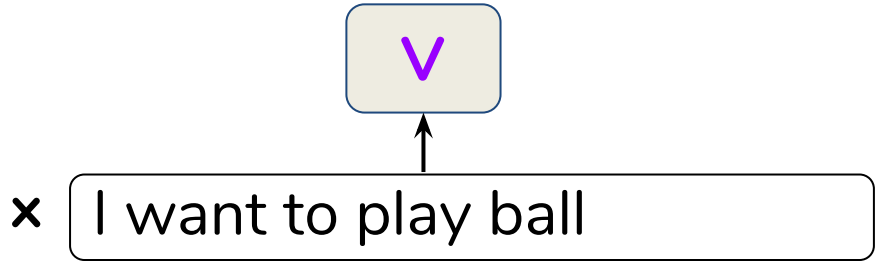
What are Anchors?



Anchor

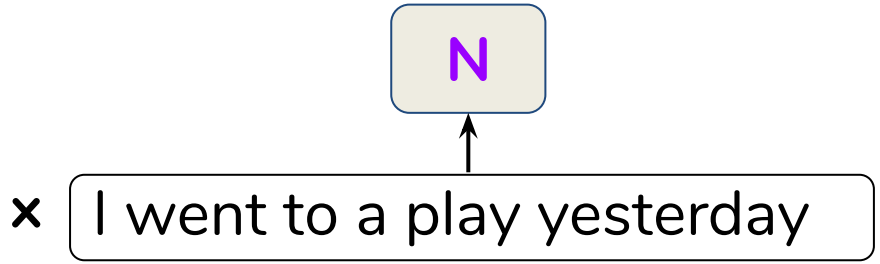
If "good" and "movie": predict Positive

What are Anchors?



Anchor

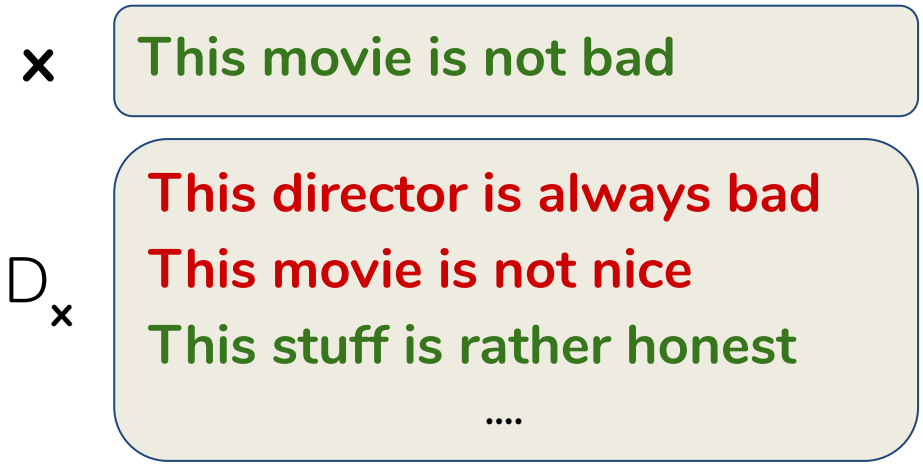
If previous word is PARTICLE:
predict VERB



If previous word is DETERMINER:
predict NOUN

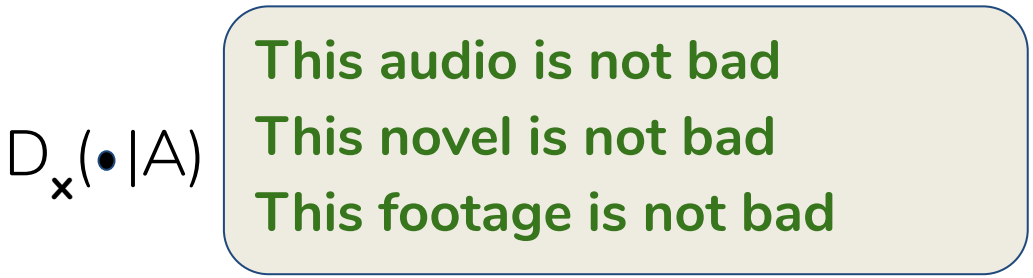
If previous word is PRONOUN:
predict VERB

Concrete Definition of Anchors



A = {'not', 'bad'}

Goals for A:



big $D_x(\bullet|A)$ (high coverage)

lots of **green** (high precision)

Computing Anchors

× **This movie is not bad**

Possible Anchors = {This},
{movie},
{is},
{not},
{bad},
{This, movie},
{This, is}
.....

$2^{|FEATURES|}$ possible Anchors

Computing Anchors

× **This movie is not bad**

(1) Consider single element rules

{This} {movie} {is} {not} {bad}

Computing Anchors

× **This movie is not bad**

(1) Consider single element rules

{This} {movie} {is} {not} {bad}

(2) Get samples from $D_{\times}(\bullet|A)$, run model, and estimate precision

Computing Anchors

× **This movie is not bad**

(1) Consider single element rules

{This} {movie} {is} {not} {bad}

(2) Get samples from $D_{\times}(\bullet|A)$, run model, and estimate precision

{This}	{movie}	{is}	{**not**}	{bad}
51%	55%	48%	**80%**	20%

Computing Anchors

x **This movie is not bad**

(1) Consider single element rules

{This} {movie} {is} {not} {bad}

(2) Get samples from $D_x(\bullet|A)$, run model, and estimate precision

{This}	{movie}	{is}	{**not**}	{bad}
51%	55%	48%	**80%**	20%

(3) Choose highest precision rule. Next, consider two element rules

....

Computing Anchors

× **This movie is not bad**

(1) Consider single element rules

{This} {movie} {is} {not} {bad}

(2) Get samples from $D_{\times}(\bullet|A)$, run model, and estimate precision

{This}	{movie}	{is}	{**not**}	{bad}
51%	55%	48%	**80%**	20%

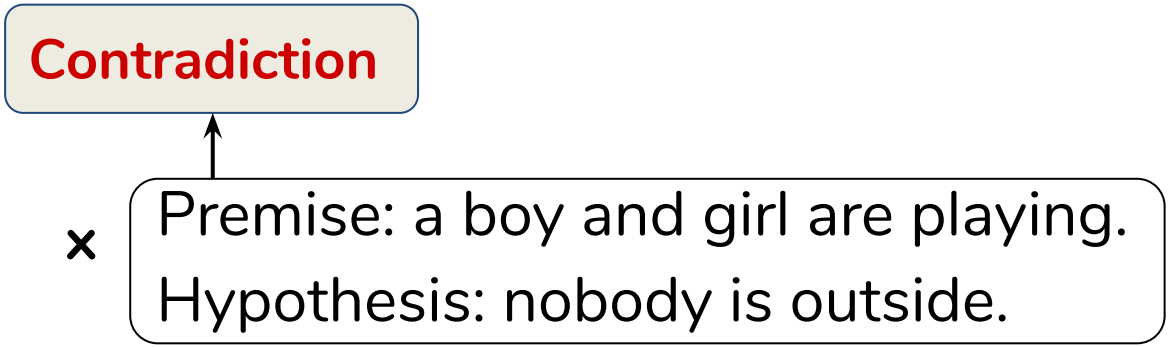
(3) Choose highest precision rule. Next, consider two element rules

....

(4) Return anchor when precision > 95%

{not, bad} precision = 96%

Use Case of Debugging With Anchors



Anchor

If "nobody" in hypothesis: predict Contradiction

Universal Adversarial Triggers

Universal Adversarial Triggers

Find a phrase that, if inserted into any input, would cause prediction y.

Universal Adversarial Triggers

Find a phrase that, if inserted into any input, would cause prediction y.

- Typically ungrammatical, although still insightful. See [Song et al. 2020, Atanasova et al. 2020, Song et al. 2020] for grammaticality.

Universal Adversarial Triggers

Find a phrase that, if inserted into any input, would cause prediction y.

- Typically ungrammatical, although still insightful. See [Song et al. 2020, Atanasova et al. 2020, Song et al. 2020] for grammaticality.

Inputs

This movie is amazing!

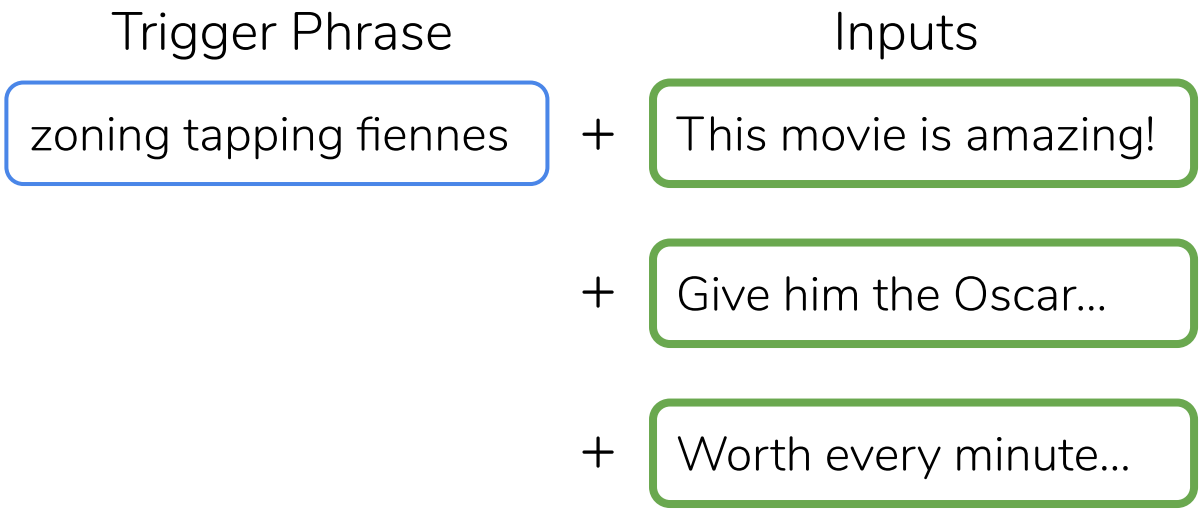
Give him the Oscar...

Worth every minute...

Universal Adversarial Triggers

Find a phrase that, if inserted into any input, would cause prediction y.

- Typically ungrammatical, although still insightful. See [Song et al. 2020, Atanasova et al. 2020, Song et al. 2020] for grammaticality.



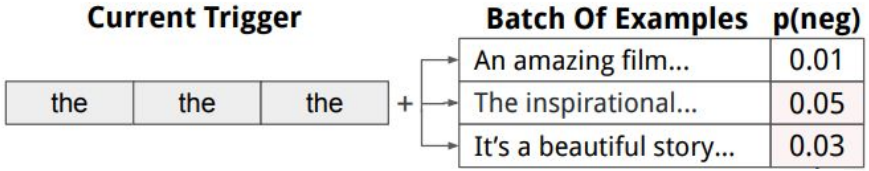
Universal Adversarial Triggers

Find a phrase that, if inserted into any input, would cause prediction y.

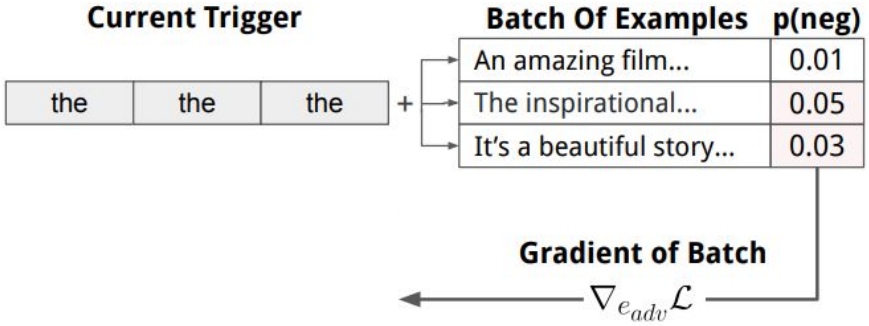
- Typically ungrammatical, although still insightful. See [Song et al. 2020, Atanasova et al. 2020, Song et al. 2020] for grammaticality.

Trigger Phrase		Inputs	Prediction
zoning tapping fiennes	+	This movie is amazing!	Positive ⇨ Negative
	+	Give him the Oscar...	Positive ⇨ Negative
	+	Worth every minute...	Positive ⇨ Negative

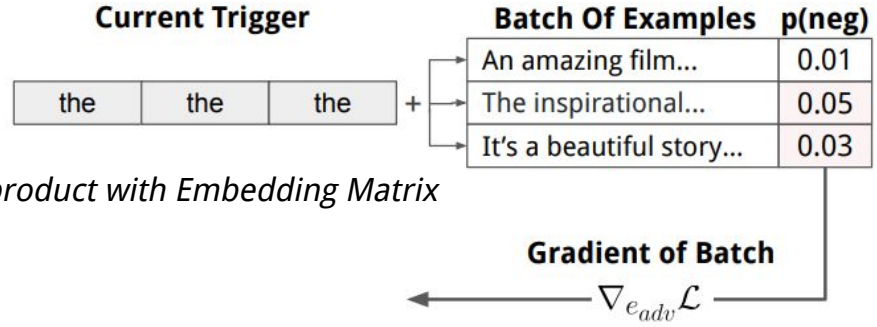
Generating Triggers



Generating Triggers

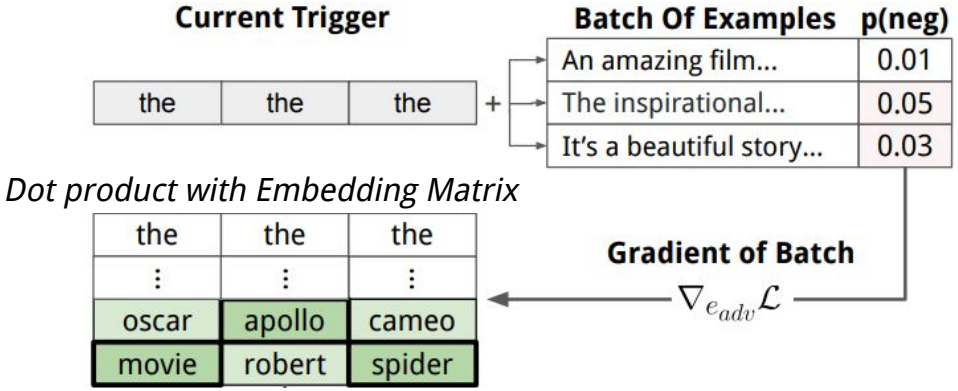


Generating Triggers

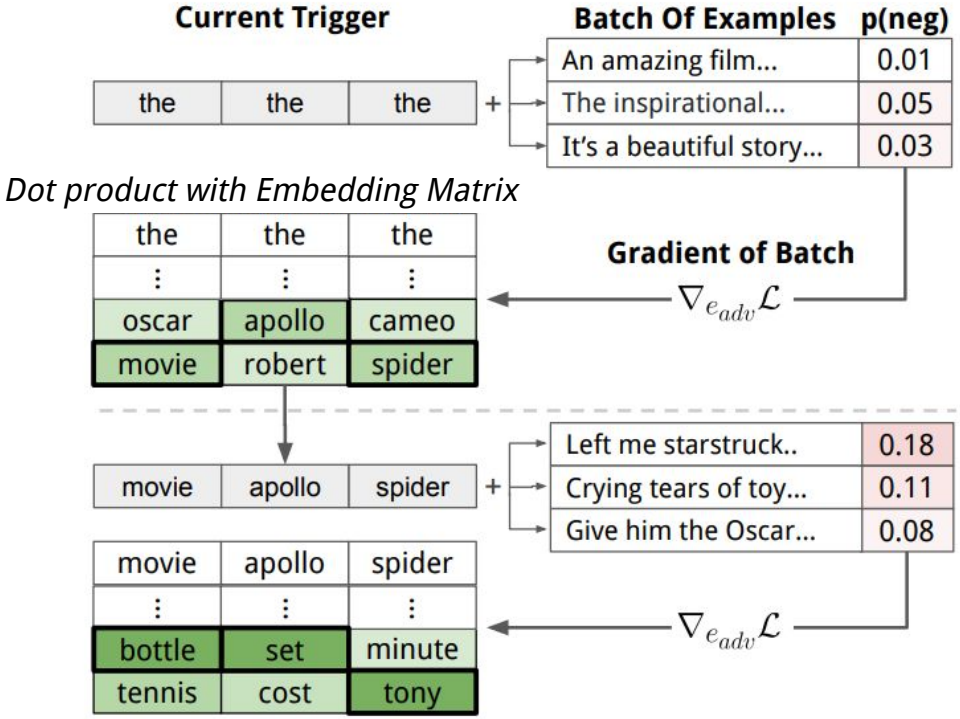


Dot product with Embedding Matrix

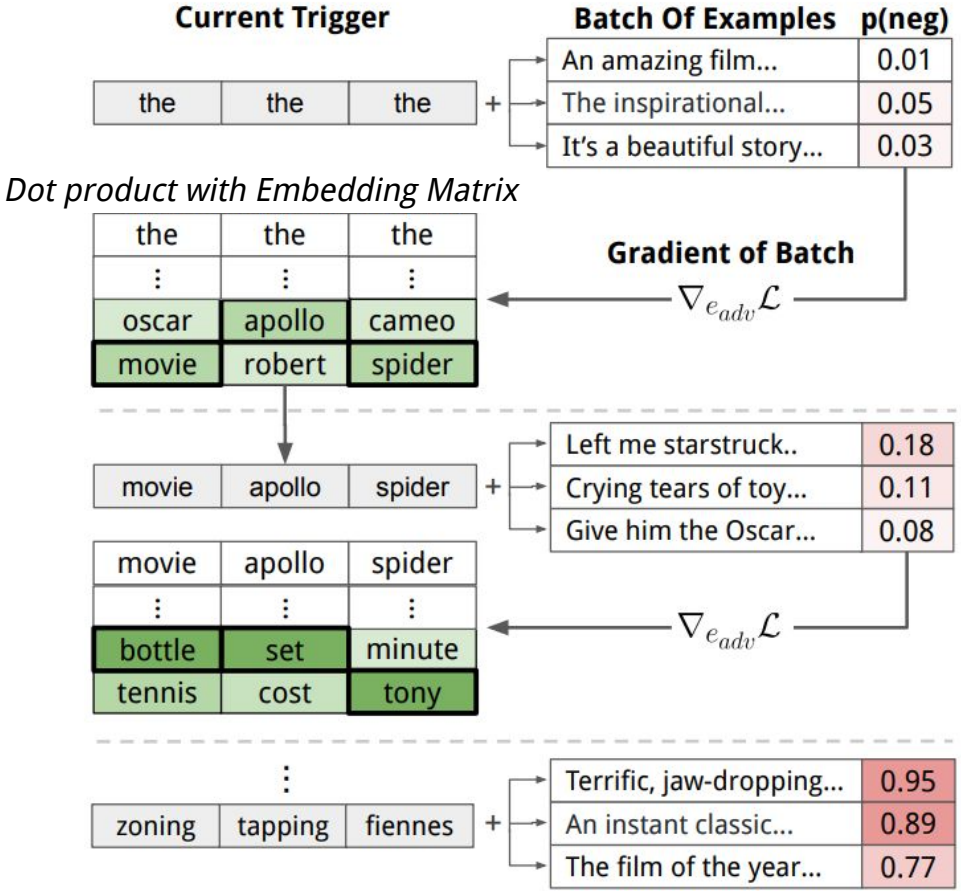
Generating Triggers



Generating Triggers



Generating Triggers



Debugging NLI With Triggers

- Prepend trigger to SNLI hypothesis (Bowman et al. 2015)
- Take examples where model predicts Entailment, try to change prediction to Contradiction

Debugging NLI With Triggers

- Prepend trigger to SNLI hypothesis (Bowman et al. 2015)
- Take examples where model predicts Entailment, try to change prediction to Contradiction

Trigger Phrase	% Contradiction
nobody	99.6%
never	99.1%
sad	99.1%
scared	98.6%

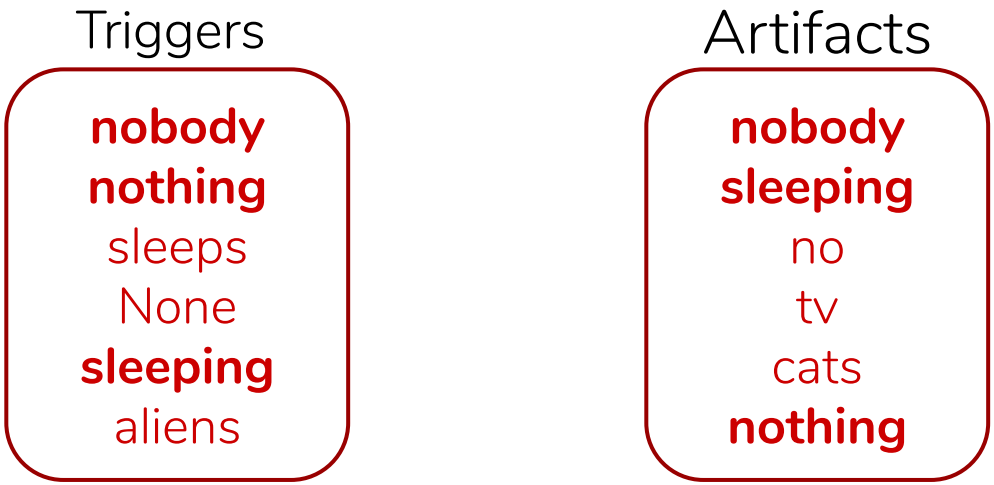
Premise: a boy and girl are playing.

Hypothesis: **nobody** two people are playing outside.

Debugging NLI With Triggers

- Why does adding “nobody” to the hypothesis change most predictions to contradiction?
- *Annotation artifacts*. SNLI has words spuriously correlated with the label [Gururangan et al. 2018, Poliak et al. 2018]
- Artifacts quantified via a word-label correlation analysis

Debugging NLI With Triggers



- Triggers align with artifacts---interpretability can automatically discover dataset biases

Debugging SQuAD with Triggers

- Neural QA models for SQuAD (Rajpurkar et al. 2016)
- Group questions by ‘wh’ word (“who”, “what”, “why”, …)
- Insert trigger phrase into input paragraph
- Trigger a specific type of span answer (e.g., dates)

Debugging SQuAD with Triggers

Where?

where CITY NAME where where where

Debugging SQuAD with Triggers

Where?

where CITY NAME where where where

Where was Evan Washburn during the game?

The broadcast team of Jim Nantz and Phil Simms called the contest, with Tracy Wolfson and Evan Washburn on the sidelines. CBS introduced new features during the telecast, including pylon cameras and microphones along with EyeVision 360—an array of 36 cameras that provides a 360-degree view of plays.

Debugging SQuAD with Triggers

Where?

where CITY NAME where where where

Where was Evan Washburn during the game?

The broadcast team of Jim Nantz and Phil Simms called the contest, with Tracy Wolfson and Evan Washburn on the sidelines. CBS introduced new features during the telecast, including pylon cameras and microphones along with EyeVision 360—an array of 36 cameras that provides a 360-degree view of plays. where new york where where where.

Debugging SQuAD with Triggers

Where?

where CITY NAME where where where

51% of predictions = CITY NAME

Where was Evan Washburn during the game?

The broadcast team of Jim Nantz and Phil Simms called the contest, with Tracy Wolfson and Evan Washburn on the sidelines. CBS introduced new features during the telecast, including pylon cameras and microphones along with EyeVision 360—an array of 36 cameras that provides a 360-degree view of plays. where new york where where where.

Debugging SQuAD with Triggers

Type	Trigger Phrase	
Why?	why how because of Noun Phrase	(e.g., pineapple pizza)
Who?	population ; NAME : who who who	(e.g., Michael Jordan)
When?	; its time about DATE when	(e.g., January, 2014)

Debugging SQuAD with Triggers

- Triggers reveal models leverage biases w.r.t question types

Debugging SQuAD with Triggers

- Triggers reveal models leverage biases w.r.t question types
  - Lexical overlap with question

Who?	population ; NAME : who who who
Where?	where CITY NAME where where where

Debugging SQuAD with Triggers

- Triggers reveal models leverage biases w.r.t question types
  - Lexical overlap with question

Who?	population ; NAME : who who who
Where?	where CITY NAME where where where

  - Local context bias

When?	; its time about DATE when
Why?	why how because of Noun Phrase

Debugging SQuAD with Triggers

- Triggers reveal models leverage biases w.r.t question types
  - Lexical overlap with question

Who?	population ; NAME : who who who
Where?	where CITY NAME where where where

  - Local context bias

When?	; its time about DATE when
Why?	why how because of Noun Phrase

- Identified *manually* in past work [Sugawara et al. 2018, Jia and Liang 2017]

Pros/Cons of Interpretation Via Decision Rules

- Can **identify** global bugs in models + datasets (e.g., annotation artifacts)
- Often hard to find **broad-coverage** rules. Leads to highly-specific rules
  - Highly-specific or uninterpretable rules are **not actionable**

Highly-specific Anchors

if ‘use’ and ‘18084tm’ and ‘just’ and ‘clipper’ and ‘center’ and ‘design’:
predict Electronics

Uninterpretable Trigger Phrases

TH PEOPLEMan goddreams Blacks

Tutorial Outline

~~(1) Overview of Interpretability~~

~~(2) What Parts of An Input Led to a Prediction?~~

~~(3) What Decision Rule Led to a Prediction?~~

QA + Break

(4) Which Training Examples Caused a Prediction?

(5) Implementing Interpretations

(6) Open Problems

Tutorial Outline

~~(1) Overview of Interpretability~~

~~(2) What Parts of An Input Led to a Prediction?~~

~~(3) What Decision Rule Led to a Prediction?~~

(4) Which Training Examples Caused a Prediction?

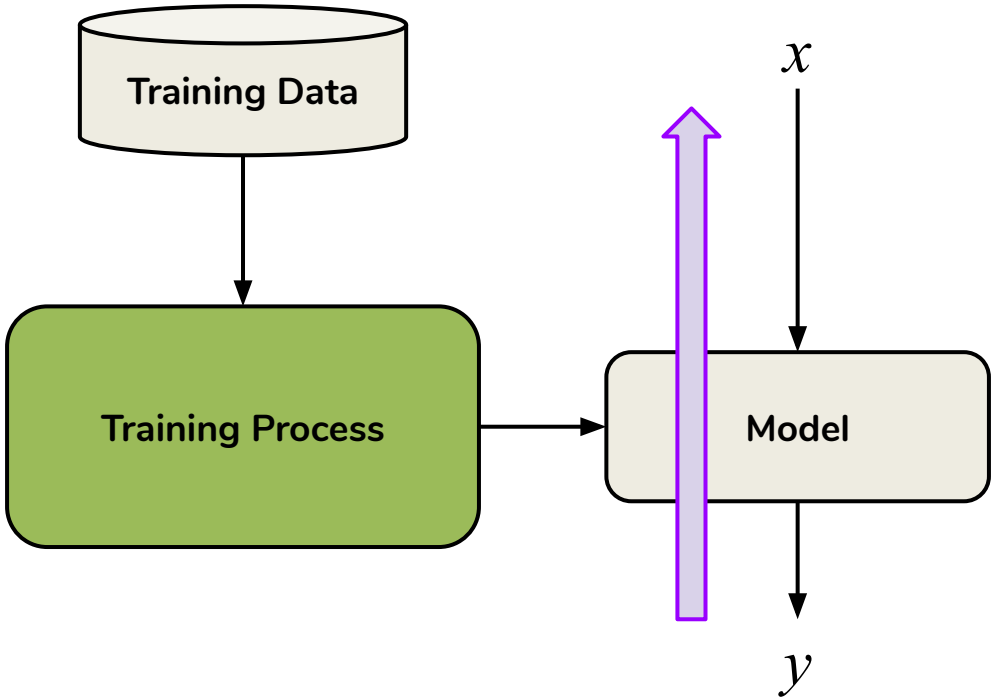
(5) Implementing Interpretations

(6) Open Problems

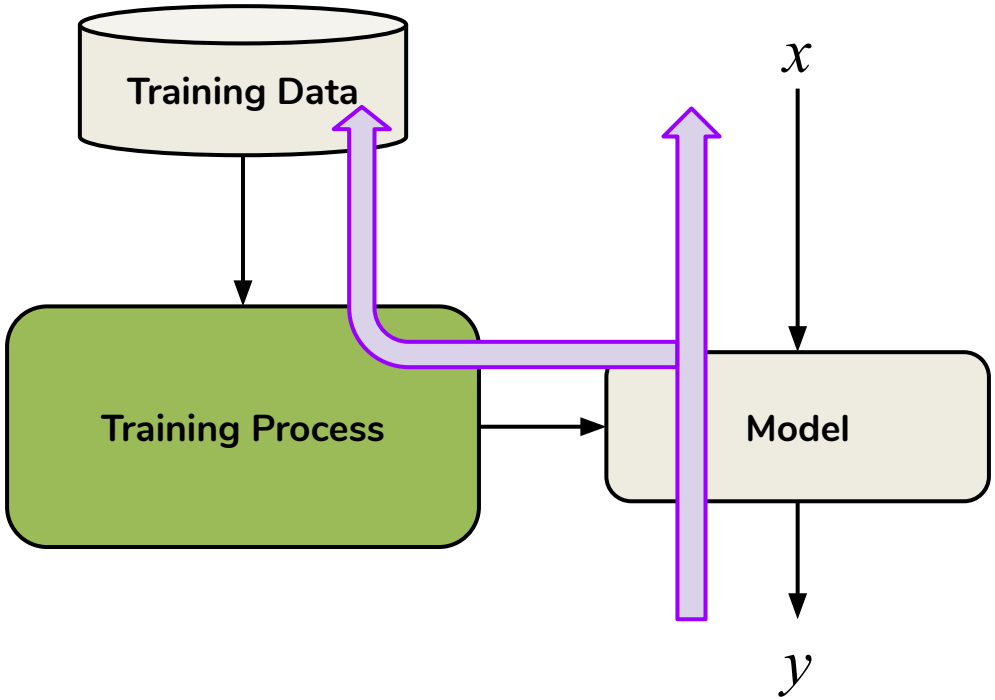
Why did my model make this prediction?

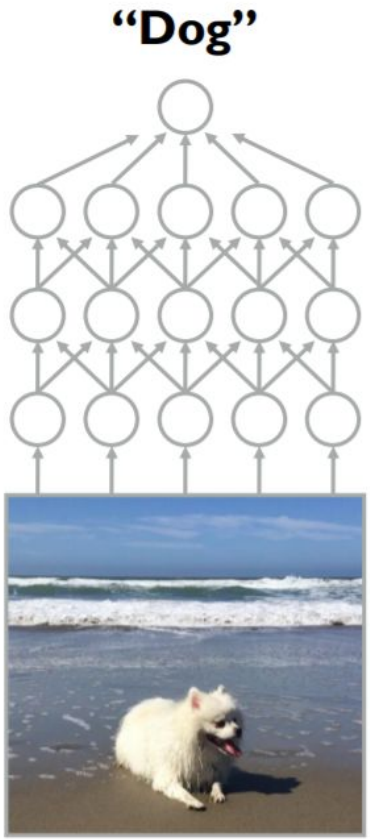
Which training examples were responsible for this prediction?

So far...

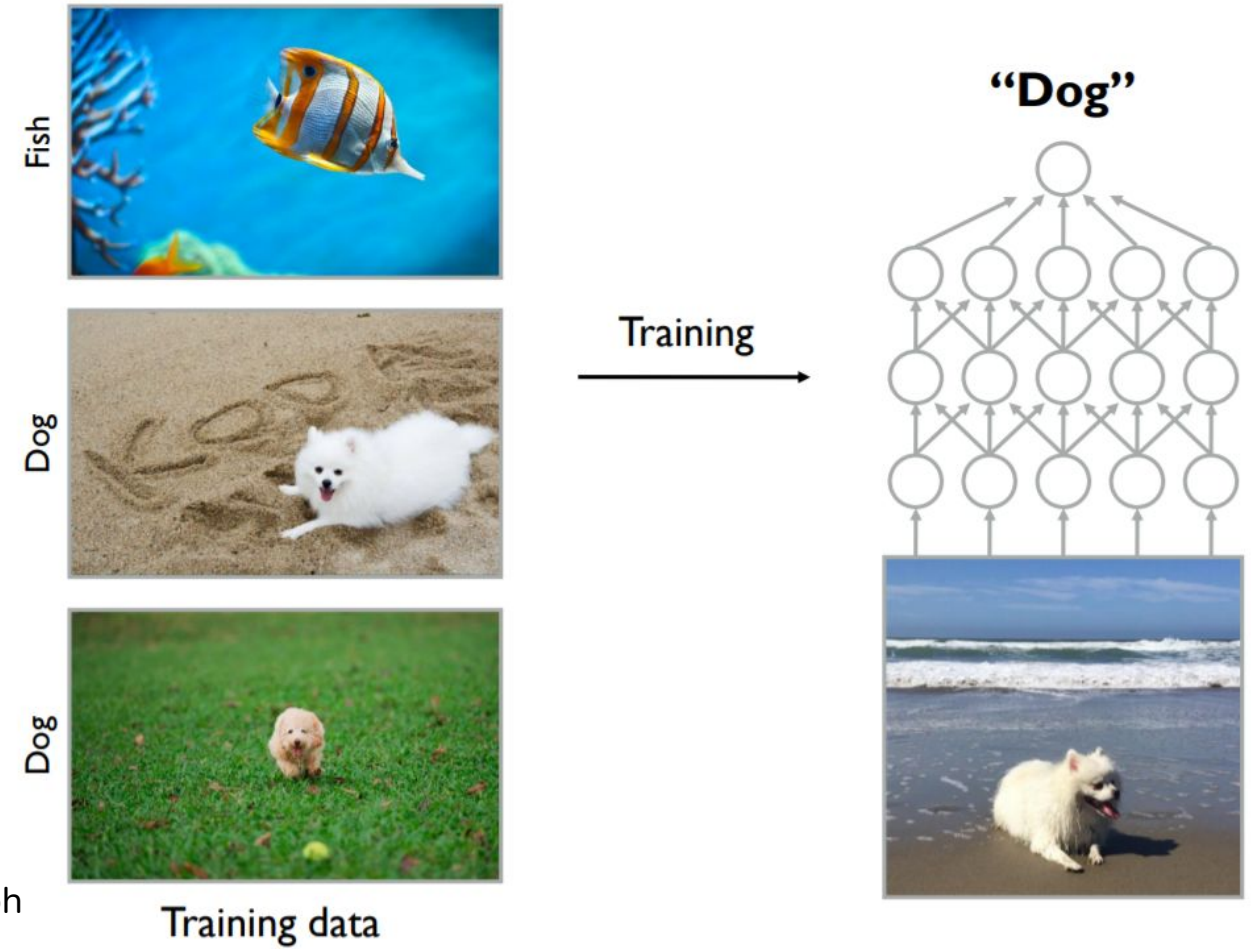


Data Influence





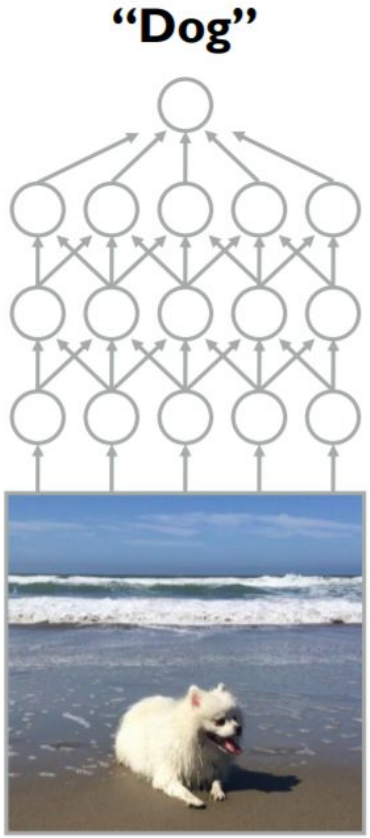
Slide: Pang Wei Koh



Slide: Pang Wei Koh

Most Important

Training



Slide: Pang Wei Koh

Data Influence: Example Use Cases [Yeh et al. 2018]

Test Example

Polar Bear ✘

Data Influence: Example Use Cases [Yeh et al. 2018]

Test Example

Polar Bear ✘

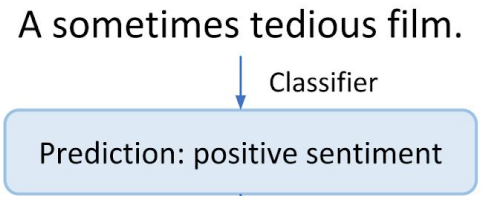
Influential Training Examples

Polar Bear ✘

Beaver

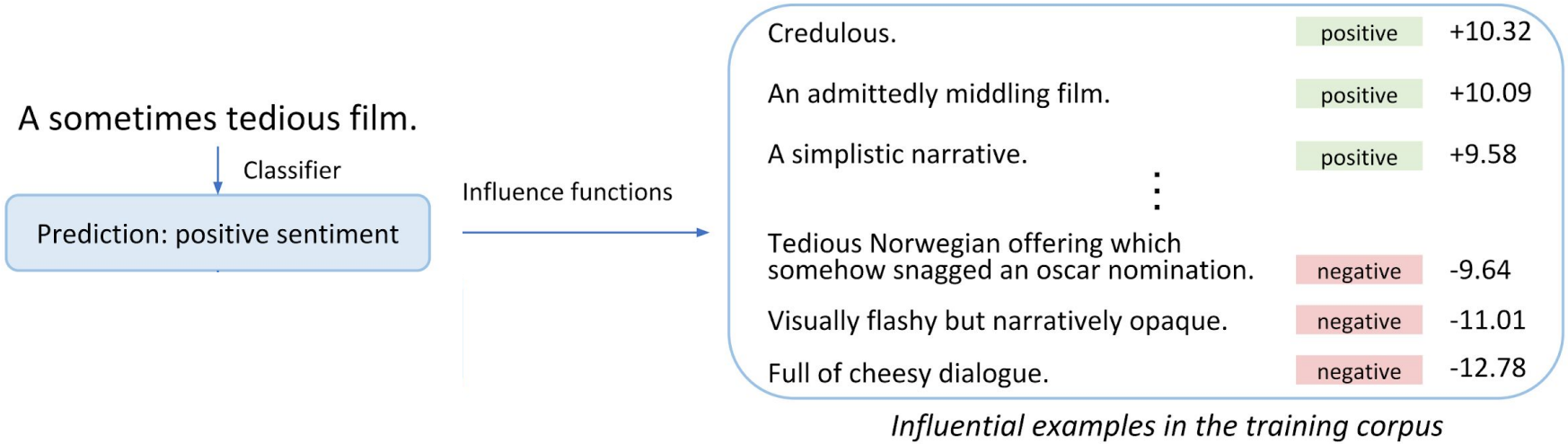
Pig

Data Influence: Example Use Cases



[Han et al. 2020]

Data Influence: Example Use Cases



Influential examples in the training corpus

[Han et al. 2020]

Benefits of Interpretation Via The Training Data

- Shows ***where*** the model picked up certain patterns

- **Actionable.** Find and fix examples that have:
  - incorrect labels
  - gender or racial biases
  - annotation artifacts

Influence Functions

Why did my model make this prediction?

Which training examples were responsible for this prediction?

Influence Functions

Why did my model make this prediction?

Which training examples were responsible for this prediction?

Which examples, if removed, would change the loss a lot?

[Koh and Liang 2017]

Influence of the Training Data

- Goal: for a given test prediction, identify the most influential training points

- Consider **test point** x, and **training point** z:

  $I(x, z)$ = How important is z for model's prediction for x

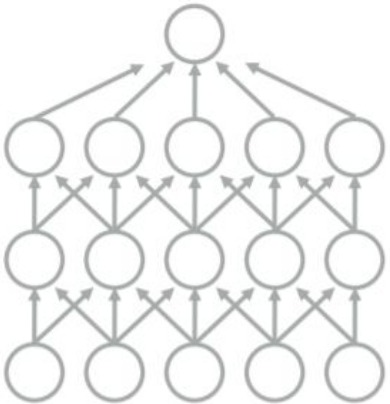
  In other words, **what is the influence of z on the prediction for x?**

1. "Remove" the training point z → change in parameters
2. Change in parameters → change in test prediction on input x

[Koh and Liang 2017]

Pick $\hat{\theta}$ to minimize $\frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta)$

Training data $z_1, z_2, \ldots, z_n$



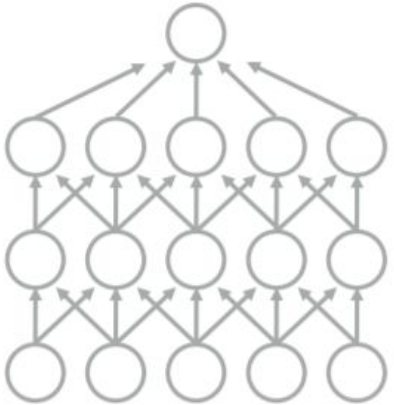
Slide: Pang Wei Koh

Pick $\hat{\theta}$ to minimize $\frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta)$

z_{train}

Training data $z_1, z_2, \ldots, z_n$

"Dog"



$\hat{\theta}$

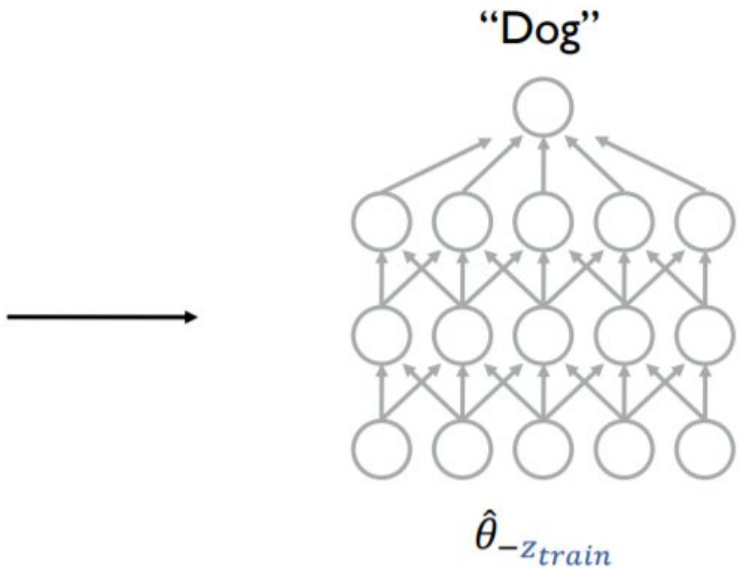
z_{train}

Training data $z_1, z_2, \ldots, z_n$

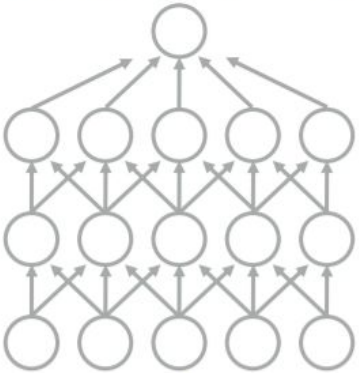
Pick $\hat{\theta}$ to minimize $\frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta)$

Pick $\hat{\theta}_{-z_{train}}$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta) - \frac{1}{n} L(z_{train}, \theta)$$



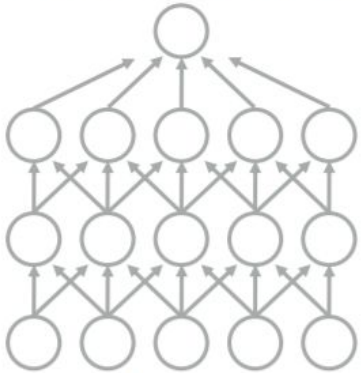
"Dog" (82% confidence)



$\hat{\theta}$

vs.

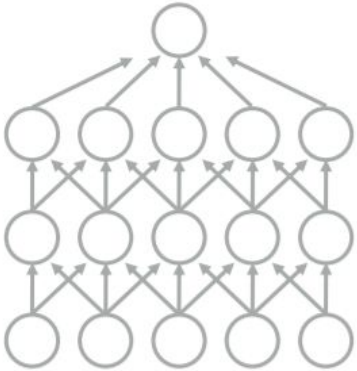
"Dog" (79% confidence)



$\hat{\theta}_{-z_{train}}$

Test input z_{test}

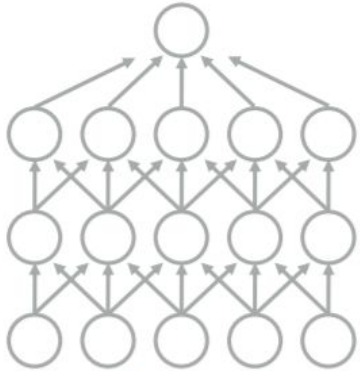
"Dog" (82% confidence)



$\hat{\theta}$

vs.

"Dog" (79% confidence)



$\hat{\theta}_{-z_{train}}$

What is $L\left(z_{test}, \hat{\theta}_{-z_{train}}\right) - L\left(z_{test}, \hat{\theta}\right)$?

Influence Functions Summary

Pros:

- Principled approach (in the convex setting) for estimating influence of individual training points
- Works empirically for many models

Cons:

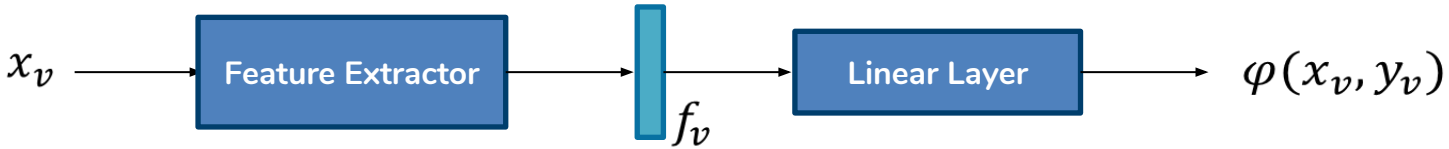
- Scales poorly with the size of the model and the training data
- Theory only applies in convex setting. Empirically, can struggle for some neural models [Basu et al. 2020]

Alternative method: representor point selection

Representer Point Selection

An efficient formulation of data influence

For a specific family of models trained with L_2 regularization:



Then, the model output can also be written as

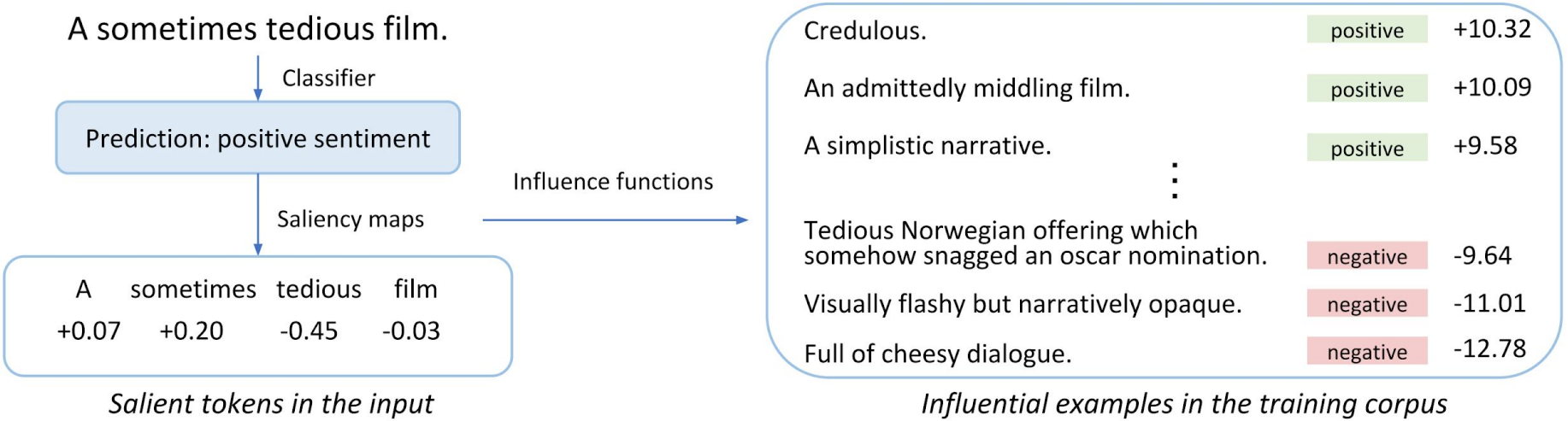
$$\varphi(x_v, y_v) = \sum_{x_t} \alpha_t f_t f_v$$

Importance of the training point *t* for test point *v*

Really **efficient** to compute!

[Yeh et al. 2018]

Use Case of Data Influence 1: Text Classification (Sentiment)



Salient tokens in the input

Influential examples in the training corpus

[Han et al. 2020]

Use Case of Data Influence 1: Text Classification (NLI)

Test input	
P: The manager was encouraged by the secretary. *H:* The secretary encouraged the manager.	{entail}

Most supporting training examples	
P: Because you're having fun. *H:* Because you're having fun.	[entail]
P: I don't know if I was in heaven or hell, said Lillian Carter, the president's mother, after a visit. *H:* The president's mother visited.	[entail]
P: Inverse price caps. *H:* Inward caps on price.	[entail]
P: Do it now, think 'bout it later. *H:* Don't think about it now, just do it.	[entail]

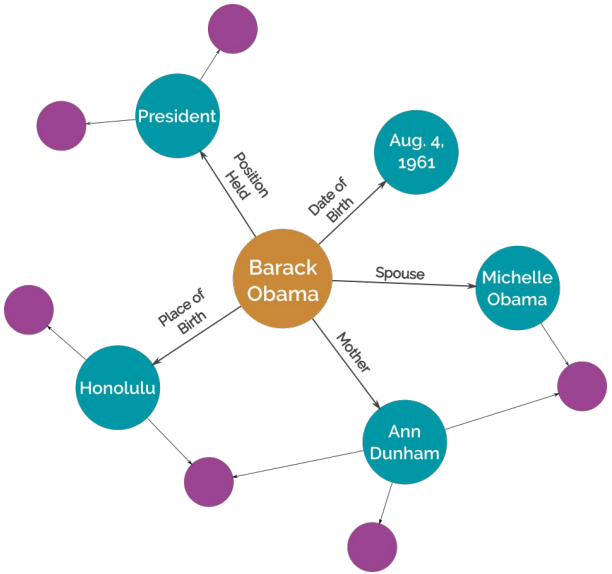
Most opposing training examples	
P: H'm, yes, that might be, said John. *H:* Yes, that might be the case, said John.	[non-entail]
P: This coalition of public and private entities undertakes initiatives aimed at raising public awareness about personal finance and retirement planning. *H:* Personal finance and retirement planning are initiatives aimed at raising public awareness.	[non-entail]

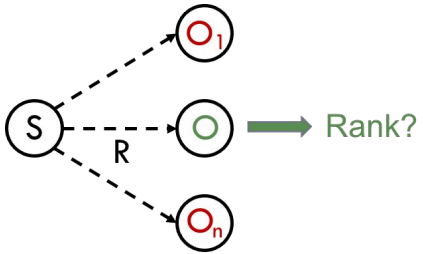
[Han et al. 2020]

Use Case of Data Influence 2: Understanding Link Prediction

- A knowledge graph is a directed graph where:
  - Vertices represent entities
  - Edges represent relations
- Often manually created/curated
  - High precision, but is quite incomplete..

Enter: completion via link prediction

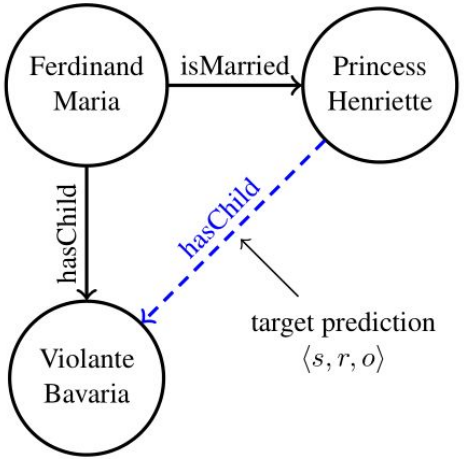


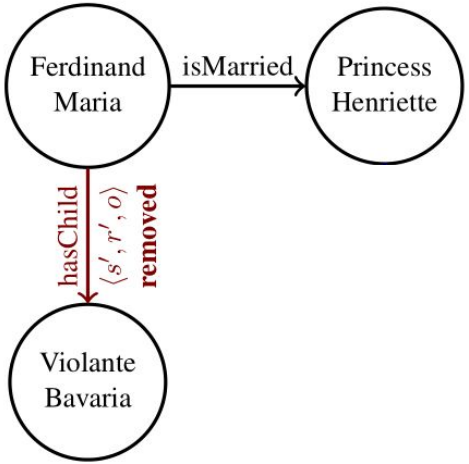


[Pezeshkpour et al. 2019]

Use Case of Data Influence 2: Understanding Link Prediction

What facts do these link prediction models use for their predictions?





[Pezeshkpour et al. 2019]

Removing Links: The Cause Of A Prediction

Which true fact do we need to remove to make something false?

isMarriedTo(a,c) ∧ **hasChild(c,b)** ⇒ hasChild(a,b)

isAffiliatedTo(a,c) ∧ **isLocatedIn(c,b)** ⇒ wasBornIn(a,b)

hasAdvisor(a,c) ∧ **graduatedFrom(c,b)** ⇒ graduatedFrom(a,b)

[Pezeshkpour et al. 2019]

Problems with Data Influence

- Influential points can be uninterpretable
  - What influence did it actually have?
- Computationally expensive [Garima et al. 2020]
  - Especially with large training data!
- Often requires approximations that may be invalid [Basu et al. 2020]
  - Would prediction really change if training example wasn't there?
- How does it interact with pretrained models?
  - Are the influential points too specific to choice of pretrained models?

Need more work in this area!

Tutorial Outline

~~(1) Overview of Interpretability~~

~~(2) What Parts of An Input Led to a Prediction?~~

~~(3) What Decision Rule Led to a Prediction?~~

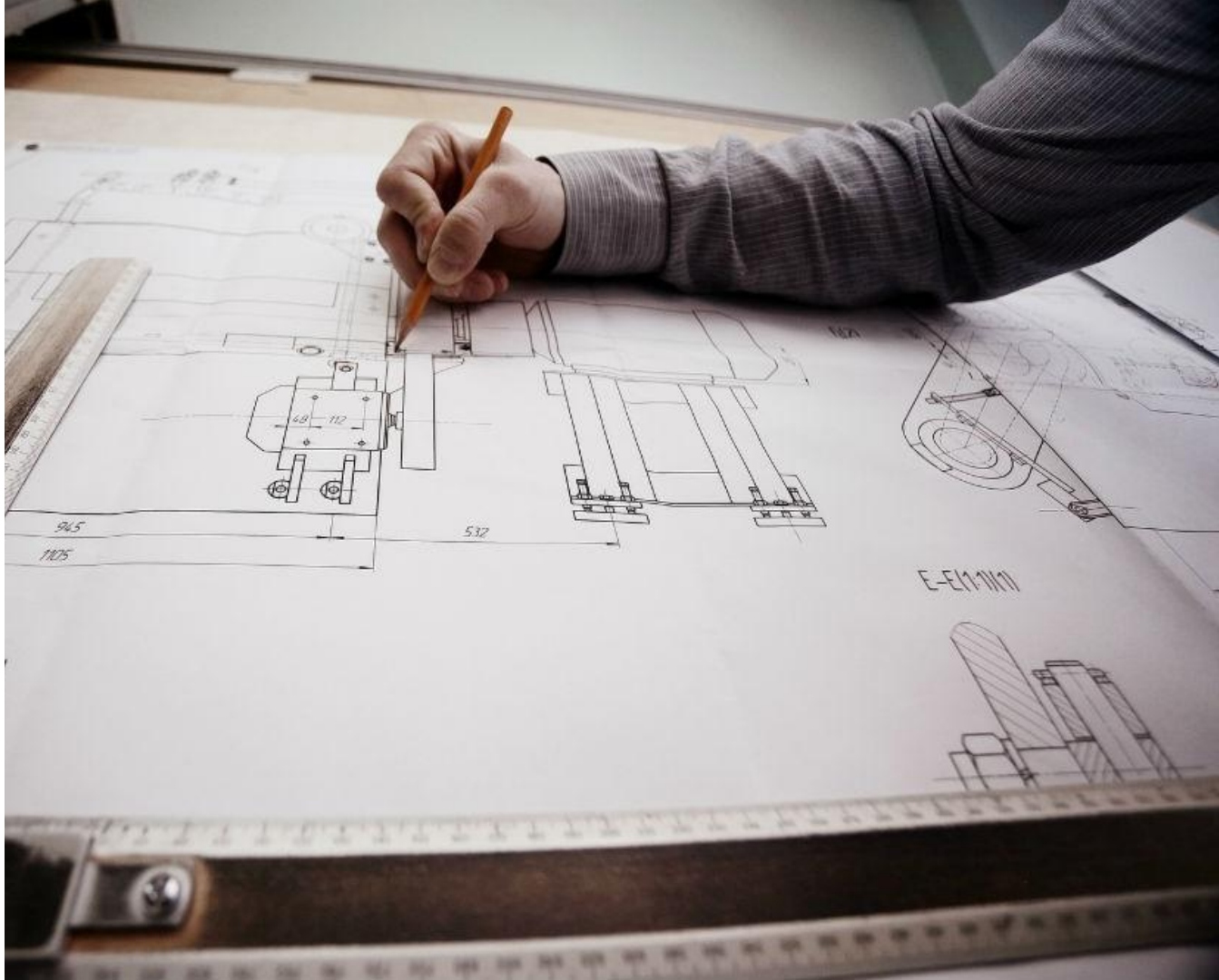
~~(4) Which Training Examples Caused a Prediction?~~

(5) Implementing Interpretations

(6) Open Problems

General considerations, not framework-specific examples

Goal: You should feel comfortable implementing these methods in whatever code you're using; it's not hard



Black-box perturbations	Input gradient access	Other requirements
LIME	Simple Gradient	Influence functions
Leave One Out	Smooth Gradient	Representer points
SEARs	Integrated Gradient	
SCPN	Input Reduction	
Bespoke adversaries	Hotflip	
Anchors	Triggers	

```
def perturb(
  instance: X or X,Y
) -> List[X]
```

```
def pred2label(
  model: Model
  instance: X
) -> List[X,Y]

def get_gradients(
  model: Model
  instance: X,Y
) -> Tensor
```

```
def perturb(instance: X or X,Y) -> List[X]
```

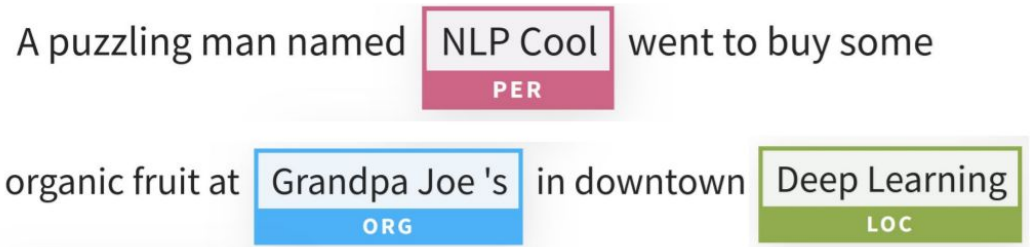
Not much interesting to say here: specific perturbation can do lots of different things:

- Drop words from input (LIME, Leave One Out)
- Use a model to produce perturbations (SCPN)
- Use a hand-written set of perturbation patterns (SEARs, Anchors, bespoke adversaries)

They also do lots of different things after perturbing

```
def pred2label(model: Model, instance: X) -> (X, Y):

    preds = model(instance)
    label = get_label(preds)
    new_instance = add_label(instance, label)
    return new_instance
```

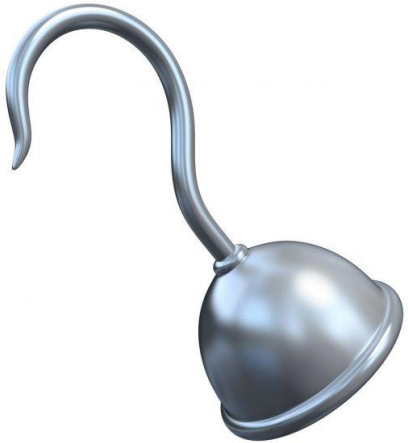
```
def pred2label(model: Model, instance: X) -> List[X, Y]:

    preds = model(instance)
    labels = get_labels(preds)
    new_instances = [
        add_label(instance, label)
        for label in labels
    ]
    # NOTE: with multiple instances, you need to
    # modify your model to mask the loss, too
    return new_instances
```



```
def get_gradients(model: Model, instance: X,Y) -> Tensor:
```

Key idea: use hooks!

- Backward hooks let you inspect (or modify) gradients
- Forward hooks let you inspect (or modify) activations
- Set hooks on the layers you need (typically the embedding layer)



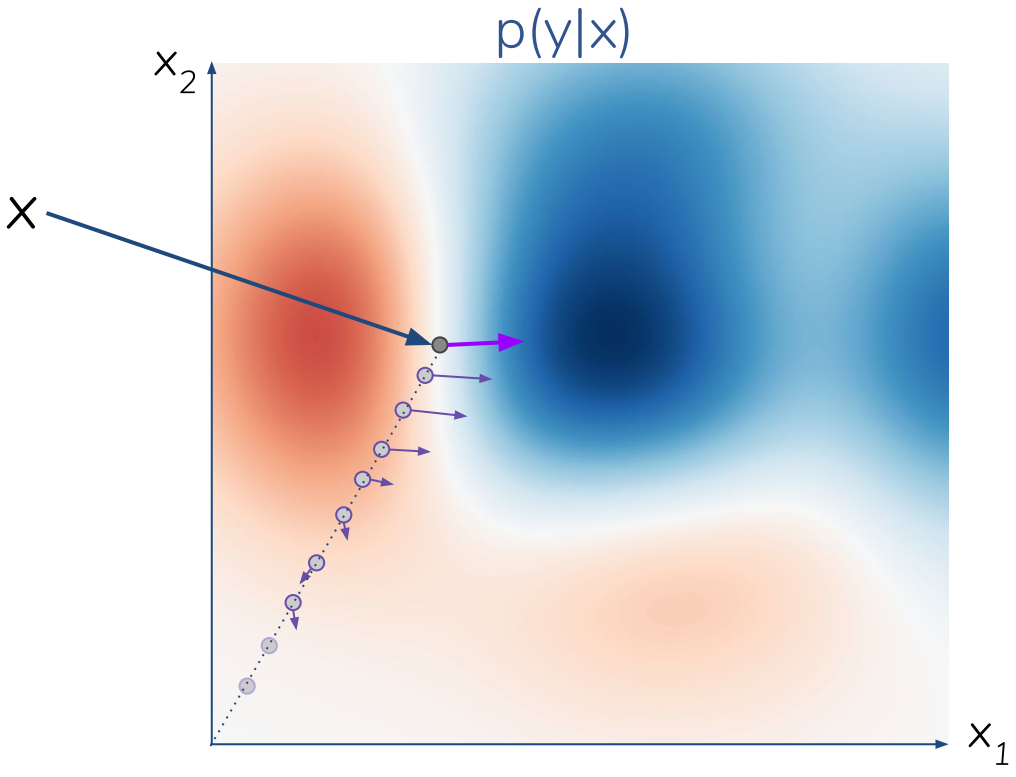
```
def get_gradients(model: Model, instance: X,Y) -> Tensor:

    embedding_gradients = []

    def grad_hook(module, grad_in, grad_out):
        embedding_gradients.append(grad_out[0])

    embedding = get_embedding_layer(model)
    handle = embedding.register_backward_hook(grad_hook)
    loss = model(instance)["loss"]  # may want logits instead
    loss.backward()  # calls grad_hook
    handle.remove()
    return embedding_gradients
```

```
def interpret(model: Model, instance: X,Y) -> SaliencyMap:
    forward_embeddings = []

    def forward_hook(module, inputs, outputs):
        forward_embeddings.append(outputs)

    embedding = get_embedding_layer(model)
    handle = embedding.register_forward_hook(forward_hook)
    # calls forward_hook (and grad_hook)
    gradients = get_gradients(model, instance)
    handle.remove()
    saliencies = dot_and_normalize(
        gradients,
        forward_embeddings,
    )
    return saliencies
```

$$-\nabla_{e(t)}\mathcal{L}_{\hat{y}} \cdot e(t)$$

Extensions of Vanilla Gradient

Integrated Gradients: average gradients along path from zero to input



[Sundararajan et al. 2017]

```
def int_grad(model: Model, instance: X,Y) -> SaliencyMap:
    grads = []
    forward_embeddings = []
    for alpha in numpy.linspace(0, 1, num=10):
        def hook(module, inputs, outputs):
            if alpha == 0:
                forward_embeddings.append(outputs)
            outputs.mul_(alpha)
            # or: outputs.add_(gaussian_noise)
        embedding = get_embedding_layer(model)
        handle = embedding.register_forward_hook(hook)
        # calls hook (and grad_hook)
        grads.append(get_gradients(model, instance))
        handle.remove()
    return average_grads(grads, forward_embeddings)
```

Lots of code available (in no particular order):

- https://captum.ai/tutorials/Bert_SQUAD_Interpret
- https://github.com/PAIR-code/lit
- https://allennlp.org/interpret
- https://github.com/QData/TextAttack
- https://github.com/interpretml/interpret-text
- https://github.com/sicara/tf-explain
- Influence Functions Code
- Influence functions for text
- Representer Points Code
- Triggers Code
- Anchors Code
- LIME Code

GitHub

Tutorial Outline

~~(1) Overview of Interpretability~~

~~(2) What Parts of An Input Led to a Prediction?~~

~~(3) What Decision Rule Led to a Prediction?~~

~~(4) Which Training Examples Caused a Prediction?~~

~~(5) Implementing Interpretations~~

(6) Open Problems

Open Problems

- Defining Interpretability
- Faithfulness of Interpretations
- Evaluating Utility of Interpretations
- New Methods and Forms of Interpretation

Defining Interpretability

What is Interpretability?

- There is debate on what the definition and goals of interpretability are
- We take a downstream task perspective: start with a goal and figure out how interpretability fits in
- Explicitly stating a goal makes the definitions, evaluation metrics, etc., of interpretability more concrete

[Miller 2017, Lipton 2018, Doshi-Velez and Kim 2017]

What is Interpretability?

- As downstream goals change, the definitions, requirements, etc., of interpretability will also change
    - This is ok because people's goals are very different!

- It's also perfectly ok for interpretability to not be necessary for your task

Faithfulness of Interpretations

Faithfulness: Interpretations Are Not Causal

- We asked questions like “What Parts of An Input **Led** to a Prediction?”
- However, most methods do not actually answer this causal question
- Can lead to unfaithful interpretations
  - doesn’t accurately represent the reasoning behind a model’s predictions

[Feng et al. 2018, Jain and Wallace 2019, Jacovi and Goldberg 2020]

Methods Can Be Completely Wrong

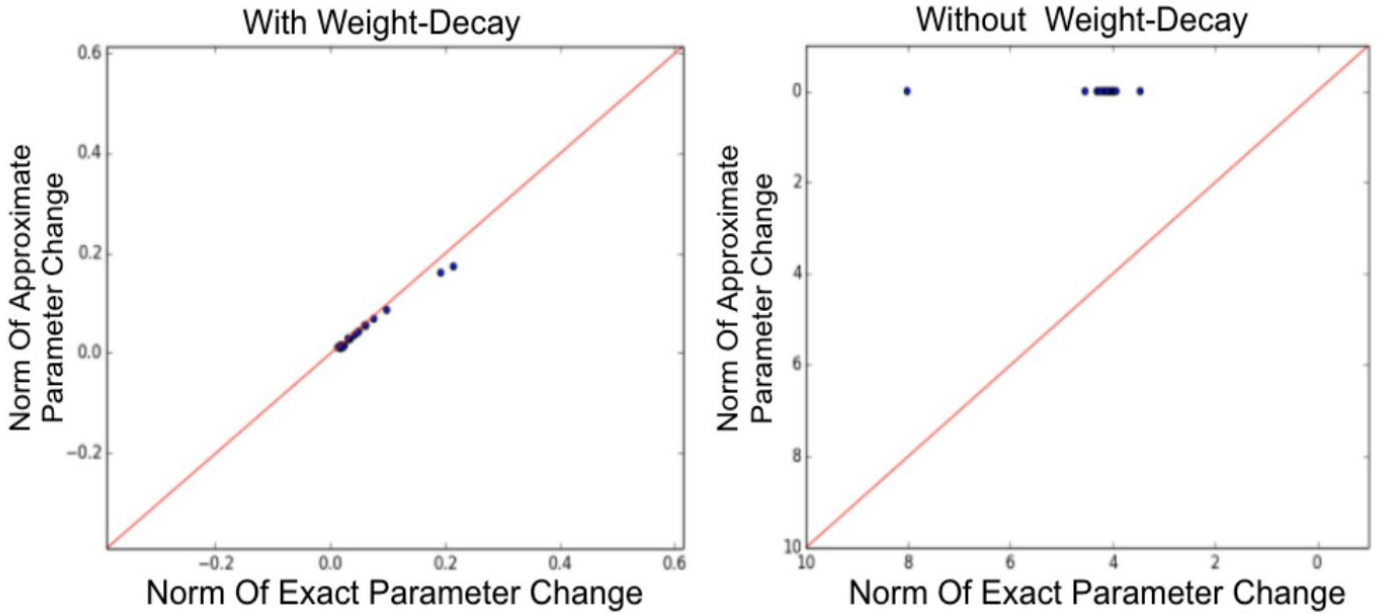
- Some saliency methods largely ignore model and instead reconstruct input

- Lesson: test for correctness with simple “sanity checks”

[Adebayo et al. 2018, Kindermans et al. 2017, Nie et al. 2018]

Approximations Can Be Loose

- Most interpretation methods make approximations. Should test if these approximations are valid!



[Basu et al. 2020, Singla et al. 2019, Han et al. 2020]

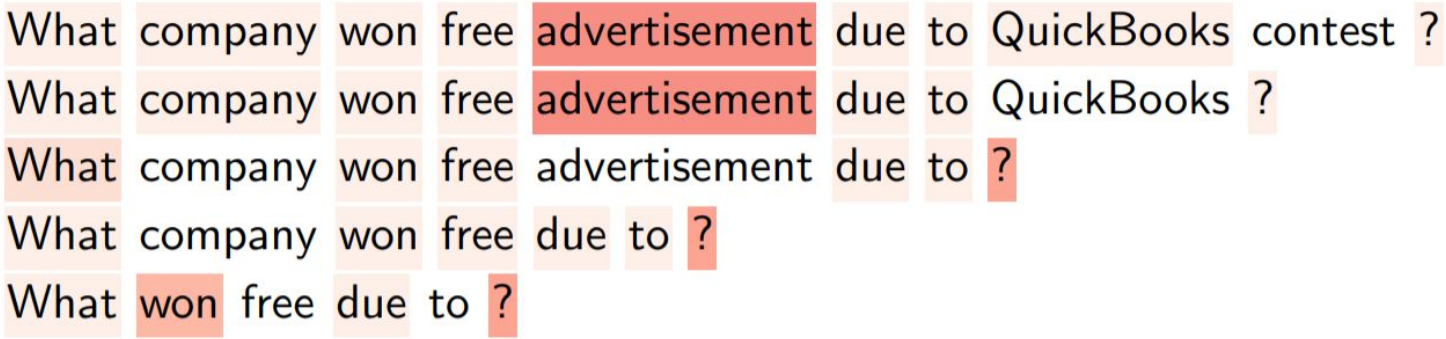
Interpretations Can Be Unstable

- Saliency maps can be highly unstable. Very similar inputs lead to very different interpretations

[Ghorbani et al. 2019, Feng et al. 2018, Alvarez-Melis et al. 2018, Yeh et al. 2019]

Interpretations Can Be Unstable

- Saliency maps can be highly unstable. Very similar inputs lead to very different interpretations



- Why? methods use gradients which are valid *infinitesimally locally*
- Lesson: test stability as noise (random or adversarial) is added?
- Do these results say more about the model or the interpretation?

[Ghorbani et al. 2019, Feng et al. 2018, Alvarez-Melis et al. 2018, Yeh et al. 2019]

Interpretations Can Be Adversarially Manipulated

- Models can be trained to guide how their interpretations look
  - e.g., via regularization or adding dummy layers

Attention	Biography	Label
Original	Ms. X practices medicine in Memphis, TN and is affiliated ... Ms. X speaks English and Spanish.	Physician
Ours	Ms. X practices medicine in Memphis , TN and is affiliated ... Ms. X speaks English and Spanish.	Physician

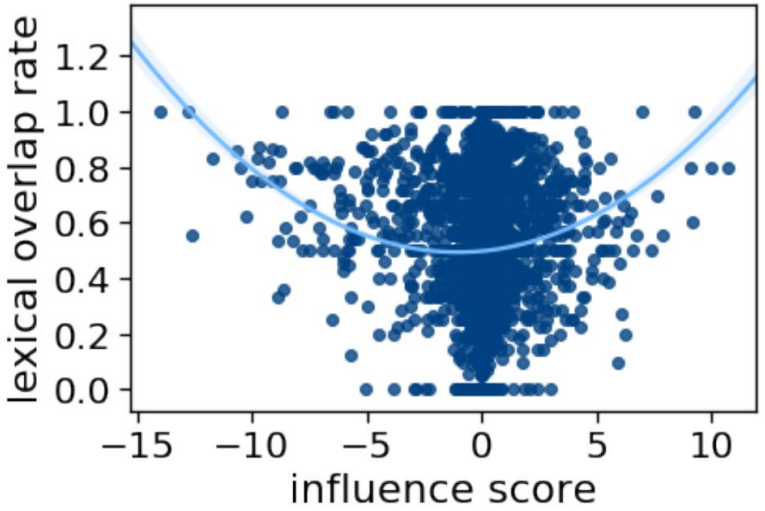
- Audits for model biases can be exploited
- If interpretations in the worst-case are completely unfaithful, what about the average case?

[Ross et al. 2017, Pruthi et al. 2020, Wang et al. 2020]

Other Faithfulness Tests

Measure alignment of interpretations with a proxy for “ground truth”

- MT interpretations and word alignment [Li et al. 2020]
- NLI interpretations and dataset artifacts [Han et al. 2020]



Faithfulness Summary

- Interpretations have varying degrees of “faithfulness” to model. Various intrinsic tests exist
- Low faithfulness means interpretations are typically useless
- Instead of intrinsic tests, can we directly evaluate an interpretation’s utility?

Evaluating The Utility of Interpretations

Evaluating the Utility of Interpretations

- Humans are the end users of interpretability, so do human studies!

- Yes, user studies are expensive in time, effort, and money. But worth it:
  - forces thought about real-world use cases
  - realistic evaluations

- We'll discuss two use cases of human studies:
  - Model debugging
  - Human-AI collaboration

Debugging Models and Datasets with Interpretability

- Ask users to find model errors with and without interpretations
  - saliency maps [Lertvittayakumjorn et al. 2020, Ribeiro et al. 2016]
  - error analysis tools [Wu et al. 2019]
  - input perturbations [Ribeiro et al. 2018, Ribeiro et al. 2020]

From: johnchad@triton.unm.edu (jchadwic)
Subject: Another request for Darwin Fish
Organization: University of New Mexico, Albuquerque
Lines: 11
NNTP-Posting-Host: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the DARWIN fish.
This is the same question I have and I have not seen an answer on

	Unaided	CHECKLIST
#Tests	5.8 ± 1.1	**13.5** ± 3.4
#Bugs ($sev \geqslant 3$)	2.2 ± 1.2	**6.2** ± 0.9

Debugging Models and Datasets with Interpretability

Good news: we are great at **finding bugs**

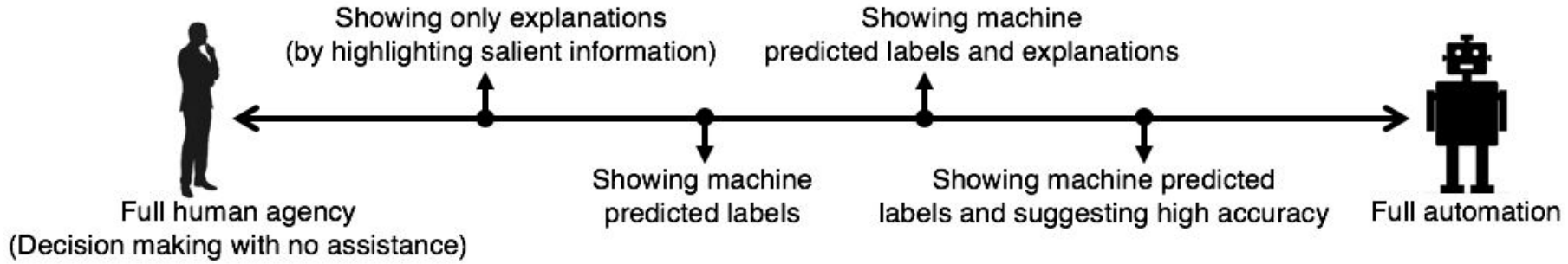
- input perturbations that cause errors
- incorrect features used by model
- common validation errors

Bad news: we don't know how to **fix bugs**

- a failure of current neural NLP models?
- how to make neural models more interactive and editable? [Amershi et al. 2014, Raghavan et al. 2006, Smith-Renner et al. 2020]

Human-AI Collaboration

- Human-AI collaboration: ML model and human team up
- Effective teams must communicate! Interpretability provides a way.

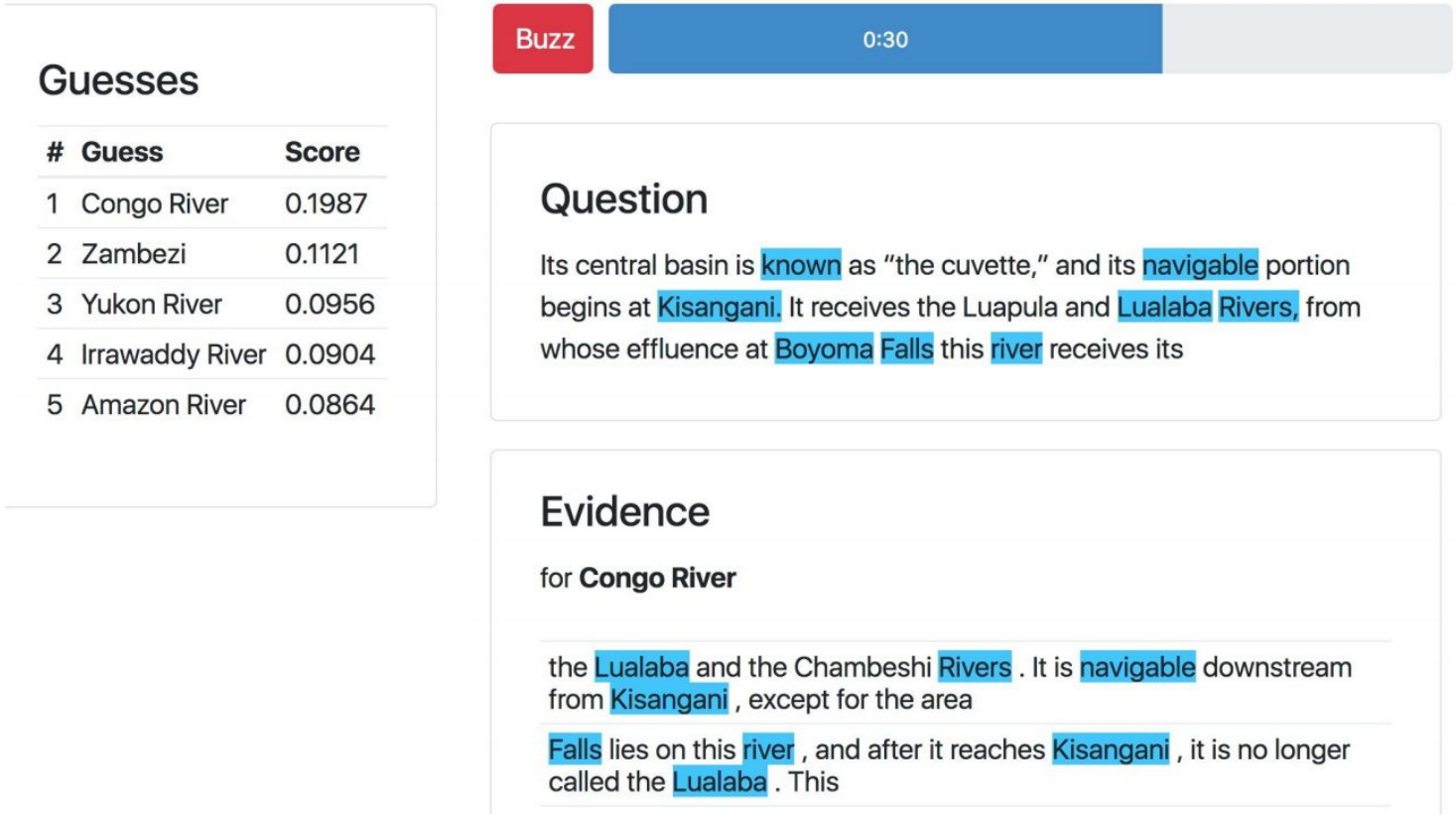


Interpretations can communicate on

- Instance-level: is the model predicting correctly on this input?
- Global-level: what types of things is the model good and bad at?

[Lai and Tan 2019]

Human-AI Collaboration



[Feng and Boyd-Graber 2019]

Human-AI Collaboration

- Run full end-to-end evaluation for different NLP tasks [Feng and Boyd-Graber 2019, Lai and Tan 2019, Chu et al. 2020, Zhang et al. 2020, Bansal et al. 2020]
  - test human accuracy and speed with and without interpretations

- Evaluate subproblem of how well humans can simulate model predictions [Nguyen et al. 2018, Chandrasekaran et al. 2018, Shen and Huang 2020, Hase and Bansal 2020]

Human-AI Collaboration

Mixed results for using interpretability in human-AI collaboration

Successes:

- Interpretations help to detect model errors
  - improves overall team accuracy
  - faster human processing times
- Interpretations help to debug/improve models in long-term

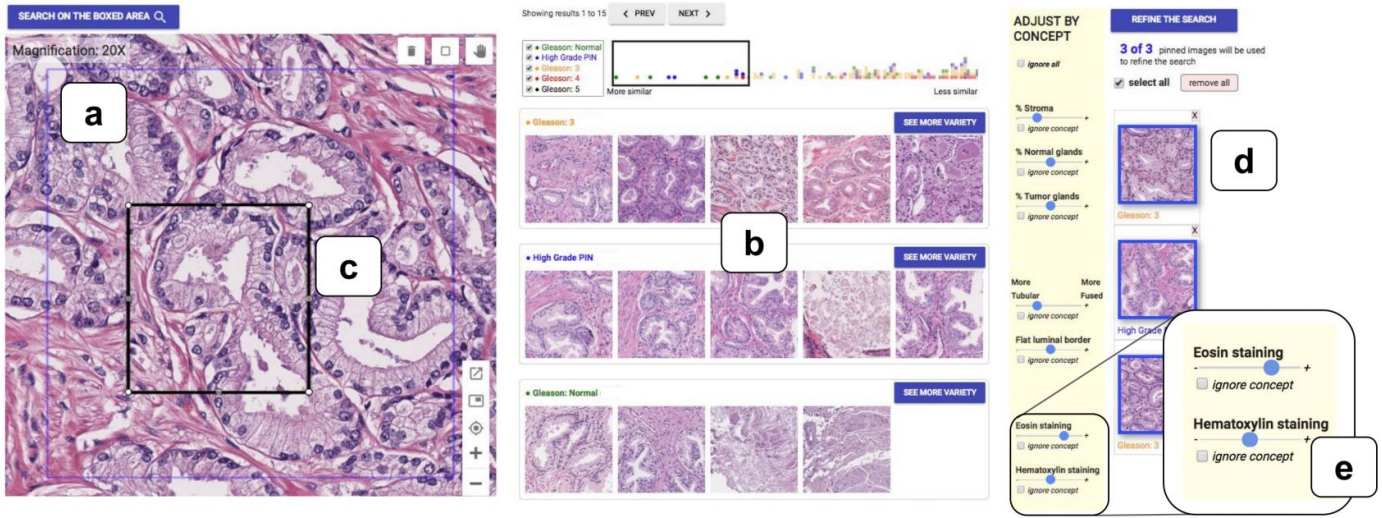
Failures:

- Misleading interpretations hinder detection of model errors
- Information overload slows or hinders humans
- Exposing failures can cause over distrust of models
- Displaying confidence sometimes as good as interpretations

New Methods and Forms of Interpretation

Closing the Loop with Humans

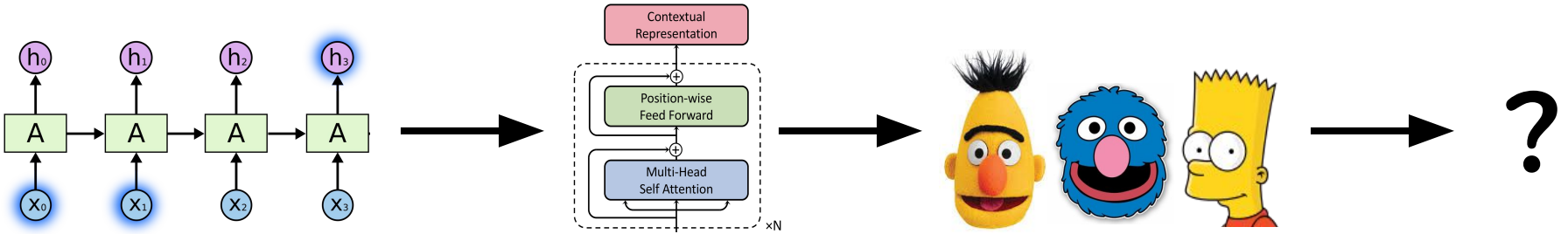
- Build interpretable systems with end users in mind
  - work with end users
  - build interactive and collaborative interfaces



[Cai et al. 2019, Beede et al. 2020]

How Much Do Methods and Knowledge “Generalize”?

- Model architectures and datasets constantly change
- Do interpretation methods “generalize” across models?
  - many were developed for vision or small LSTM models
- Does knowledge “generalize” as models change?
  - “models trained on X dataset incorrectly use Y feature”



LSTMs Transformers Pretraining

New Interpretation Methods

- Plenty of exciting directions for new methods

Areas of emphasis:

- Global decision rules
- Understanding training examples
- Interpretation beyond classification
- Generating text explanations

Tutorial Outline

~~(1) Overview of Interpretability~~

~~(2) What Parts of An Input Led to a Prediction?~~

~~(3) What Decision Rule Led to a Prediction?~~

~~(4) Which Training Examples Caused a Prediction?~~

~~(5) Implementing Interpretations~~

~~(6) Open Problems~~

Summary

(1) Overview of Interpretability

- Find errors + bugs in models and data
- Understand models so that they can trusted (or not trusted)
- Aid human-AI collaboration

(2) What Parts of An Input Led to a Prediction?

- Saliency maps
- Input perturbations and adversarial attacks

(3) What Decision Rule Led to a Prediction?

- Anchors
- Universal Adversarial Triggers

Summary

(4) What Training Examples Led to a Prediction?

- Influence Functions
- Representor Points

(5) Implementing Interpretations

- Pretty easy!

(6) Open Problems

- Defining interpretability
- Faithfulness of interpretations
- Evaluation

Thank You!

Eric Wallace
@Eric_Wallace_
ericwallace@berkeley.edu

Matt Gardner
@nlpmattg
mattg@allenai.org

Sameer Singh
@sameer_
sameer@uci.edu

Slides and Video ericswallace.com/interpretability