Interpretability

Ksenia Sokolova & Michael Hu
COS 598C
April 21, 2020

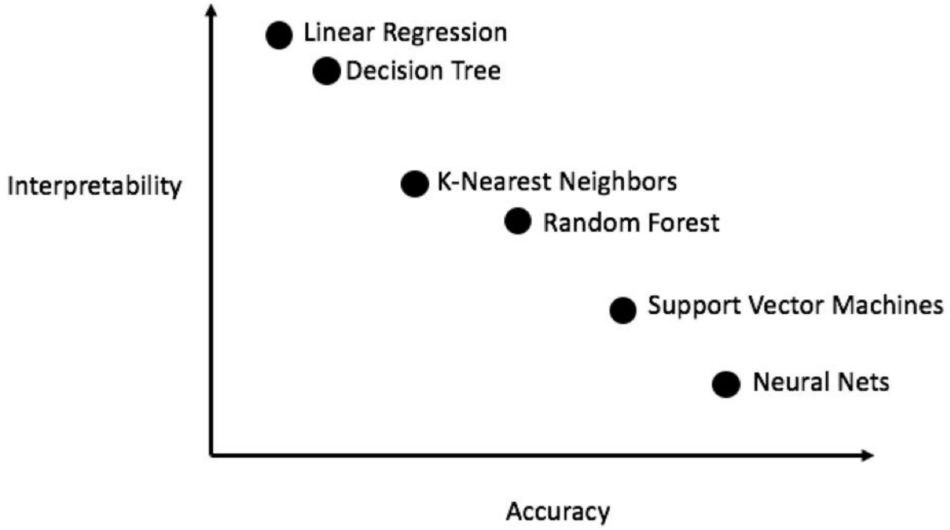
Outline

- Interpretability introduction
- AllenNLP Interpret
  - Framework overview
  - Saliency maps
  - Adversarial attacks
- Probing
  - Overview
  - Using control tasks
  - Experiments

A philosophical shift

Hand-engineered models:
- Lots of components
- Easy to intuit what each component is learning

New concept:
- One model
- Black-box: hard to tell what the model is learning



Interpretability background

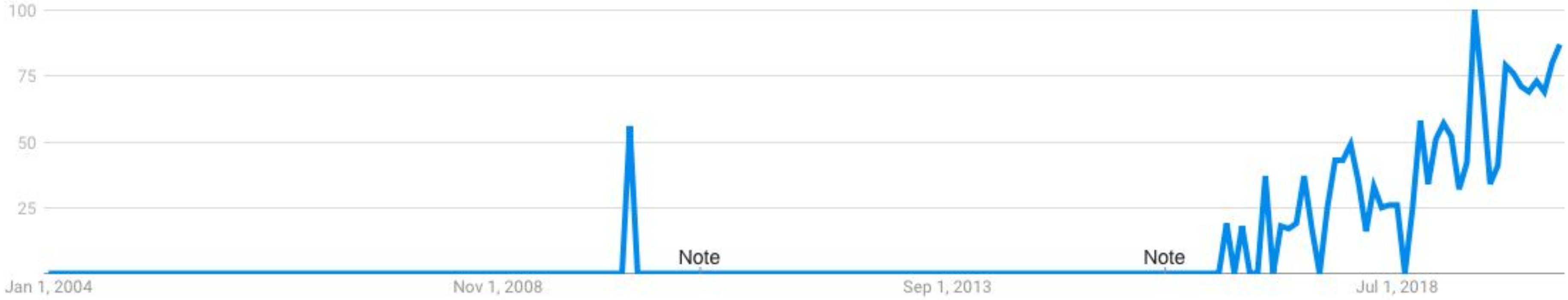
- Problems of annotation artifacts, biases and adversarial attacks

What does it mean for a model to be interpretable?

- No mathematical definition of interpretability.
  - *"Interpretability is the degree to which a human can understand the cause of a decision." (*)*
  - *"Interpretability is the degree to which a human can consistently predict the model's result" (**)*
- Interpret on the level of **predictions**
  - Why did the model make certain predictions?
- Interpret on the level of **components**
  - How does the model make predictions?

*Miller, Tim. "Explanation in artificial intelligence: Insights from the social sciences." arXiv Preprint arXiv:1706.07269. (2017)
** Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. "Examples are not enough, learn to criticize! Criticism for interpretability." Advances in Neural Information Processing Systems (2016)

Interpretability is becoming more popular



Google trends result for ‘explainable AI’

Outline

- Interpretability introduction
- AllenNLP Interpret
    - Framework overview
    - Saliency maps
    - Adversarial attacks
- Probing
    - Overview
    - Using control tasks
    - Experiments

Open source toolkit with methods to analyze model performance and reasoning

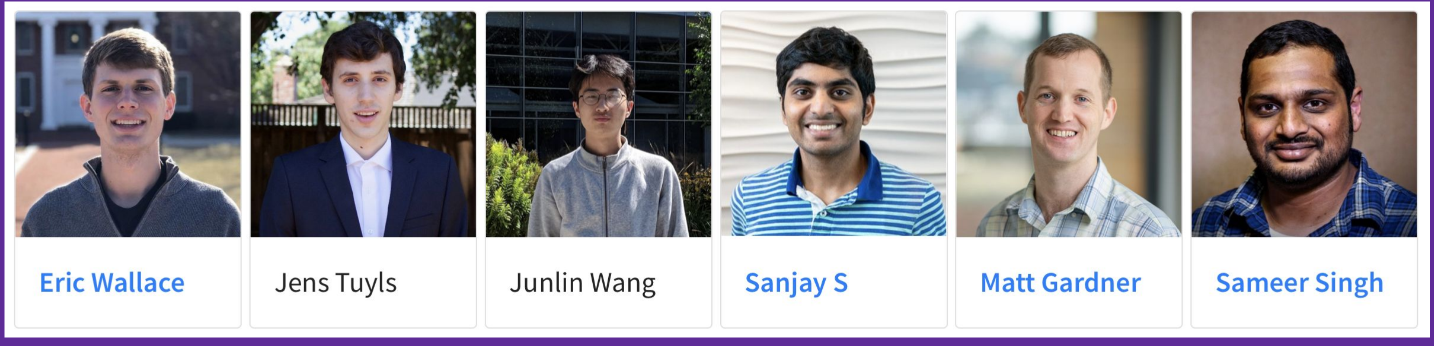
Turning supervised tasks into tools for interpreting representations

AllenNLP Interpret:
A Framework for Explaining Predictions of NLP Models

Eric Wallace[1]   Jens Tuyls[2]   Junlin Wang[2]   Sanjay Subramanian[1]
Matt Gardner[1]   Sameer Singh[2]

[1]Allen Institute for Artificial Intelligence   [2]University of California, Irvine

ericw@allenai.org, sameer@uci.edu



AllenNLP

An open-source NLP research library, built on PyTorch

Package Overview

allennlp	an open-source NLP research library, built on PyTorch
allennlp.commands	functionality for a CLI and web service
allennlp.data	a data processing module for loading datasets and encoding strings as integers for representation in matrices
allennlp.models	a collection of state-of-the-art models
allennlp.modules	a collection of PyTorch modules for use with text
allennlp.nn	tensor utility functions, such as initializers and activation functions
allennlp.training	functionality for training models

Introduction

Why did my model make this prediction?

Most of the open-source interpretation before either were narrow focused or analyzed existing models

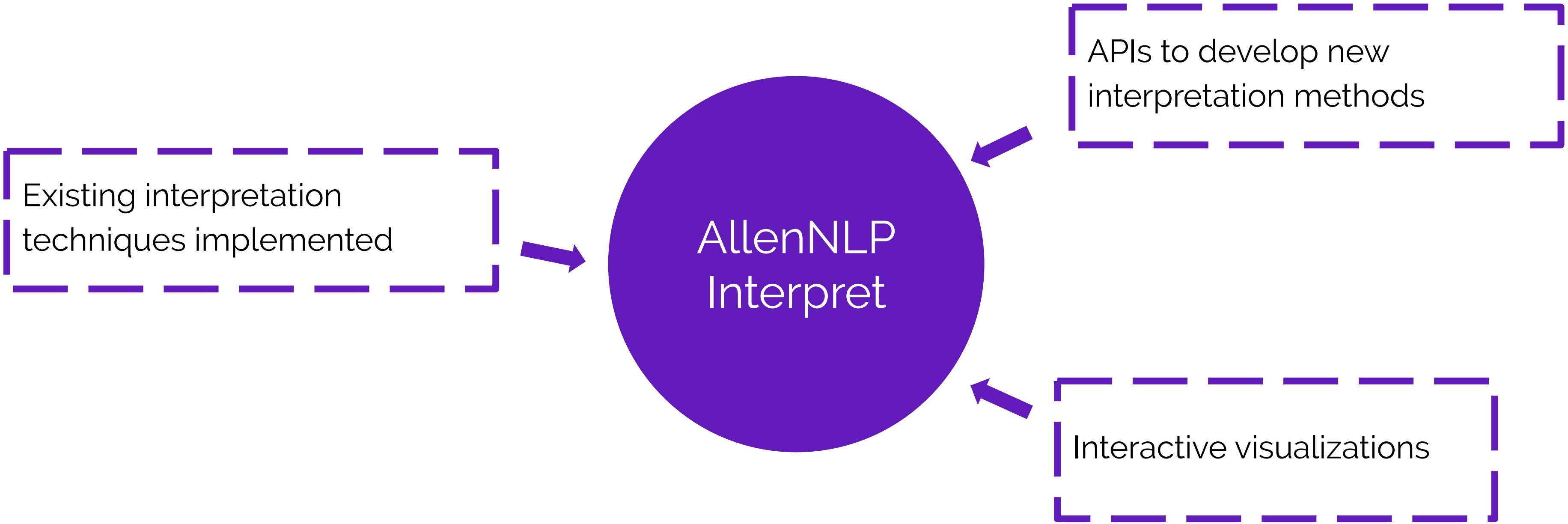
Introduction

Why *did* ***my*** *model make this prediction?*

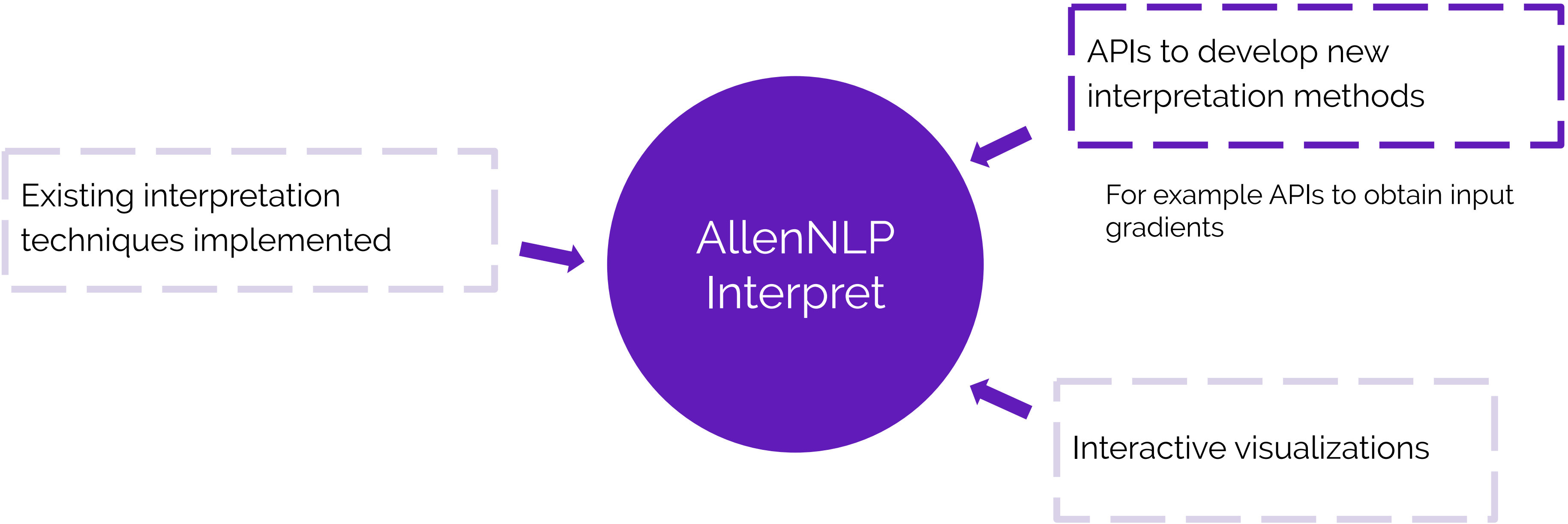
Most of the open-source interpretation before either were narrow focused or analyzed existing models

Allen NLP Interpret: allows to apply existing interpretation methods to *new models*, as well as develop *new interpretation methods*

AllenNLP Interpret Toolkit



AllenNLP Interpret Toolkit



AllenNLP Interpret Toolkit

Existing interpretation techniques implemented

AllenNLP Interpret

APIs to develop new interpretation methods

Interactive visualizations

HTML and JavaScript components that are available for clear vis

Available Interpretations

Available Interpretations

Saliency based

Adversarial attacks

Saliency maps

Identifying the importance of the input tokens using gradients

Attempt to highlight regions which model was "looking at" when making decisions

Images from: Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep inside convolutional networks: Visualising image classification models and saliency maps.* 2014. In *ICLR*.

Saliency maps implemented in AllenNLP Interpret

Vanilla Gradient	Integrated Gradient	SmoothGrad
Simonyan et al, 2014	Sundararajan et al, 2017	Smilkov et al, 2017

Vanilla Gradient

Gradient of the loss with respect to each token

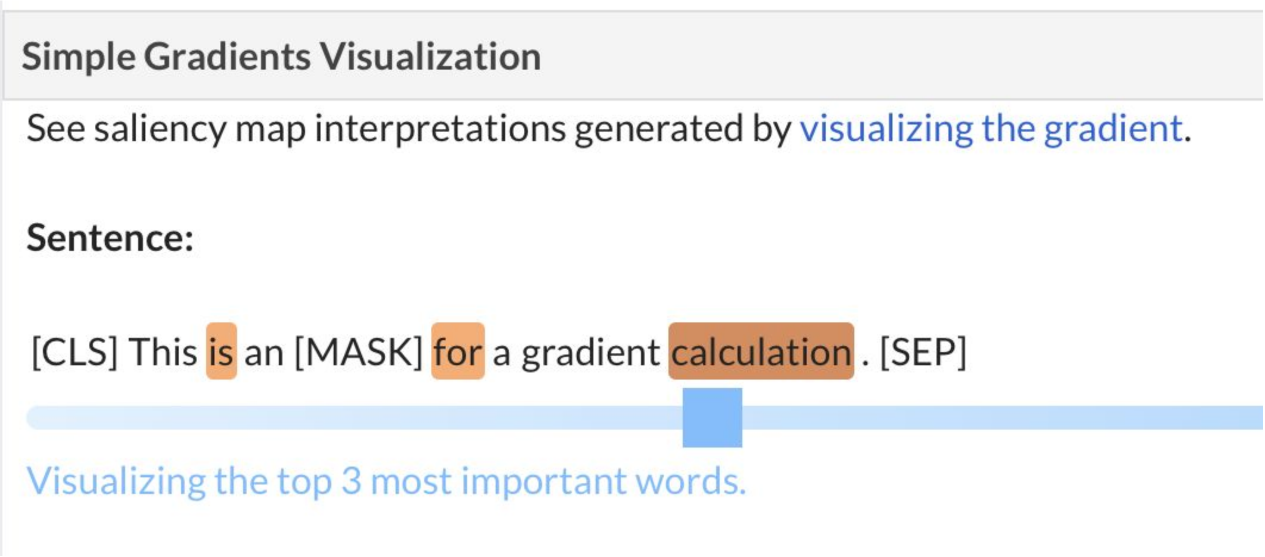
- Derivative of the input is found by backpropagation on the trained model
- Saliency map for each token

Method published in: Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep inside convolutional networks: Visualising image classification models and saliency maps.* 2014. In *ICLR*.

Vanilla Gradient

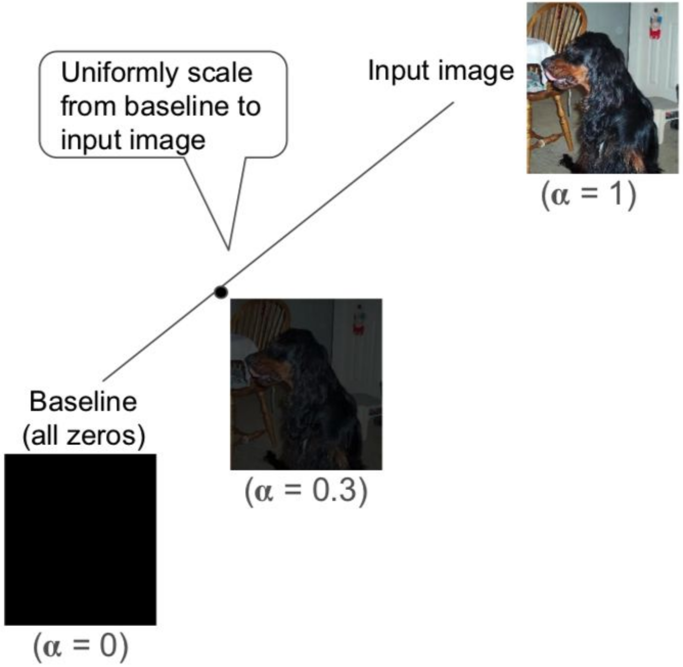
Gradient of the loss with respect to each token

Example for BERT



Method published in: Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep inside convolutional networks: Visualising image classification models and saliency maps.* 2014. In *ICLR*.

Integrated Gradients



An intuitive understanding:

- Construct a sequence of entries, interpolating from baseline to the actual entry
- Average the gradients

Image from: http://theory.stanford.edu/~ataly/Talks/sri_attribution_talk_jun_2017.pdf

Integrated Gradients

Define a baseline x', which is an input absent of information. Word importance is determined by integrating the gradient along the path from this baseline to the original input

$$\phi_i^{IG}(f, x, x') = \overbrace{(x_i - x'_i)}^{\text{Difference from baseline}} \times \underbrace{\int_{\alpha=0}^{1}}_{\text{From baseline to input...}} \overbrace{\frac{\delta f(x' + \alpha(x - x'))}{\delta x_i}}^{\text{...accumulate local gradients}} d\alpha$$

- Avoids problems with local gradients being saturated

Method published in: Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. *Axiomatic attribution for deep networks.* In ICML.

What is an “empty input”?

Integrated Gradients

Define a baseline x′, which is an input absent of information. Word importance is determined by integrating the gradient along the path from this baseline to the original input

$$\phi_i^{IG}(f, x, x') = \overbrace{(x_i - x'_i)}^{\text{Difference from baseline}} \times \underbrace{\int_{\alpha=0}^{1}}_{\text{From baseline to input...}} \overbrace{\frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha}^{\text{...accumulate local gradients}}$$

- Avoids problems with local gradients being saturated

Method published in: Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. *Axiomatic attribution for deep networks.* In ICML.

Integrated Gradients

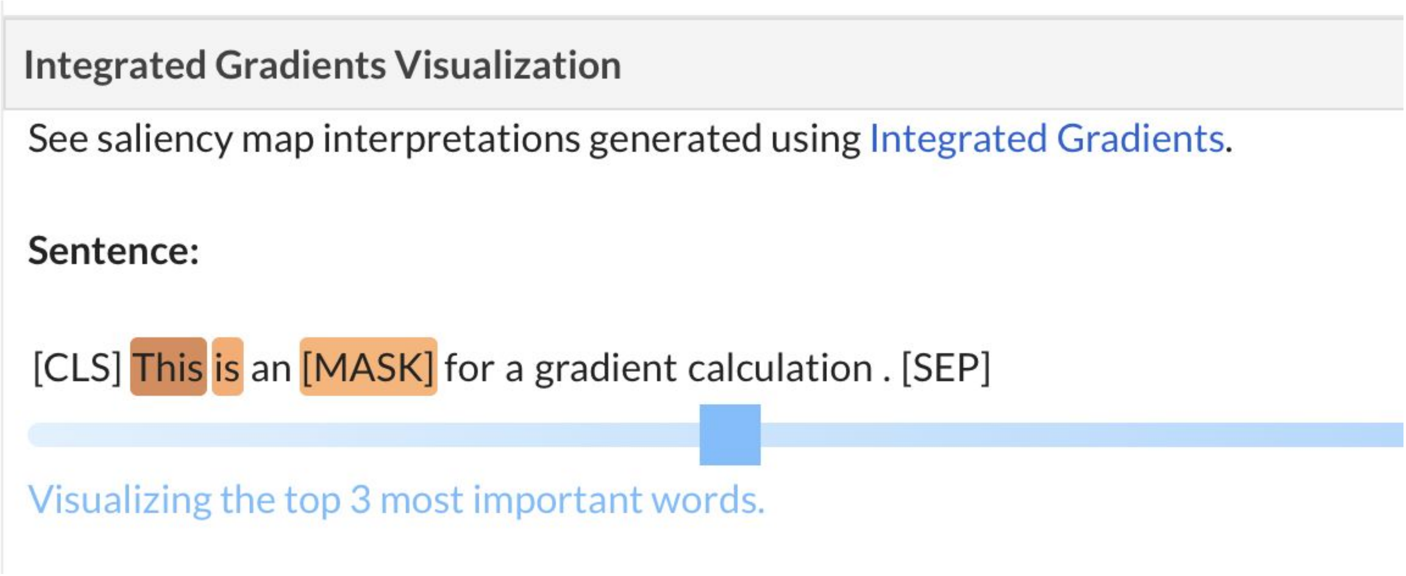
In practice a discrete sum approximation is used, with a scale parameter.

```
# List of Embedding inputs
embeddings_list: List[numpy.ndarray] = []

# Use 10 terms in the summation approximation of the integral in integrated grad
steps = 10

# Exclude the endpoint because we do a left point integral approximation
for alpha in numpy.linspace(0, 1.0, num=steps, endpoint=False):
    # Hook for modifying embedding value
    handle = self._register_forward_hook(alpha, embeddings_list)

    grads = self.predictor.get_gradients([instance])[0]
    handle.remove()

    # Running sum of gradients
    if ig_grads == {}:
        ig_grads = grads
    else:
        for key in grads.keys():
            ig_grads[key] += grads[key]

# Average of each gradient term
for key in ig_grads.keys():
    ig_grads[key] /= steps
```

Method published in: Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. *Axiomatic attribution for deep networks.* In ICML.

Integrated Gradients

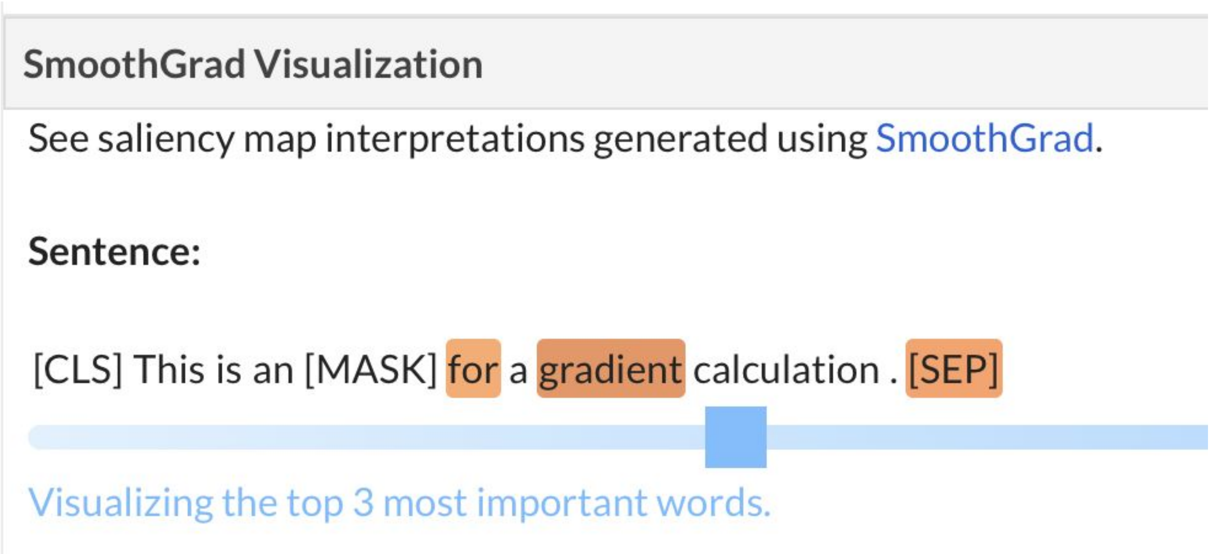
Define a baseline x′, which is an input absent of information. Word importance is determined by integrating the gradient along the path from this baseline to the original input



Method published in: Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. *Axiomatic attribution for deep networks.* In ICML.

SmoothGrad

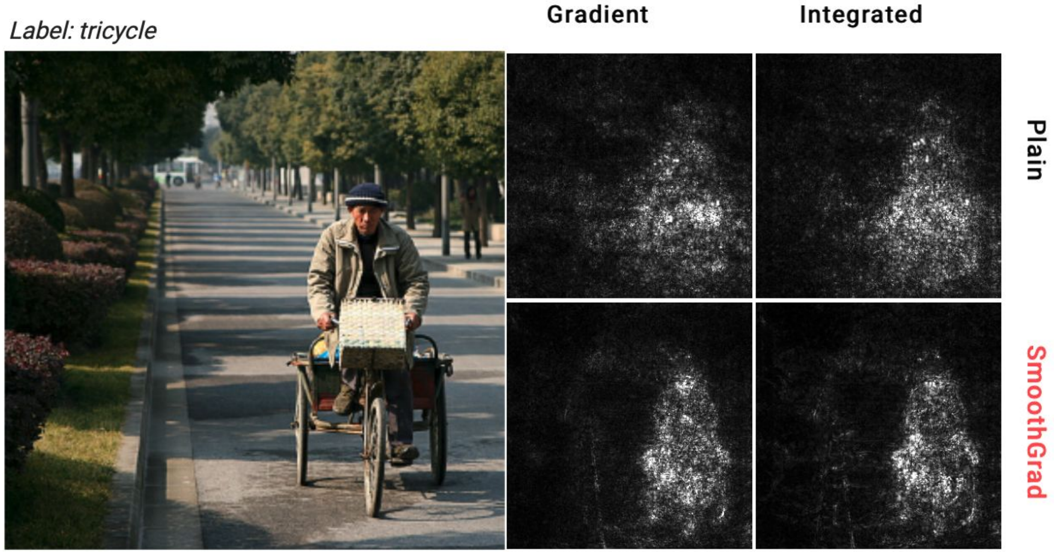
Average the gradient over many noisy versions of the input by adding Gaussian noise to embeddings and taking averages



Method published in: Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viegas, and Martin Wattenberg. 2017. *SmoothGrad: removing noise by adding noise.* In ICML Workshop on Visualization for Deep Learning

SmoothGrad

Average the gradient over many noisy versions of the input by adding Gaussian noise to embeddings and taking averages



Method published in: Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viegas, and Martin Wattenberg. 2017. *SmoothGrad: removing noise by adding noise.* In ICML Workshop on Visualization for Deep Learning

Saliency maps: comparison

Sentence:

Hand [MASK] is the act of cleaning one's hands for the purpose of removing soil, grease, microorganisms, or other unwanted substances.

Mask 1 Predictions:

71.5%	cleaning
25.3%	washing
0.6%	wash
0.5%	wiping
0.4%	removal

Saliency maps: comparison

Sentence:

Hand [MASK] is the act of cleaning one's hands for the purpose of removing soil, grease, microorganisms, or other unwanted substances.

Mask 1 Predictions:

71.5% **cleaning**

25.3% **washing**

0.6% **wash**

0.5% **wiping**

0.4% **removal**

Vanilla gradient

[CLS] Hand [MASK] is the act of cleaning one ' s hands for the purpose of removing soil , g ##rease , micro ##or ##gan ##isms , or other unwanted substances . [SEP]

Integrated gradient

[CLS] Hand [MASK] is the act of cleaning one ' s hands for the purpose of removing soil , g ##rease , micro ##or ##gan ##isms , or other unwanted substances . [SEP]

SmoothGrad

[CLS] Hand [MASK] is the act of cleaning one ' s hands for the purpose of removing soil , g ##rease , micro ##or ##gan ##isms , or other unwanted substances . [SEP]

Adversarial Attacks

HotFlip

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. *HotFlip: White-box adversarial examples for text classification.* In *ACL*.

uses the gradient to swap out words from the input in order to change the model's prediction

Input Reduction

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In EMNLP.

remove as many words as possible from the input without changing a model's prediction

HotFlip Attack

Use gradients to estimate an individual change that would have the greatest effect, followed by a beam search to find an optimal manipulation strategy

South Africa's historic Soweto township marks its 100th birthday on Tuesday in a mood of optimism. 57% **World**
South Africa's historic Soweto township marks its 100th birthday on Tuesday in a moo**P** of optimism. 95% **Sci/Tech**
Chancellor Gordon Brown has sought to quell speculation over who should run the Labour Party and turned the attack on the opposition Conservatives. 75% **World**
Chancellor Gordon Brown has sought to quell speculation over who should run the Labour Party and turned the attack on the o**B**position Conservatives. 94% **Business**

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. *HotFlip: White-box adversarial examples for text classification.* In *ACL*.

HotFlip Attack

Example from AllenNLP paper:

Original Input: an interesting story about two lovers , I would recommend it to anyone !

Flipped Input: an interesting story about two lovers , I would recommend it to inadequate !

Prediction changed to: Negative

From online demo:

Original Input: [CLS] This is an [MASK] of a Hot ##F ##lip attack for the presentation . [SEP]

Flipped Input: [CLS] design is an [MASK] of the Hot ##F ##lip render for the kernel . [SEP]

Prediction changed to: extension

Input Reduction

Answer

Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss, Hugo Weaving, and Joe Pantoliano

Passage Context

The Matrix is a 1999 science fiction action film written and directed by The Wachowskis, starring **Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss, Hugo Weaving, and Joe Pantoliano** . It depicts a dystopian future in which reality as perceived by most humans is actually a simulated reality called "the Matrix", created by sentient machines to subdue the human population, while their bodies' heat and electrical activity are used as an energy source. Computer programmer "Neo" learns this truth and is drawn into a rebellion against the machines, which involves other people who have been freed from the "dream world."

Question

Who stars in The Matrix?

Original Input: ~~Who~~ stars ~~in~~ ~~The~~ Matrix ~~?~~

Reduced Input: stars Matrix

Input reduction

Textual entailment example (decomposable attention combined with ELMo model, trained on SNLI dataset)

Premise: Two women are wandering along the shore drinking iced tea.

Original Input: ~~Two~~ ~~women~~ ~~are~~ ~~sitting~~ ~~on~~ ~~a~~ ~~blanket~~ ~~near~~ ~~some~~ ~~rocks~~ ~~talking~~ ~~about~~ politics ~~.~~

Reduced Input: politics

Result: contradiction for both original and reduced inputs

Adversarial attacks and interpretability

Adversarial attacks can help to diagnose model vulnerabilities

Training using adversarial examples could provide more interpretable saliency maps *

Alternatively, can use interpretability to detect adversarial attacks **

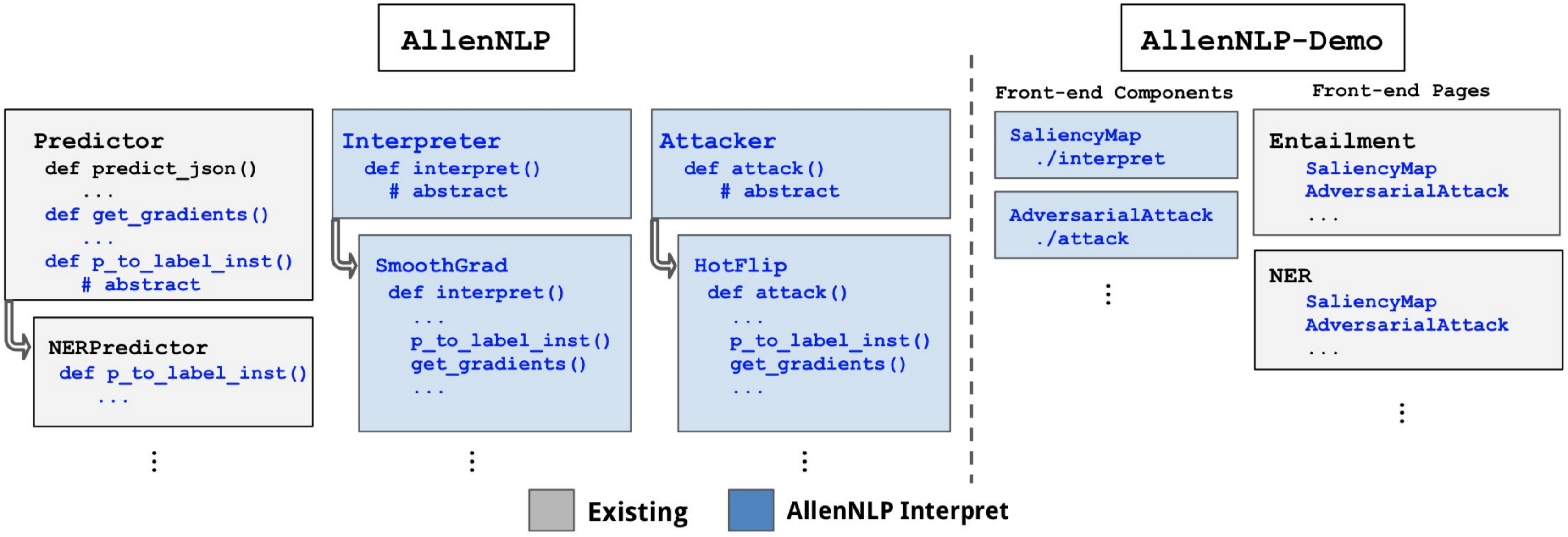
* Etmann, Christian, et al. "On the connection between adversarial robustness and saliency map interpretability." *arXiv preprint arXiv:1905.04172* (2019).
** Tao, Guanhong, et al. "Attacks meet interpretability: Attribute-steered detection of adversarial samples." *Advances in Neural Information Processing Systems*. 2018.

Quick demo

Available Models in AllenNLP Interpret

- Reading Comprehension
  - NAQANet, BiDAF
- Masked Language Modeling
  - BERT, RoBERTa and more
- Text Classification and Textual Entailment
  - BiLSTM and self-attention classifiers
- Named Entity Recognition (NER) and Coreference Resolution

System Overview



Conclusions - what is AllenNLP Interpret

Open-source flexible toolkit that facilitates model analysis

- Convenient framework that can run on custom models
- Existing methods as well as ability to add custom analysis methods
- Ready-to-use visualization toolkit
- Interesting demo online to play with in your free time

Conclusions: saliency maps and adversarial attacks

Saliency maps allow to assess the importance of input tokens for prediction using gradients

- Convenient tool to gain insights into the model
- Different methods might produce different results

Adversarial attacks demonstrate network misbehaviors and can be used to produce more interpretable results (or alternatively interpretable models should be less susceptible to adversarial attacks)

Probing

Probing is a hot new interpretability technique developed to understand large models.

A philosophical shift

Hand-engineered models:

- Lots of components
- Easy to intuit what each component is learning

Unsupervised representations

- One model, one component
- Black-box: hard to tell what the model is learning

Hypothesis: Deep learners encode linguistic properties in their intermediate representations.

How to test this hypothesis?

Formal definitions

Sentence 1	The	cat	ran	quickly	.
Part-of-speech	DT	NN	VBD	RB	.

$$\text{a sentence}: \quad x_{1:T} = \{x_1, x_2, ..., x_T\}$$

$$\text{intermediate representations}: \quad h_{1:T}$$

$$\text{output labels}: \quad y_{1:T}$$

$$\text{a task}: \quad f(x_{1:T}) = y_{1:T}$$

$$\text{a probe}: \quad f_\theta(h_{1:T}) = \hat{y}_{1:T}$$

Probing

$$f_\theta(h_{1:T}) = \hat{y}_{1:T}$$

Probes are **supervised** models.

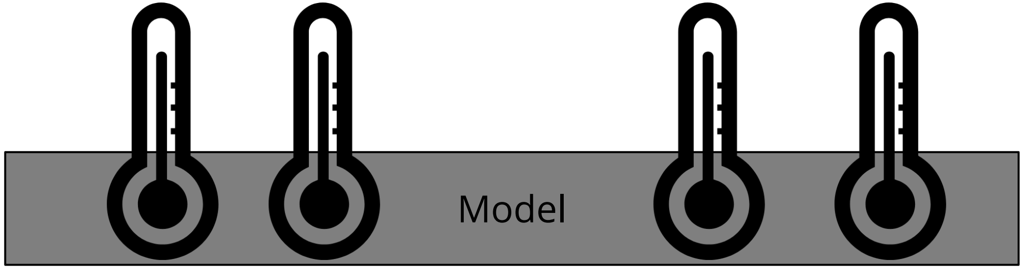
A probe consumes the representation of another model to perform a task.

Argument: Good performance by a probe on some task → the upstream model encodes linguistic information about that task.

Say we study BERT embeddings. If a probe consumes BERT embeddings and performs well on POS tagging, we say BERT implicitly encodes parts of speech.

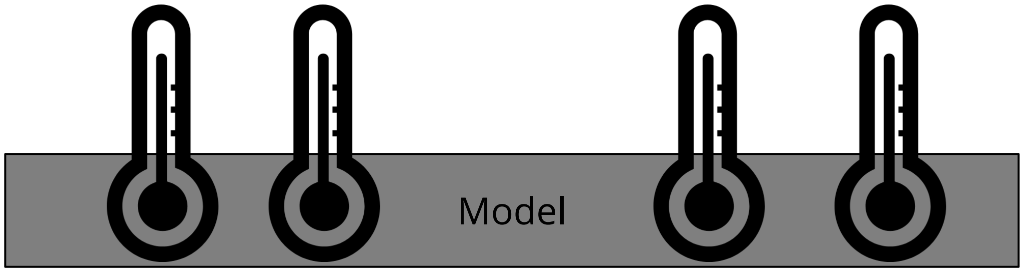
Probing

- Term coined by Guillaume Alain & Yoshua Bengio
- "Understanding intermediate layers using **linear classifier probes**." ICLR 2017
- Used linear classifiers on ResNets
- "We suggest that the reader think of those probes as thermometers used to measure the temperature simultaneously at many different locations."



Why this analogy makes sense

- Probes have access to all intermediate layers
- Probes map the representations to a continuous space
- Using probes, we get a reading for one value.



Why do we care about probing?

A: It's a simple method for peering into the black box.

Probing for contextualized word embeddings

- "What do you learn from context? Probing for sentence structure in contextualized word representations." ICLR 2019.
- Answer: mostly syntactic information. Contextualized embeddings provide gains in probe performance on syntactical tasks. (POS tagging)
- Probing for syntax.
- **Uses MLP probes.**

POS	The important thing about Disney is that it is a global $[\text{brand}]_1$. → NN (Noun)
Constit.	The important thing about Disney is that it $[\text{is a global brand}]_1$. → VP (Verb Phrase)
Depend.	$[\text{Atmosphere}]_1$ is always $[\text{fun}]_2$ → nsubj (nominal subject)
Entities	The important thing about $[\text{Disney}]_1$ is that it is a global brand. → Organization
SRL	$[\text{The important thing about Disney}]_2$ $[\text{is}]_1$ that it is a global brand. → Arg1 (Agent)
SPR	$[\text{It}]_1$ $[\text{endorsed}]_2$ the White House strategy. . . → {awareness, existed_after, . . . }
Coref.^{O}	The important thing about $[\text{Disney}]_1$ is that $[\text{it}]_2$ is a global brand. → True
Coref.^{W}	$[\text{Characters}]_2$ entertain audiences because $[\text{they}]_1$ want people to be happy. → True Characters entertain $[\text{audiences}]_2$ because $[\text{they}]_1$ want people to be happy. → False
Rel.	The $[\text{burst}]_1$ has been caused by water hammer $[\text{pressure}]_2$. → Cause-Effect(e_2, e_1)

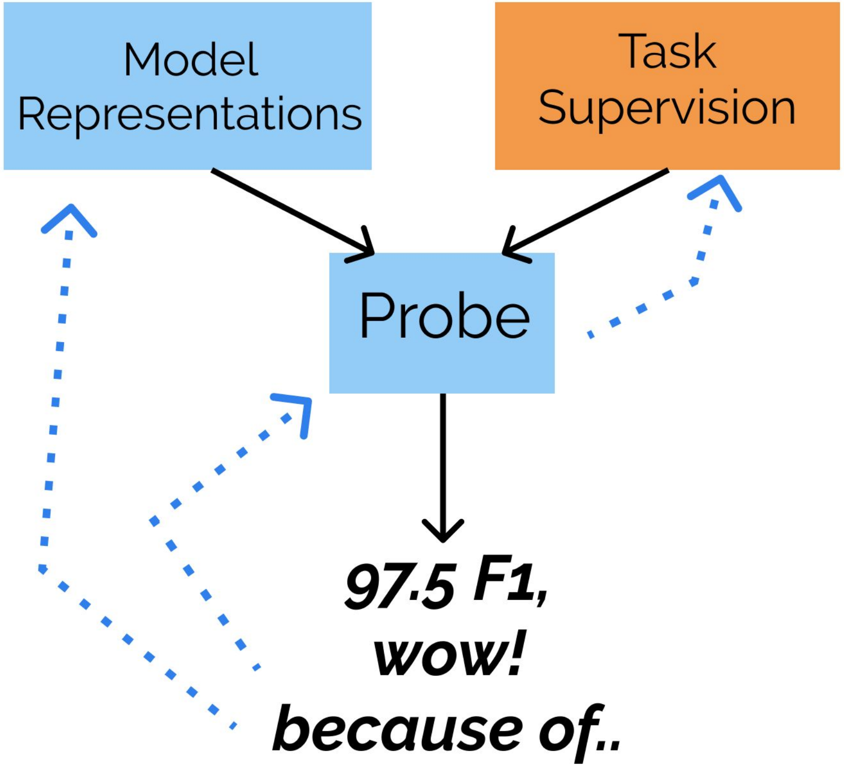
Table 1: Example sentence, spans, and target label for each task. O = OntoNotes, W = Winograd.

A summary of labeling tasks used in probing.

Probing for sentence embeddings

- Ray Mooney, ACL 2014, opening talk: "You can't cram the meaning of a whole f**king sentence into a single f**king vector!"
- "What You Can Cram into a Single Vector: Probing Sentence Embeddings for Linguistic Properties." ACL 2018.
- Probe sentence encoders for surface-level, syntactic, and semantic information.
- Result: BiLSTM encoder beats simple bag-of-vectors baseline.
- **Uses MLP probes.**

Issues with probing: attribution



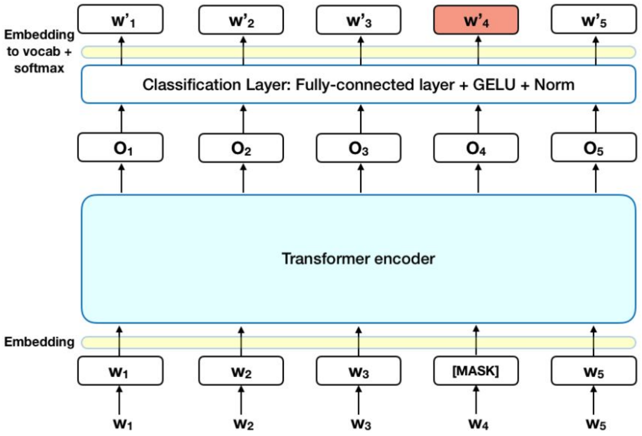
Do we attribute good probe performance to:

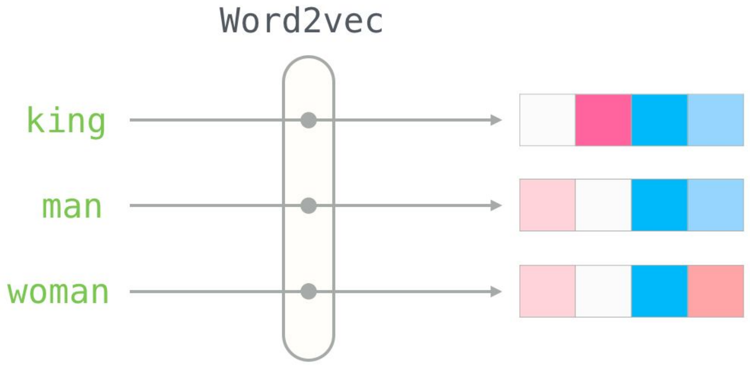
1. The upstream model's representation?
2. Training?

Figure credit: John Hewitt

Issues with probing: capacity

Representations are **lossless.**





If your probe is too complex, all you're doing is feeding embeddings to a model!

Ex: BERT would be a terrible probe.

Issues with probing

To say anything conclusive about probing, **we need metrics.**

We need to distinguish between:

1. Probes that work because of emergent linguistic representations.
2. Probes that treat deep representations as word embeddings.

→ **control tasks**

We need to quantify how reflective a probe is of its input representation.

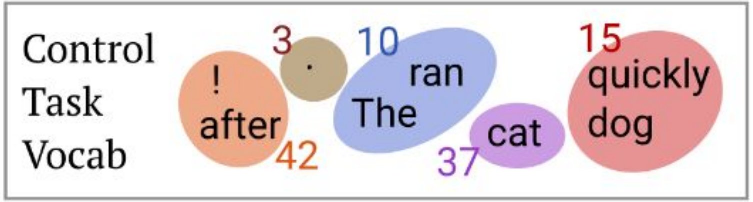
→ **selectivity**

Designing and Interpreting Probes with Control Tasks

John Hewitt and Percy Liang

Control tasks

1. Generate a label for each word in a vocabulary independently at random (**randomness**).
2. Assign this label to that word for the rest of the experiment (**structure**).



Sentence 1	The	cat	ran	quickly	.
Part-of-speech	DT	NN	VBD	RB	.
Control task	**10**	**37**	**10**	**15**	**3**
Sentence 2	The	dog	ran	after	!
Part-of-speech	DT	NN	VBD	IN	.
Control task	**10**	**15**	**10**	**42**	**42**

Control tasks are defined per task.

Control tasks

A probe dependent on linguistic representation should perform badly on control tasks.

Why: control task labels do not correspond with real linguistic knowledge.

So: a probe can only perform well if it uses word embeddings from representation to memorize control task labels.

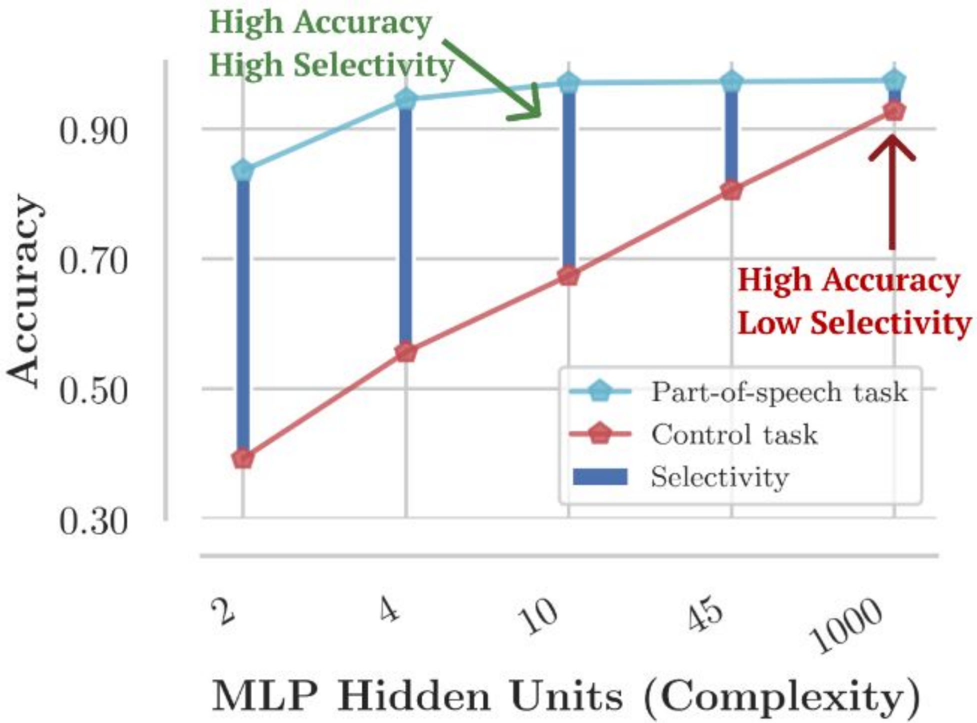
Control tasks trap probes that are too smart.

Selectivity: a balancing act

We want a probe that:

1. Has enough capacity to draw out info from representation
2. Does not have enough capacity to memorize the task

selectivity = linguistic accuracy (1) - *control accuracy* (2)

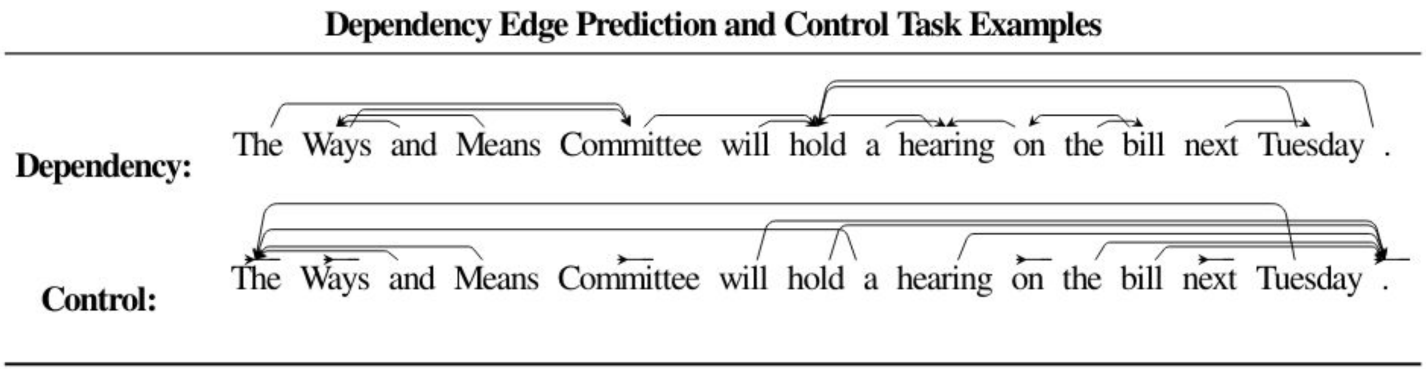


Selectivity: how reflective a probe is of its input representation.

Experiments

Setting: Tasks

1. Part of Speech tagging
    - 45 possible tags (NN, VB, etc.)
    - Assign 45 randomized control tags.
2. Dependency edge prediction
    - A lot of possibilities.
    - 3 control tags: attach to self (*i*), attach to first (*1*), or attach to last (*T*).



Setting: Probe families

$$f_\theta(h_{1:T}) = \hat{y}_{1:T}$$

1. POS tagging: 3 probes.

$$\begin{aligned}
\text{linear probe :} &\quad y_i \sim softmax(Ah_i) \\
\text{2-layer MLP (MLP-1) :} &\quad y_i \sim softmax(W_2 g(W_1 h_i)) \\
\text{3-layer MLP (MLP-2) :} &\quad y_i \sim softmax(W_3 g(W_2 g(W_1 h_i)))
\end{aligned}$$

2. Dependency edge prediction: 3 probes, replace linear with bilinear.

$$\text{bilinear probe :} \quad y_i \sim softmax(\mathbf{h_{1:T}}^T A h_i)$$

Setting: Complexity Control (= Regularization)

Probes can't be too complex.

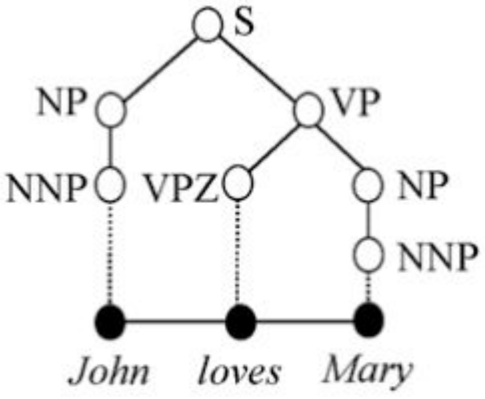
$$y_i \sim softmax(Ah_i)$$

1. Rank/hidden dimensionality constraint.
   - Factorize A = LR, force L to have dimension *l*
   - Force MLPs to have hidden state size *l*
2. Dropout (temporarily zero out nodes).
3. Constraining the number of training examples.
4. L2 regularization (weight decay).
5. Early stopping.

Dataset

Penn Treebank: dataset of sentences from Wall Street Journal

- Sentences labeled with parts of speech and dependency trees.



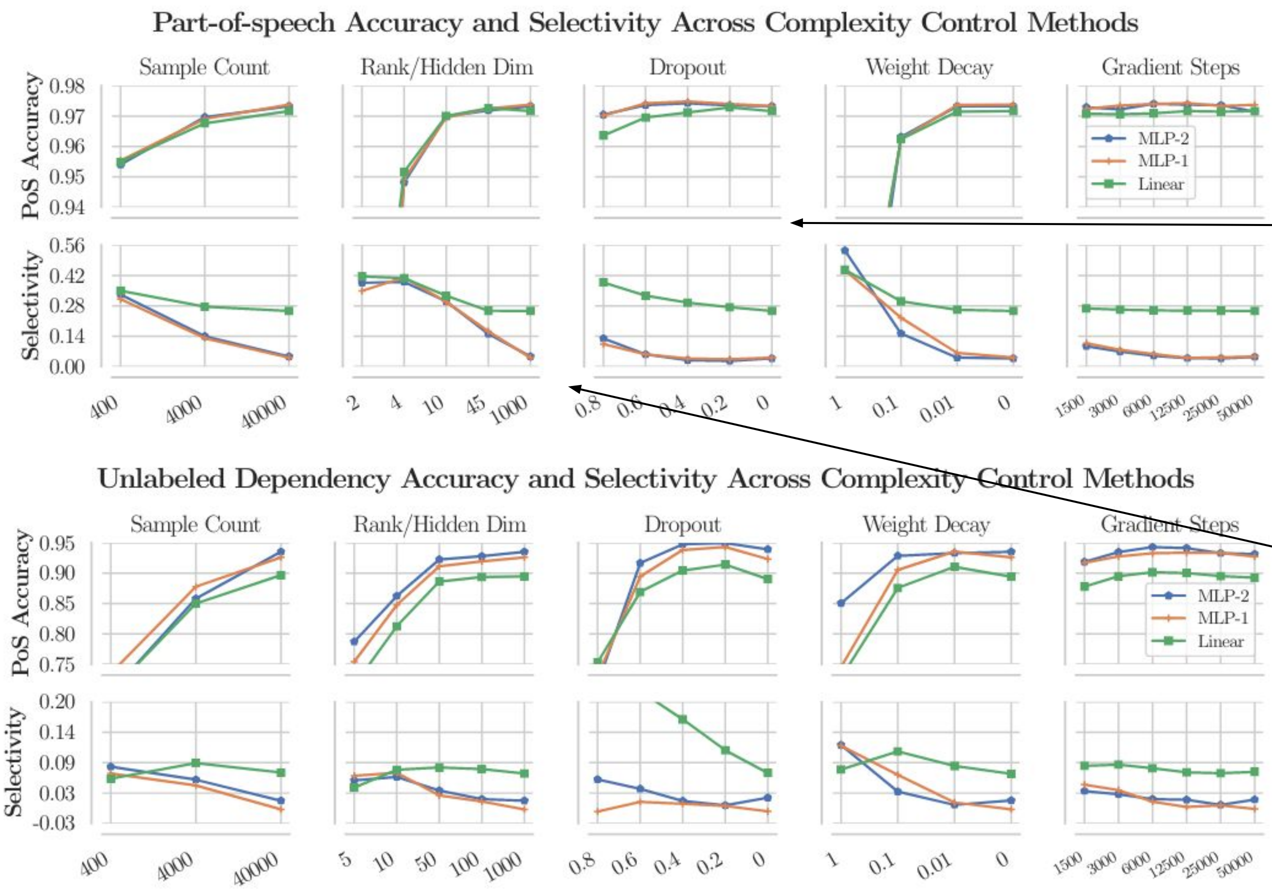
Probe	PoS	Ctl	Select.	Dep	Ctl	Select.
Probes with Default Hyperparameters						
Linear	97.2	71.2	26.0	-	-	-
Bilinear	-	-	-	89.0	82.4	6.6
MLP-1	97.3	92.8	4.5	92.3	93.0	-0.7
MLP-2	97.3	93.2	4.2	93.9	92.0	1.9
Probes with 0.4 Dropout						
Linear	97.1	67.3	29.8	-	-	-
Bilinear	-	-	-	90.4	73.7	16.7
MLP-1	97.5	93.4	4.1	93.8	93.1	0.7
MLP-2	97.4	94.1	3.4	94.7	93.5	1.3
Probes Designed with Control Tasks						
Linear	97.0	64.0	33.0	-	-	-
Bilinear	-	-	-	91.0	83.1	7.9
MLP-1	97.2	80.6	16.6	90.5	84.3	6.2
MLP-2	97.2	81.7	15.4	92.8	89.8	3.0

Probes with hyperparameters tuned for selectivity.

Results for different probe families, under various hyperparameter settings.

	Parts-of-speech		Dependencies	
	Acc.	Select.	Acc.	Select.
Linear	97.2	**25.5**	-	-
Bilinear	-	-	89.4	**16.6**
MLP-1layer	**97.3**	4.65	**92.5**	3.35
MLP-2-2layer	**97.3**	5.25	**93.4**	4.17

Figure credit: John Hewitt



Dropout doesn't really help.

Constraining rank seems to help

Effectiveness of various regularization strategies. What we want: big increase in accuracy, small decrease in selectivity.

How hard is it to find selective probes?

Results:

- Dropout and early stopping don't help selectivity
- Constraining hidden state dimensionality is effective!
  - Used MLP hidden state size of 10 for POS and 50 for dependency head prediction

Author conclusions:

- Current probes are needlessly overparameterized! They have too much capacity.
- The most selective probes are linear or bilinear models.
- MLPs have the best accuracy on dependency edge prediction.
  - → some syntactical info can't be extracted by a bilinear probe.

POS error analysis

- Linear models tend to classify adjective-noun pairs as noun-noun pairs

Kan.-based/JJ National/NNP Pizza/NNP

rental/JJ equipment/NN

- MLPs tend to pluralize singular nouns.

Environmental/NNP Systems/NNP Co./NNP

Cara/NNP Operations/NNP Co./NNP

7.8/CD %/NN stake/NN in/IN Dataproducts/NNP

- Hypothesis: MLPs have enough capacity to get confused by the 's'.

Selectivity and layer differences

Claim: the first layer of ELMo (ELMo1) is better for POS tagging than ELMo2.

Hewitt & Liang: not so fast! ELMo1 is closer to a straight word representation.

Notes for the next slide:

- Recall that ELMo runs a character CNN over the words before feeding into biLSTMs.
- As a baseline, Hewitt & Liang run an untrained biLSTM and call this representation **Proj0.**

POS tagging: We see an increase in selectivity, for a comparably smaller decrease in accuracy.

Implication: Because ELMo1 is closer to a word representation, probes on ELMo1 are leveraging the word identity and not the encoded linguistic knowledge.

Part-of-speech Tagging				
	Linear		MLP-1	
Model	Accuracy	Selectivity	Accuracy	Selectivity
Proj0	96.3	20.6	97.1	1.6
ELMo1	97.2	26.0	97.3	4.5
ELMo2	96.6	31.4	97.0	8.8

Dependency Edge Prediction				
	Bilinear		MLP-1	
Model	Accuracy	Selectivity	Accuracy	Selectivity
Proj0	79.9	-4.3	86.5	-9.0
ELMo1	89.7	6.7	92.5	-1.0
ELMo2	84.5	6.2	89.5	1.4

Probe performance on different layers of ELMo

Linear Probe Parts-of-speech

	Acc.	Select.
Proj0	96.3	20.6
ELMo1	**97.2**	26.0
ELMo2	96.6	**31.4**
ELMo1-ELMo2	+0.6	-5.4
ELMo2-Proj0	+0.3	+10.8

Figure credit: John Hewitt

"Without considering selectivity, [we might think] that ELMo2 encodes nothing about part-of-speech, since it doesn't beat the Proj0 baseline.

"Taking selectivity into account, we see that probes on ELMo2 are unable to rely on word identity features like those on Proj0. To achieve high accuracy, they **must** rely on emergent properties of the representation."

-Hewitt & Liang

Summary

- Use **control tasks** to identify models using representations as word embeddings.
- Probes should be **selective**. They should perform poorly on the control task.
- Linear and bilinear probes are the most selective.
- Many probes nowadays are too powerful.

Q1: In Hewitt and Liang et al 2019, why do they claim that linear and bilinear classifiers work better as probes than multi-layer perceptrons?

Linear and bilinear classifiers work better because they are more selective. Selective probes better reflect linguistic properties of the representation. MLPs tend to be too overparameterized, allowing them to memorize control-task mappings.

Where probing is going

- Designing and Interpreting Probes with Control Tasks:
  - appeared on arXiv September 2019
  - published November 2019, EMNLP best paper runner-up
- Probing is a nascent technique: there's no consensus on best practice.
- Thoughts? Is probing a good technique?

References

Hewitt, John, and Percy Liang. “Designing and Interpreting Probes with Control Tasks.” ArXiv:1909.03368 [Cs], Sept. 2019. arXiv.org, http://arxiv.org/abs/1909.03368.

Alain, Guillaume, and Yoshua Bengio. “Understanding Intermediate Layers Using Linear Classifier Probes.” ArXiv:1610.01644 [Cs, Stat], Nov. 2018. arXiv.org, http://arxiv.org/abs/1610.01644.

Conneau, Alexis, et al. What You Can Cram into a Single Vector: Probing Sentence Embeddings for Linguistic Properties. May 2018. arxiv.org, https://arxiv.org/abs/1805.01070v2.

Tenney, Ian, et al. What Do You Learn from Context? Probing for Sentence Structure in Contextualized Word Representations. May 2019. arxiv.org, https://arxiv.org/abs/1905.06316v1.

https://nlp.stanford.edu/~johnhew/interpreting-probes.html

Sturmfels, et al., "Visualizing the Impact of Feature Attribution Baselines", Distill, 2020.