Lecture 19: Debugging and Probing

What's going on with my model?!

Harvard IACS

Chris Tanner

Featuring the hit song, "Debuggin' out"

by A Tribe Called Quest (ATCQ)

ANNOUNCEMENTS

- No more homework
- HW3 and Quiz 6 and Quiz 7 and Phase 3 are being graded
- Research Project Phase 4 will be a soft-assessment

Many of today's slides are based on, inspired by,
or directly from Mohit Iyyer (UMass Amherst),
Graham Neubig (CMU), Sam Bowman (NYU),
Yonatan Belinkov (Technion)

Outline

- Model Debugging
- Interpretable Evaluation
- Interpreting Predictions (Probing)
- Workshop time

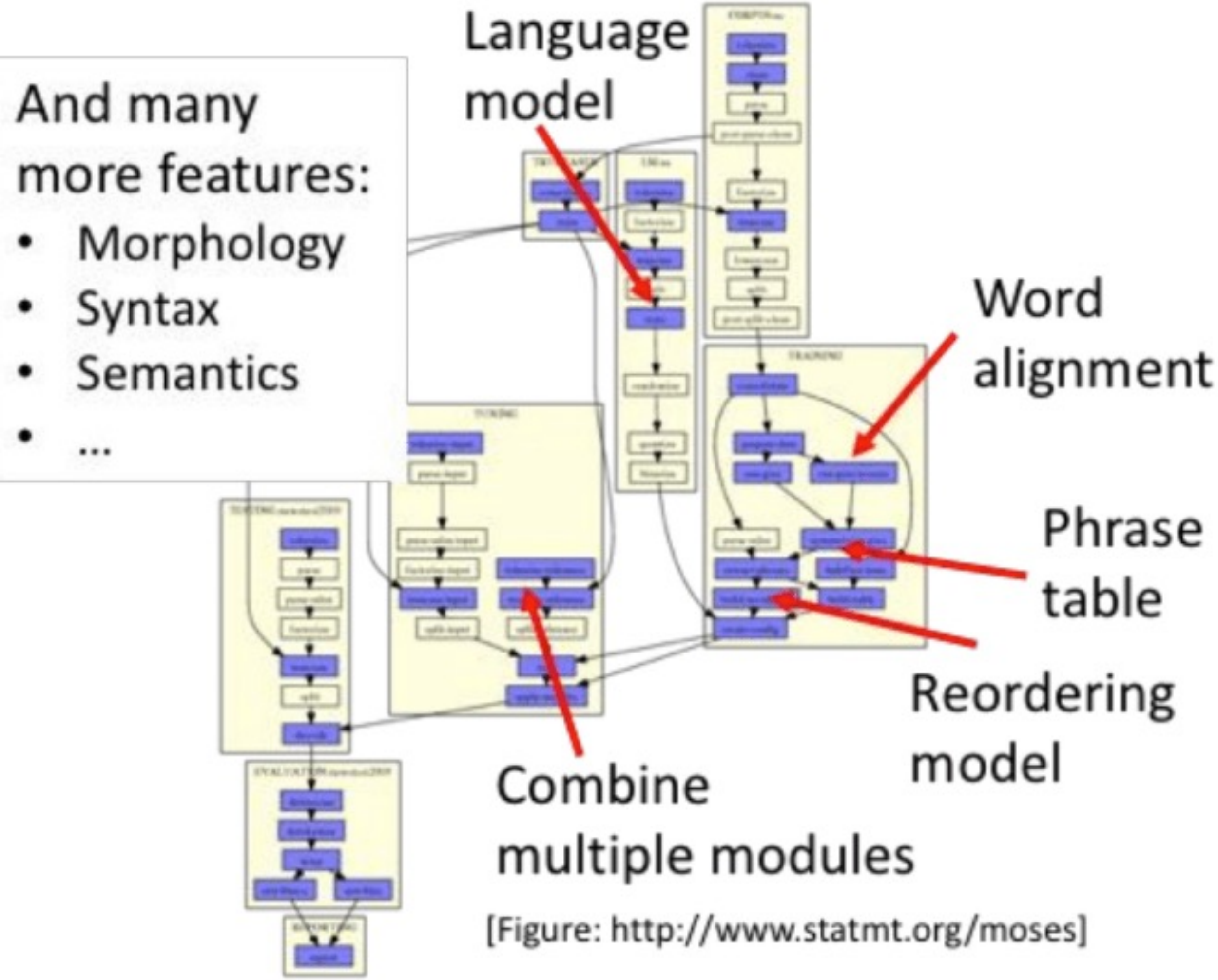
Outline

- Model Debugging
- Interpretable Evaluation
- Interpreting Predictions (Probing)
- Workshop time

Premise

In the "old days" (aka pre-2013), NLP models were comprised of many hand-engineered features.

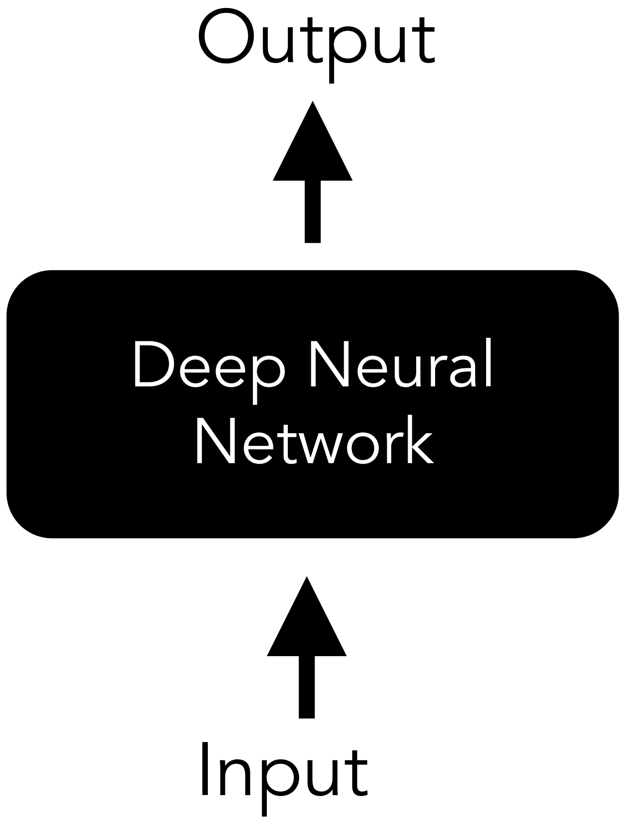
Debugging was straight-forward.



[Figure: http://www.statmt.org/moses]

The black box

- Do neural nets learn any kind of interpretable structure?
- Can we explain how well they generalize?
- When will they succeed and fail?
- Why do they make particular decisions?



Why should we care?

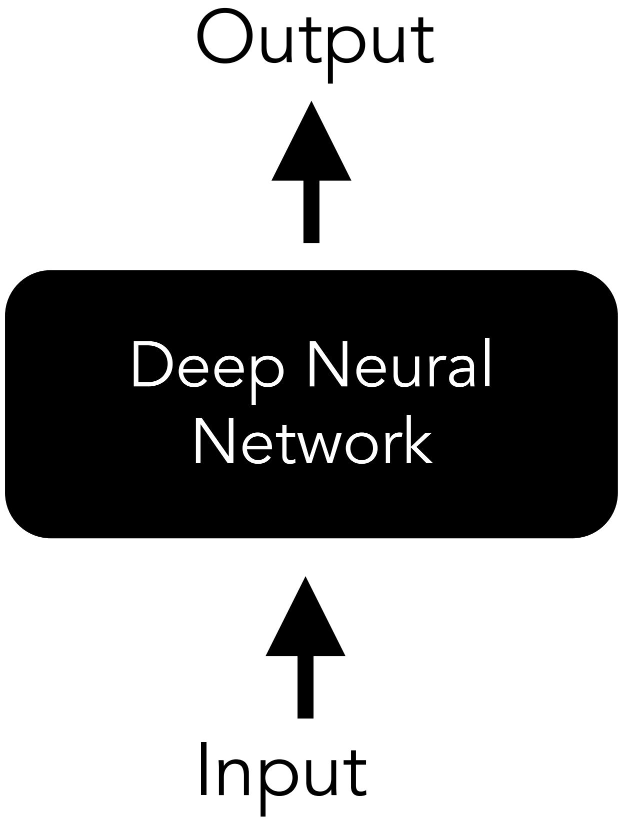
Most of deep learning research:
- Trail and error, shots in the dark
- Better understanding → better systems

Accountability, trust, and bias in machine learning
- "Right to explanation", EU regulation
- Life threatening situations: healthcare, autonomous cars
- Better understanding → more accountable systems

NNs aid scientific study of language (Linzen 2019)
- Models of human language acquisition and processing
- Better understanding → better understanding of humans

Situation

- You've developed a deep learning NLP model
- Code looks correct to you
- It has low accuracy or makes odd errors
- What do you do?



Slide based on, inspired by, or directly from Graham Neubig

Debugging

- Debugging allows you to identify problems in your assumptions or implementation
- Models are often **complicated and opaque**
- **Everything is a hyperparameter**
  (e.g., network size, model variations, batch size, optimization, learning rate)
- Non-convex, stochastic optimization has **no guarantee of converging loss**

Possible Causes

Debug incrementally!

- **Training time problems**
  - Lack of model capacity
  - Inability to train model properly
  - Training time bug
- **Decoding time bugs**
  - Disconnect between test and decoding
  - Failure of search algorithm
- **Overfitting**
- **Mismatch between optimized function and eval**

Training

- Look at the **loss function** calculated on the **training set**
  - Is the loss function going down?
  - Is it going down basically to zero if you run training long enough (e.g. 20-30 epochs)?
  - If not, does it go down to zero if you use very small datasets?

Deliberately try to overfit

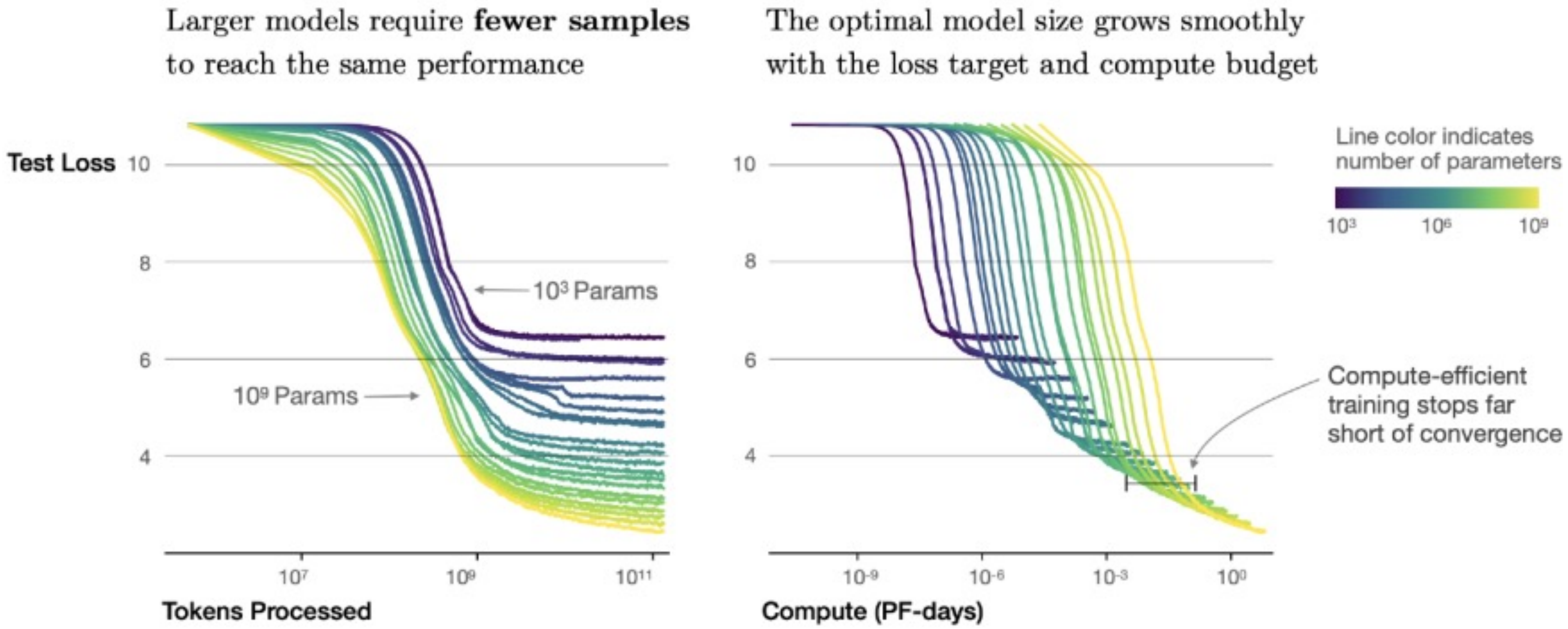
Model expressivity

Larger models tend to perform better, esp. when pre-trained (e.g. Raffel et al. 2020)

Model	GLUE Average	CoLA Matthew's	SST-2 Accuracy	MRPC F1	MRPC Accuracy	STS-B Pearson	STS-B Spearman
Previous best	89.4[a]	69.2[b]	97.1[a]	**93.6**[b]	**91.5**[b]	92.7[b]	92.3[b]
T5-Small	77.4	41.0	91.8	89.7	86.6	85.6	85.0
T5-Base	82.7	51.1	95.2	90.7	87.5	89.4	88.6
T5-Large	86.4	61.2	96.3	92.4	89.9	89.9	89.2
T5-3B	88.5	67.1	97.4	92.5	90.0	90.6	89.8
T5-11B	**90.3**	**71.6**	**97.5**	92.8	90.4	**93.1**	**92.8**

Model expressivity

Larger models can learn with fewer steps (Kaplan et al. 2020, Li et al. 2020)



Model expressivity

- If increasing model size doesn't help, you may have an optimization problem
- Check your
  - **optimizer** (Adam? standard SGD?)
  - **learning rate** (is the rate you're using standard, are you using decay?)
  - **initialization** (uniform? Glorot?)
  - **minibatching** (are you using sufficiently large batches?)
- Pay attention to these details when replicating previous work

Training/Test Discrepancies

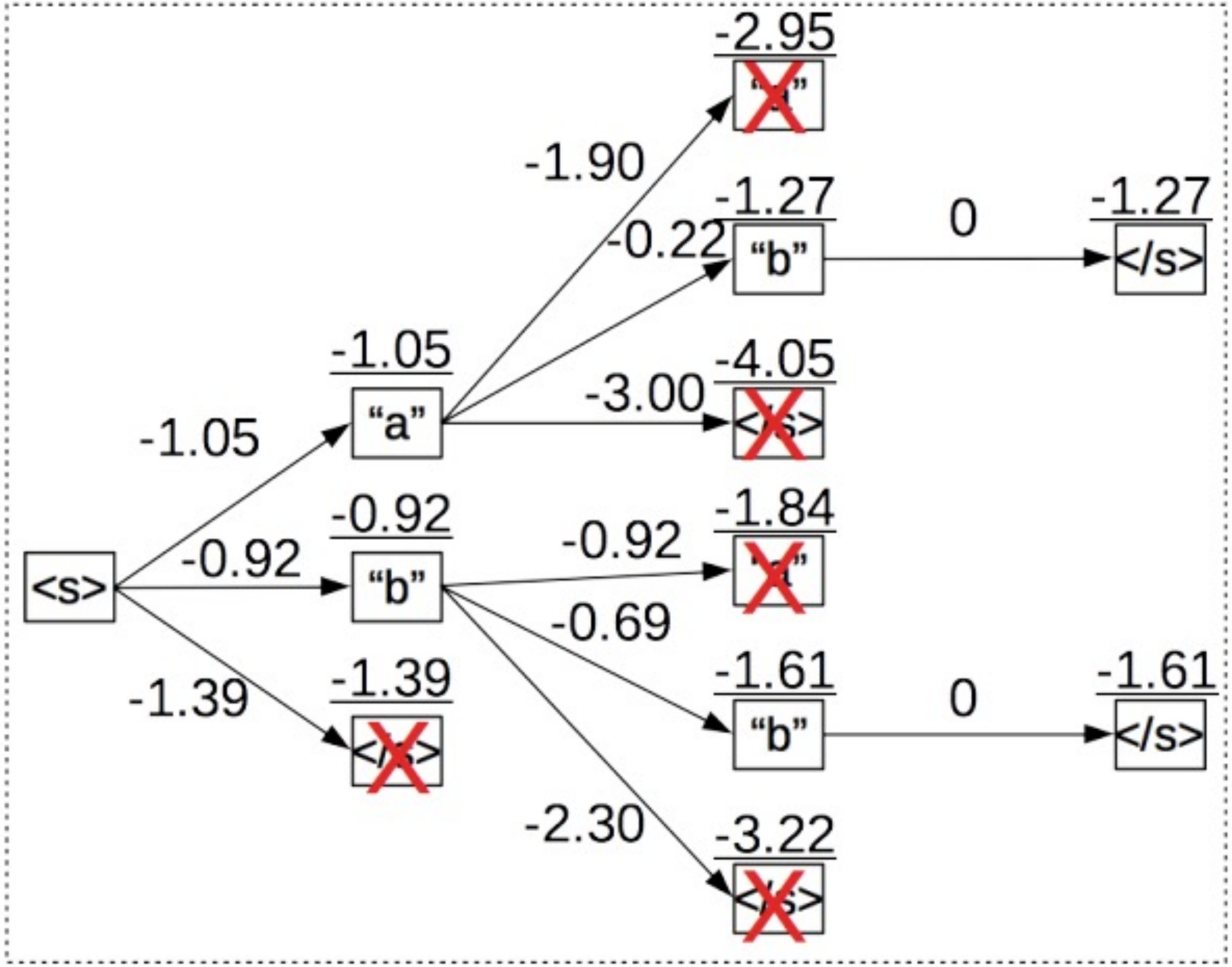
- Usually your loss calculation and prediction will be implemented in different functions
- Especially true for structured prediction models (e.g. encoder-decoders)
- Like all software engineering: **duplicated code is a source of bugs**!
- Also, usually loss calculation is minibatched, generation not.

Debugging Mini-batching

- Debugging mini-batched loss calculation
  - Calculate loss with **large batch size** (e.g. 32)
  - Calculate loss for **each sentence individually and sum**
  - The values should be the same (modulo numerical precision)
- Create a unit test that tests this!

Beam Search

Instead of picking the single-most probable word, maintain several paths

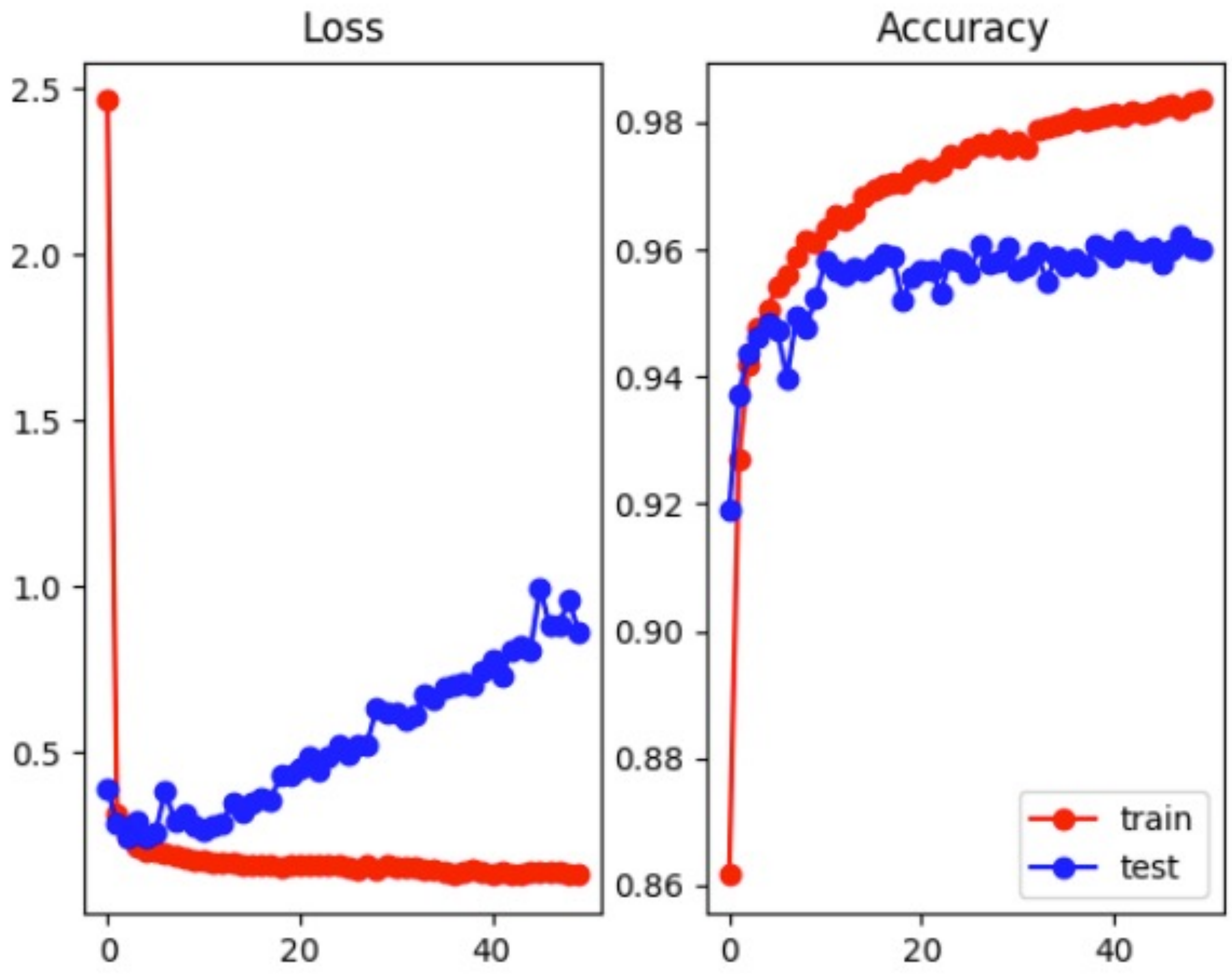


Beam Search

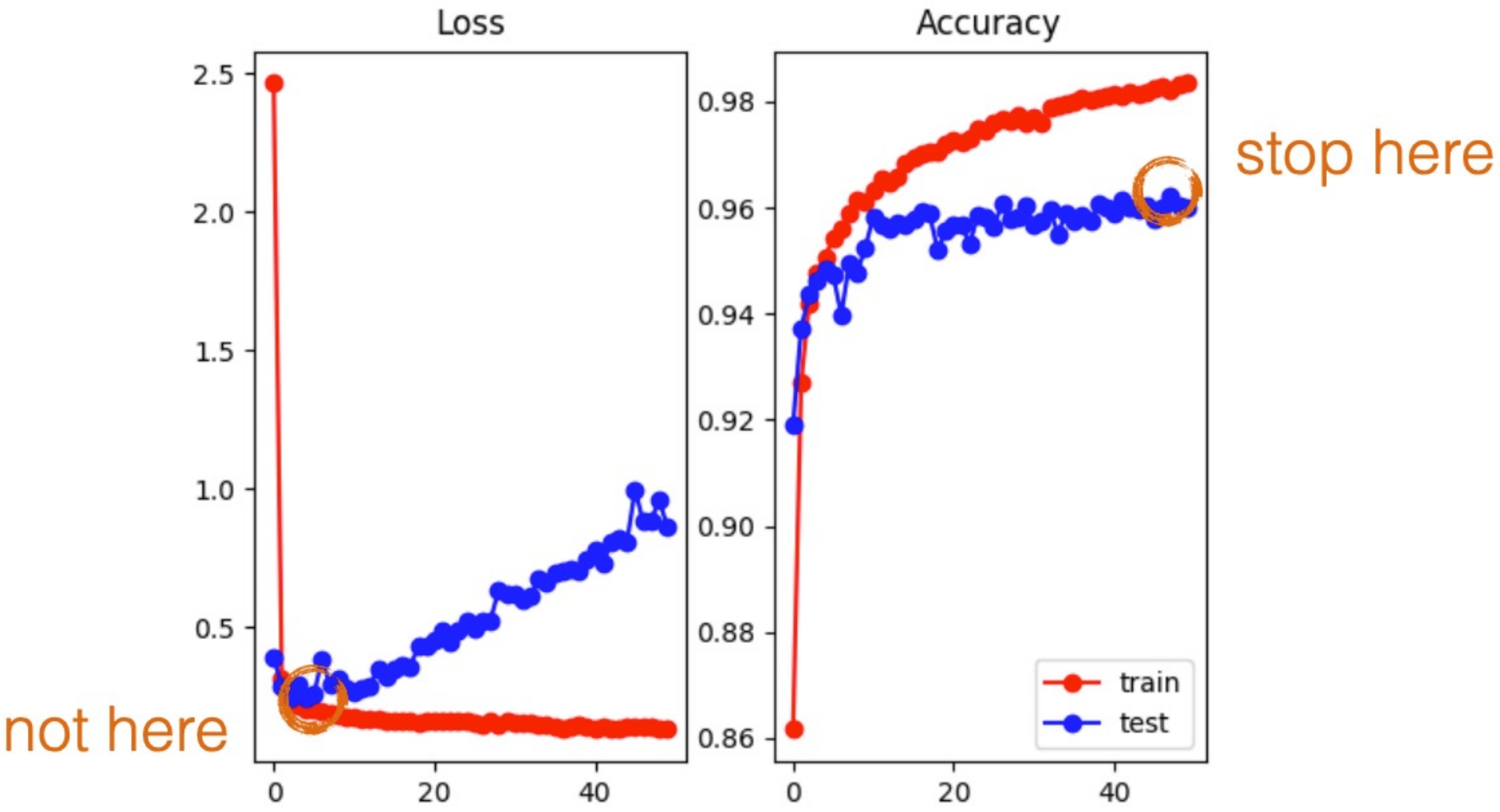
- As you make search better, the model score should get better (almost all the time)
- Search w/ varying beam sizes and make sure you get a better overall model score with larger sizes
- Create a unit test testing this!

Loss function vs Evaluation metric

- Very common to optimize for maximum likelihood for training
- Likelihood isn't necessarily correlated with accuracy
- Why?



Early Stopping with Evaluation Metric



Outline

- Model Debugging
- Interpretable Evaluation
- Interpreting Predictions (Probing)
- Workshop time

Outline

- Model Debugging
- Interpretable Evaluation
- Interpreting Predictions (Probing)
- Workshop time

Look at your data

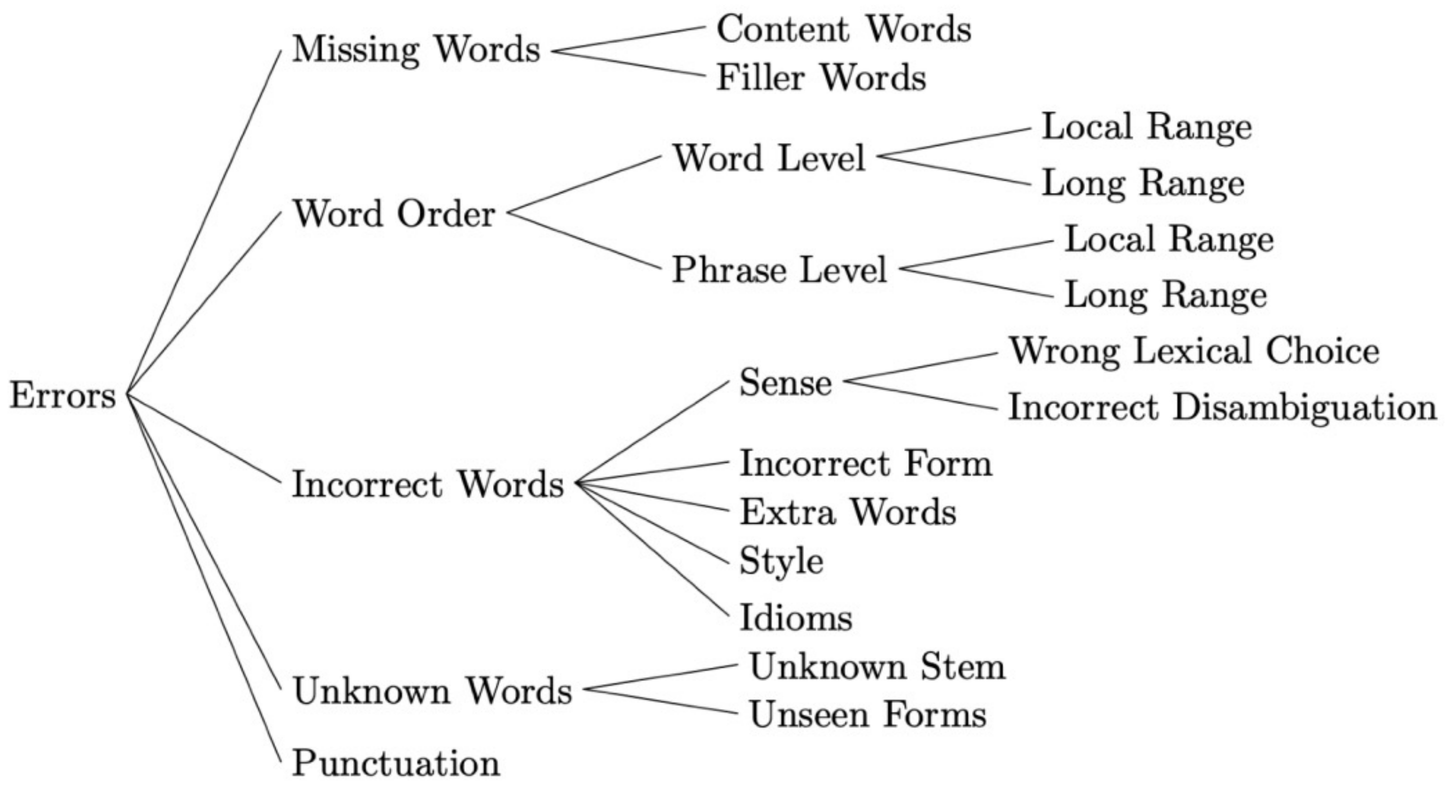
- Both bugs and research directions can be found by **looking at your model outputs**

- The first word of the sentence is dropped every generation
  > went to the store yesterday
  > bought a dog
  → implementation error?

- The model is consistently failing on named entities
  → need a better model of named entities?

Inspect and categorize your errors

- **Look at 100-200 errors**
- Try to **group them** into a typology (pre-defined or on the fly)
- Example: Vilar et al. (2006)



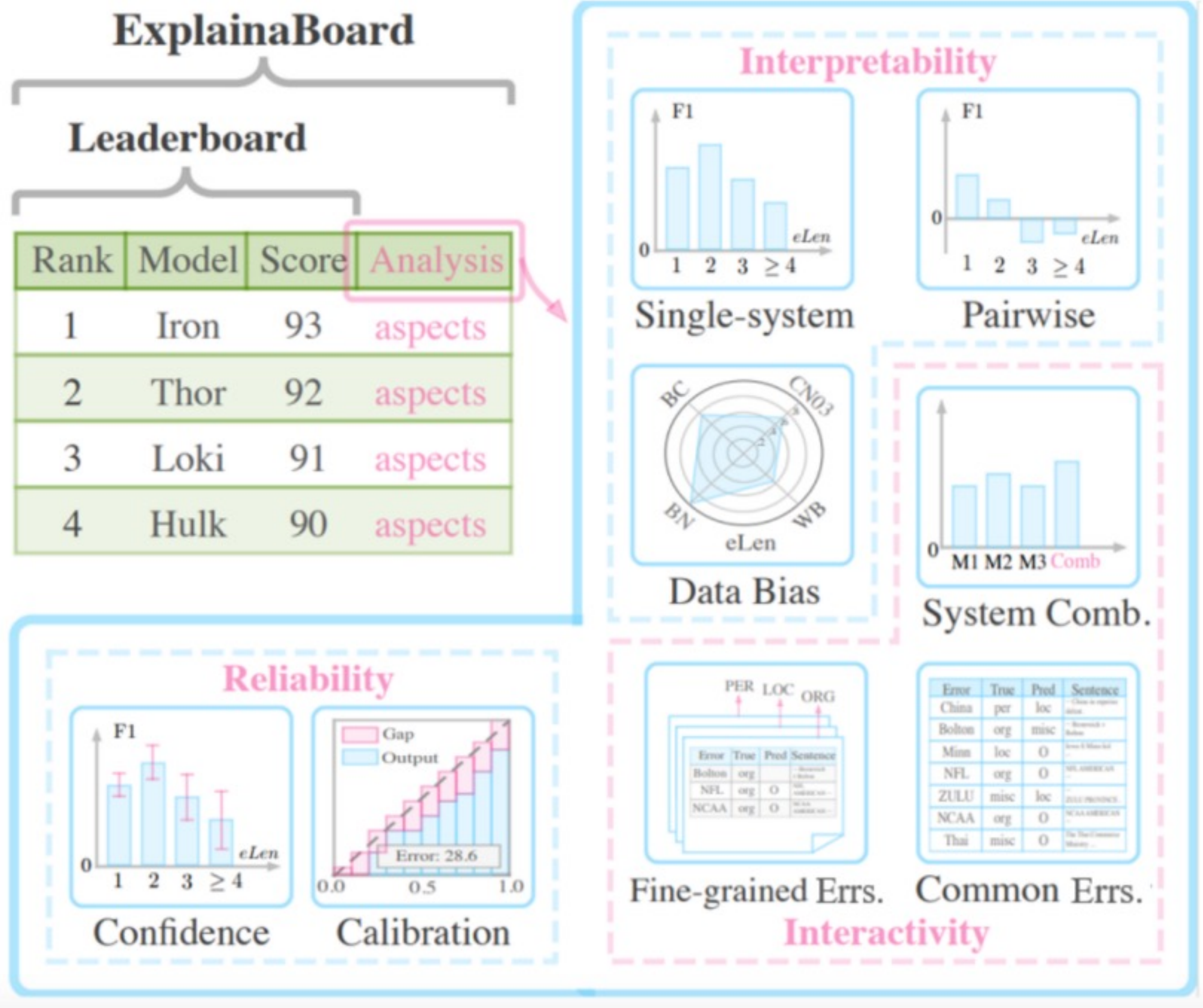
Quantitative Analysis

- Measure gains quantitatively. What is the phenomenon you chose to focus on? Is that phenomenon getting better?
  - **You focused on low-frequency words:** is accuracy on low frequency words increasing?
  - **You focused on syntax:** is syntax or word ordering getting better, are you doing better on long-distance dependencies?
  - **You focused on search:** how many search errors are being reduced?

Example: ExplainaBoard

- Summary of many different NLP tasks from a variety of aspects

http://explainaboard.nlpedia.ai/



Outline

- Model Debugging
- Interpretable Evaluation
- Interpreting Predictions (Probing)
- Workshop time

Outline

- Model Debugging
- Interpretable Evaluation
- Interpreting Predictions (Probing)
- Workshop time

The big picture

Understanding the **general properties of the model**

- Analysis and debugging
- Easier, generally used to guide engineering work or answer specific scientific questions

Understanding the properties of a model applied to a **specific example**

- Explainability and interpretability
- Harder, generally used to validate a model's decision in high-stakes situations (e.g., medical, legal, financial, etc)
- The EU has the 'right to explanation' for computational models used to make decisions about people

Slide based on, inspired by, or directly from Sam Bowman

Motivation

- You want to know which words were used in making a classification decision to verify its accuracy.
- You want to know whether your model has legitimately learned a difficult pattern, or if it’s focused on spurious correlations.
- You want to understand what information a pre-trained model has captured internally

Example

- **Task:** predict probability of death for patients with pneumonia
- **Why**: so that high-risk patients can be admitted, low risk patients can be treated as outpatients
- Rule based classifier

HasAsthma(X) —> LowerRisk(X)

Example from Caruana et al.

LIME

Linear models are very interpretable

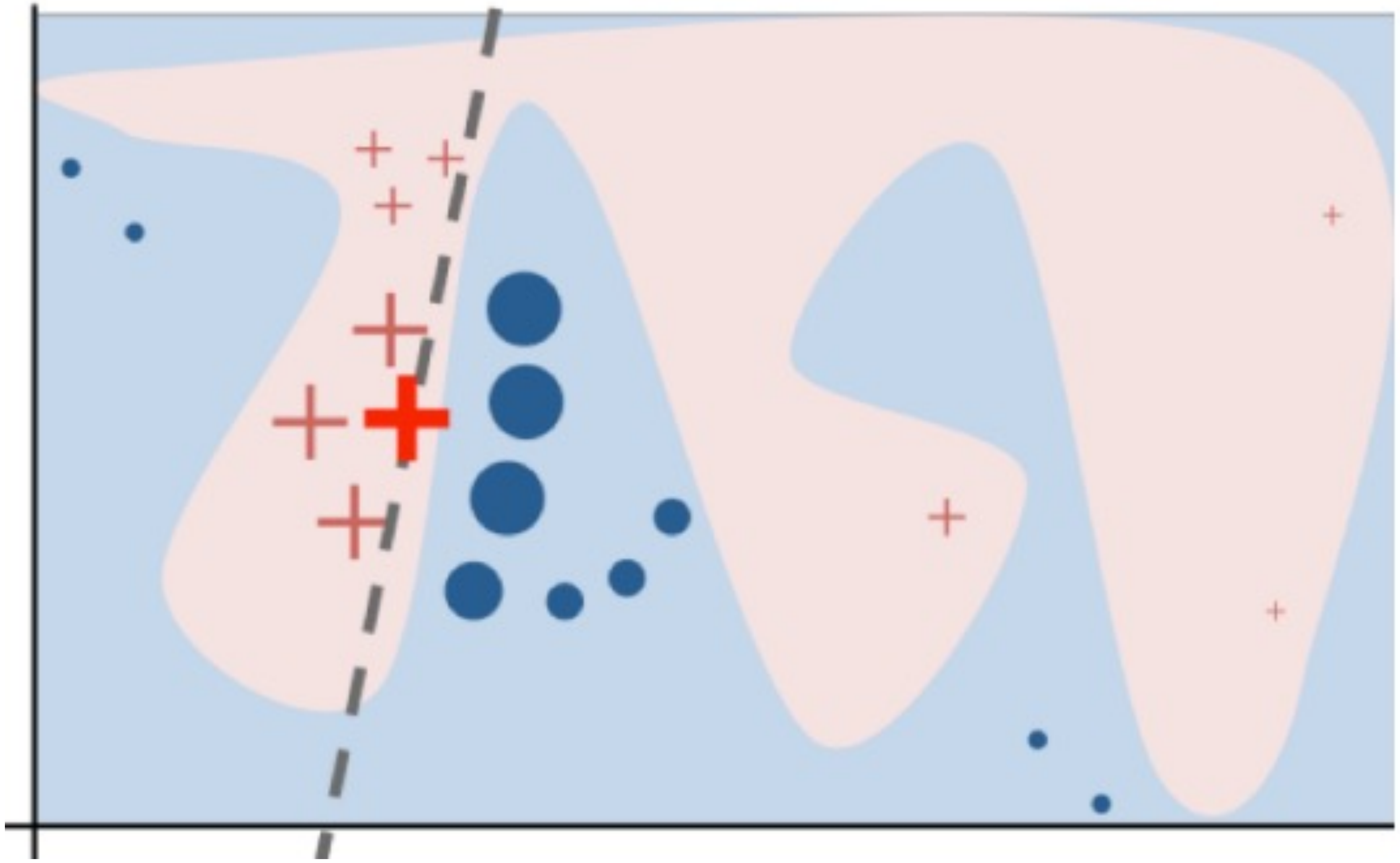
$$f(\underset{\text{House}}{\mathbf{x}}) = \overset{2.0}{w_1}\underset{\text{Area}}{\mathbf{x}_1} + \overset{0.01}{w_2}\underset{\text{Population Density}}{\mathbf{x}_2}$$

Estimates Rent

- How the answer is computed? (mechanistic details)
- Relative importance of each feature?
- How did we end up with these parameters?
  - What was the training objective?
  - What was the data? Which city? Is it representative?

LIME

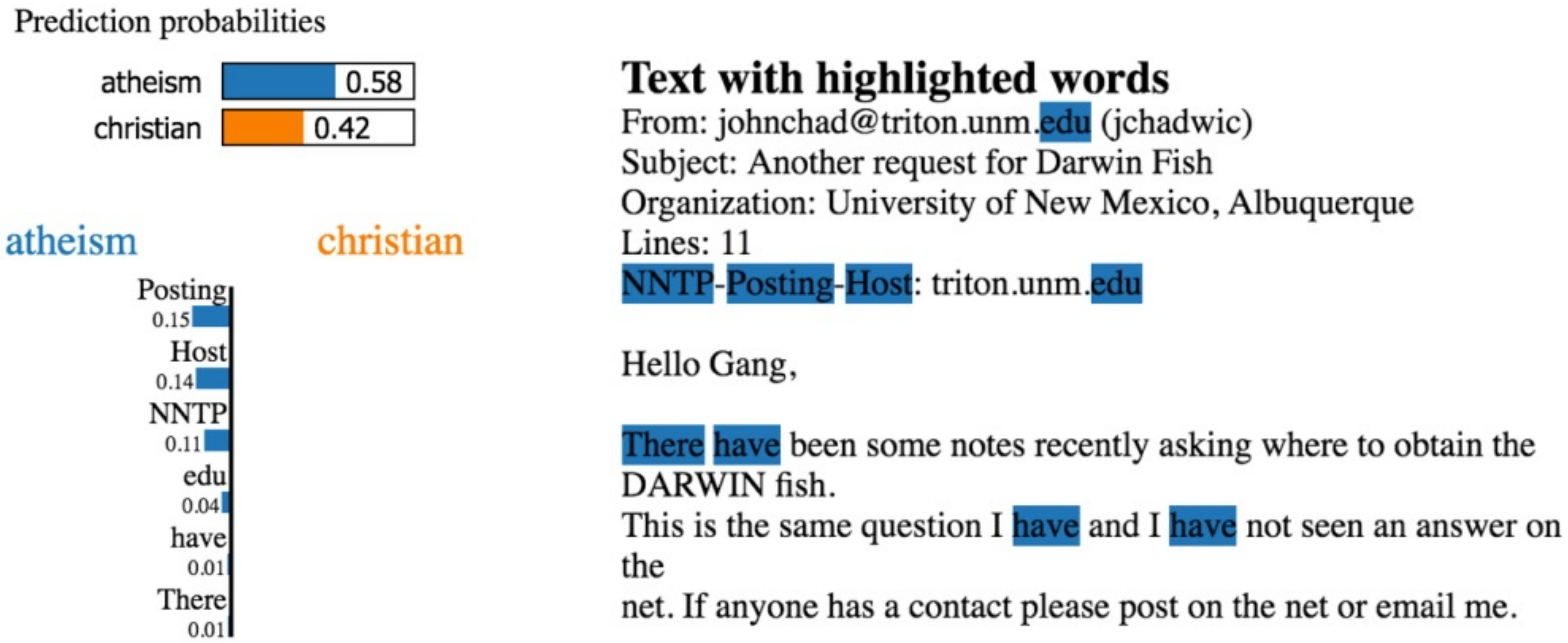
Select a specific neighborhood of data and a subset of the features



Slide based on, inspired by, or directly from Graham Neubig

LIME

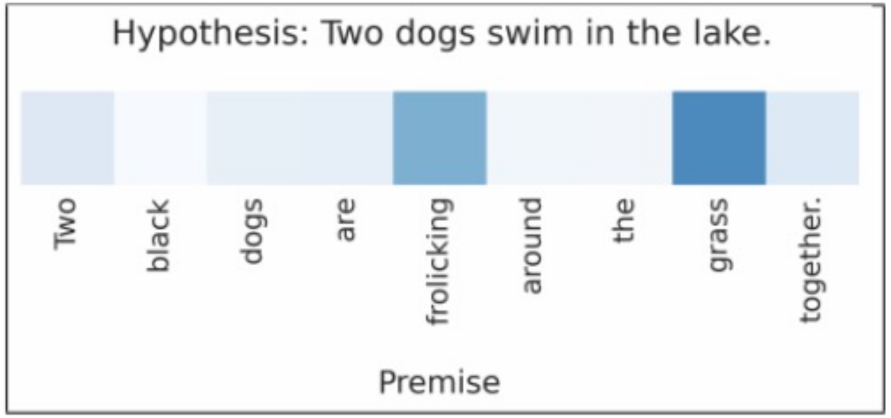
Select a specific neighborhood of data and a subset of the features



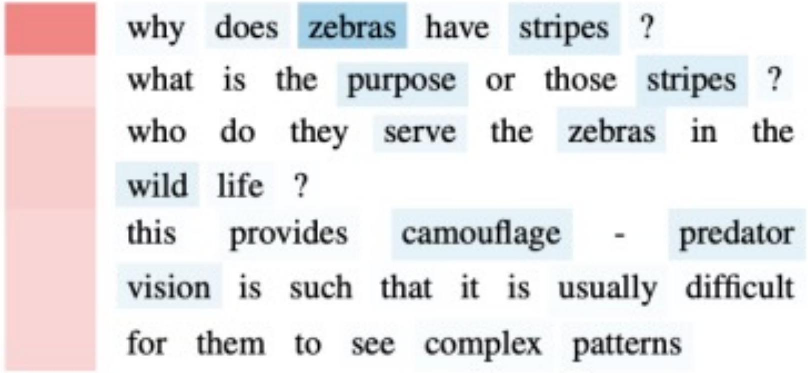
Ribeiro et al, KDD 2016

Slide based on, inspired by, or directly from Graham Neubig

Inspecting Attention



Entailment
Rocktäschel et al, 2015

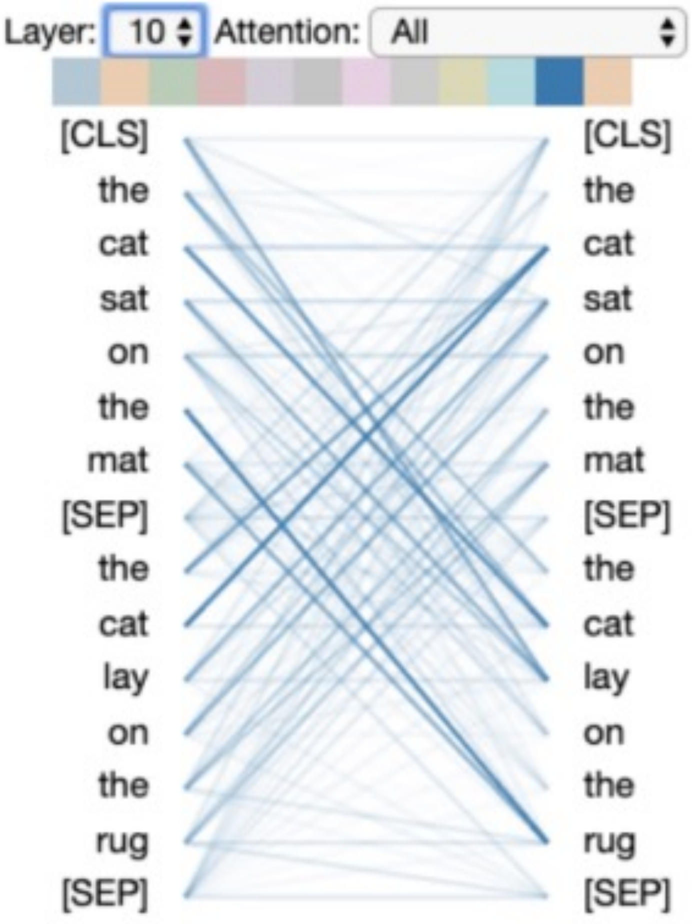


Document classification
Yang et al, 2016

Inspecting Attention

A stop sign is on a road with a mountain in the background.

Image captioning
Xu et al, 2015



BERTViz
Vig et al, 2019

Attention

Does Attention answer all of our questions?

Provides all the insights we want?

Attention

LSTMVis
Childbook with TopK
source
states::states1
pos
115
Selection
threshold
0.3
mask
zoom
t
i1
i2
a
c
d
e1
e2
add meta track
37 149 190 197 269 278 320
Matching
match
fast
precise
mask
stats
meta
match count
ner
pos
g1
h
g2
match count
pos
7.0000
ADJ
b
f

Attention is not Explanation

Sarthak Jain
Northeastern University
jain.sar@husky.neu.edu

Byron C. Wallace
Northeastern University
b.wallace@northeastern.edu

1. Attention is only mildly correlated with other importance score techniques

2. Counterfactual attention weights should yield different predictions, but they do not

Attention is not not Explanation

Sarah Wiegreffe*
School of Interactive Computing
Georgia Institute of Technology
saw@gatech.edu

Yuval Pinter*
School of Interactive Computing
Georgia Institute of Technology
uvp@gatech.edu

"Attention *might* be an explanation."

- Attention scores can provide a (plausible) explanation not **the explanation.**
- Attention is not explanation if you don't need it
- Agree that attention is indeed manipulable,

BERTology

studying the inner working of large-scale Transformer language models like BERT

- what are captured in different model components, e.g., attention / hidden states?

Tools

BERTology - HuggingFace's Transformers
https://huggingface.co/transformers/bertology.html

- accessing all the hidden-states of BERT
- accessing all the attention weights for each head of BERT
- retrieving heads output values and gradients

Tools

AllenNLP Interpret

https://allennlp.org/interpret

Allen Institute for AI

AllenNLP

Simple Gradients Visualization

See saliency map interpretations generated by visualizing the gradient.

Saliency Map:

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

Mask 1 Predictions:

47.1% **nurse**

16.4% **woman**

10.0% **doctor**

3.4% **mother**

3.0% **girl**

Findings

Are Sixteen Heads Really Better than One? Michel et al., NeurIPS 2019

> large percentage of attention heads can be removed at test time without significantly impacting performance

What Does BERT Look At? An Analysis of BERT's Attention, Clark el al., BlackBoxNLP 2019

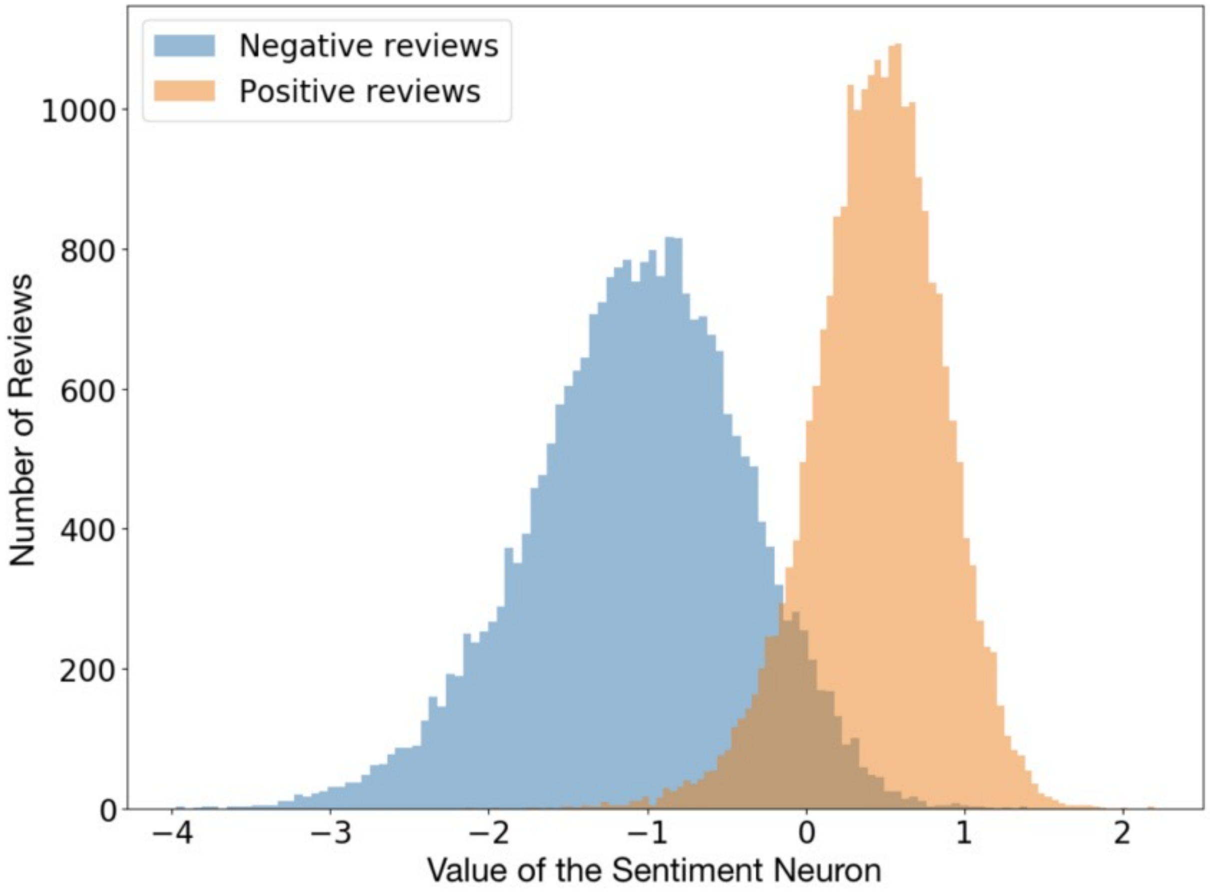
> substantial syntactic information is captured in BERT's attention

What if we fall back to just single neurons?

https://openai.com/blog/unsupervised-sentiment-neuron/

Sentiment neuron

While training the linear model with L1 regularization, we noticed it used surprisingly few of the learned units. Digging in, we realized there actually existed a single "sentiment neuron" that's highly predictive of the sentiment value.



The sentiment neuron within our model can classify reviews as negative or positive, even though the model is trained only to predict the next character in the text.

Slide based on, inspired by, or directly from Mohit Iyyer

Probing

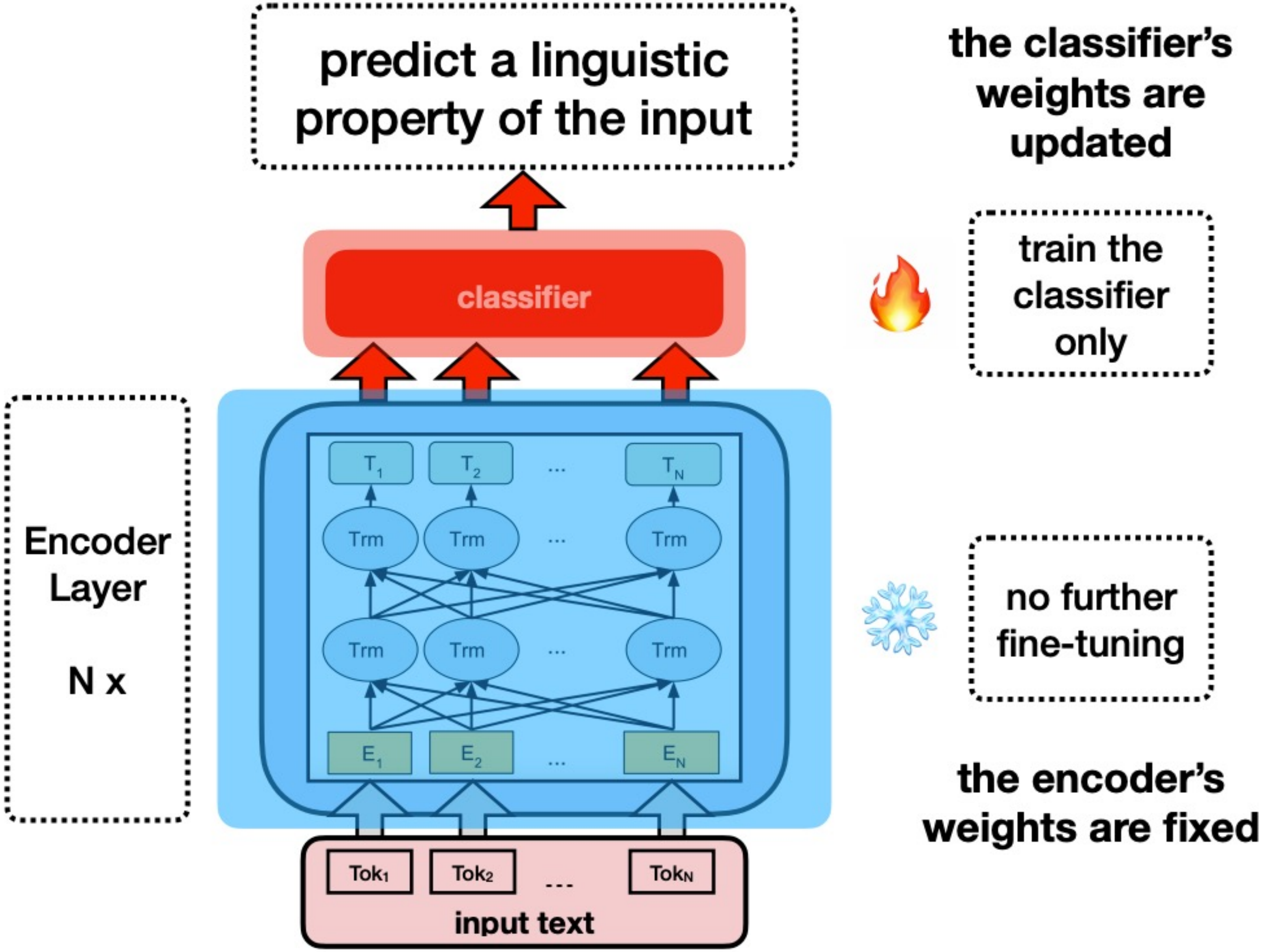
Given an encoder model (e.g., BERT) pretrained on a certain task, we use the representations it produces to train a classifier (without further fine-tuning the model) to predict a linguistic property of the input text.

Probing

If we can train a simple classifier to predict a property of the input text based on its representation, it means the property is encoded somewhere in the representation.

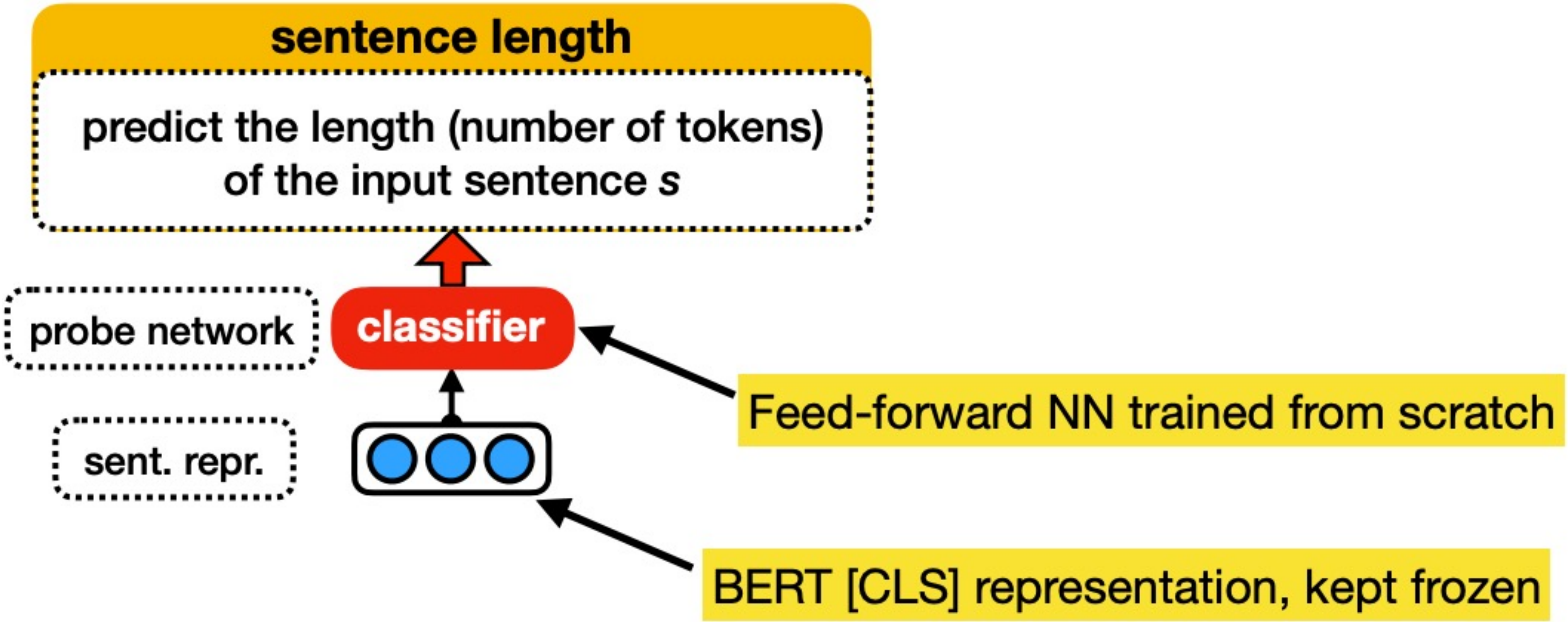
If we cannot, it may or may not be encoded.

Probing



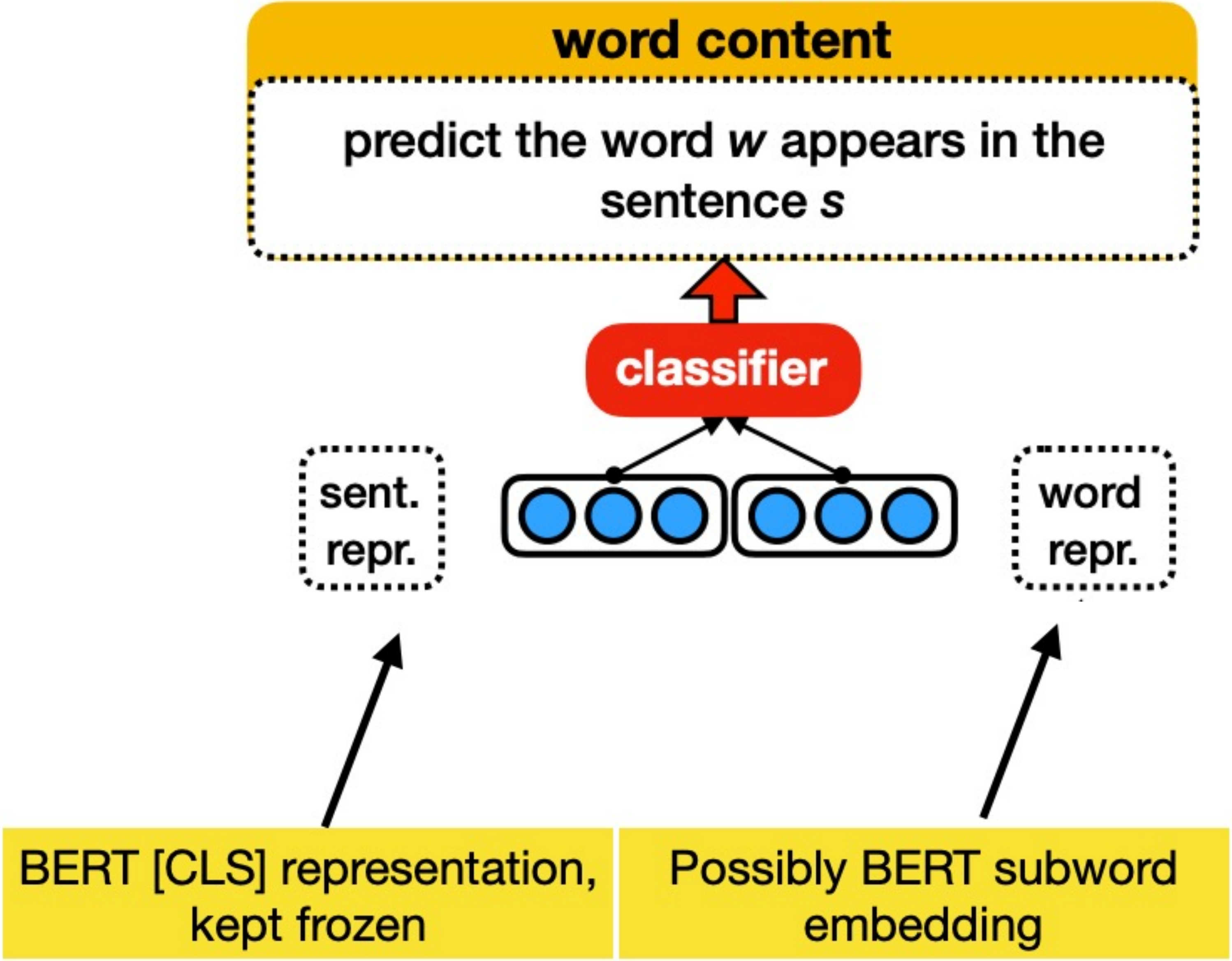
Slide based on, inspired by, or directly from Mohit Iyyer

Probing



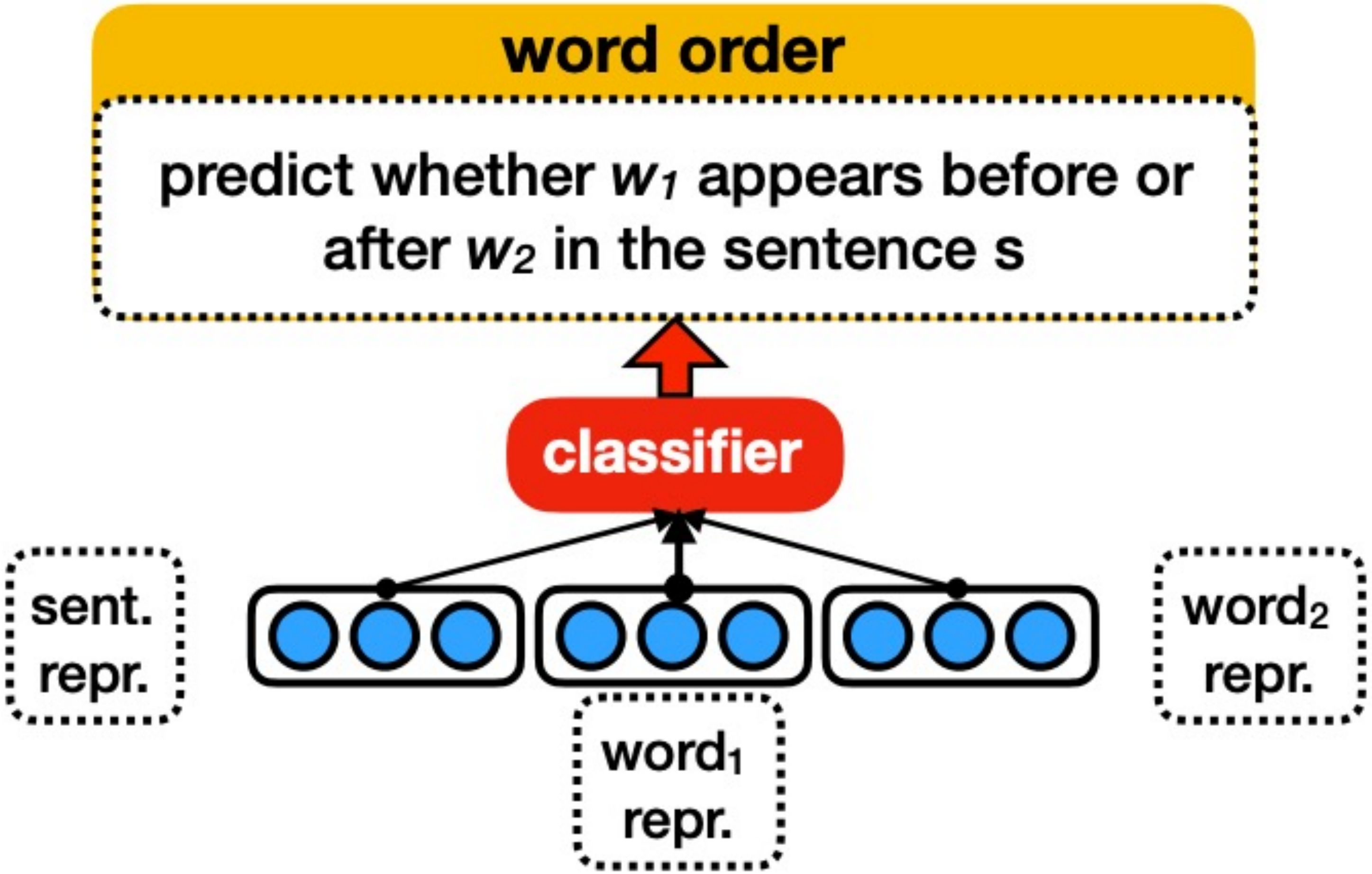
(Adi et al., 2017)

Probing



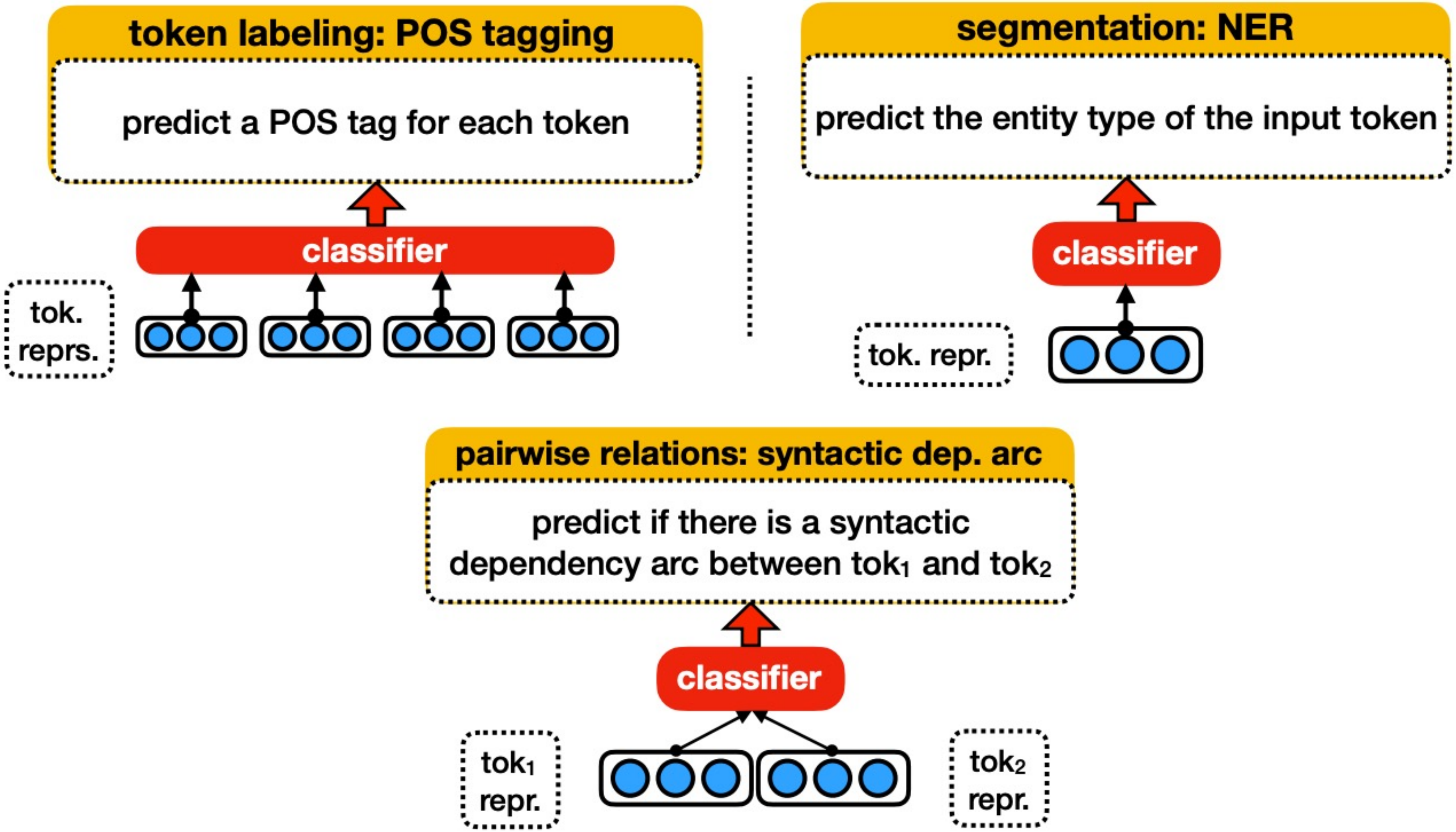
(Adi et al., 2017)

Probing



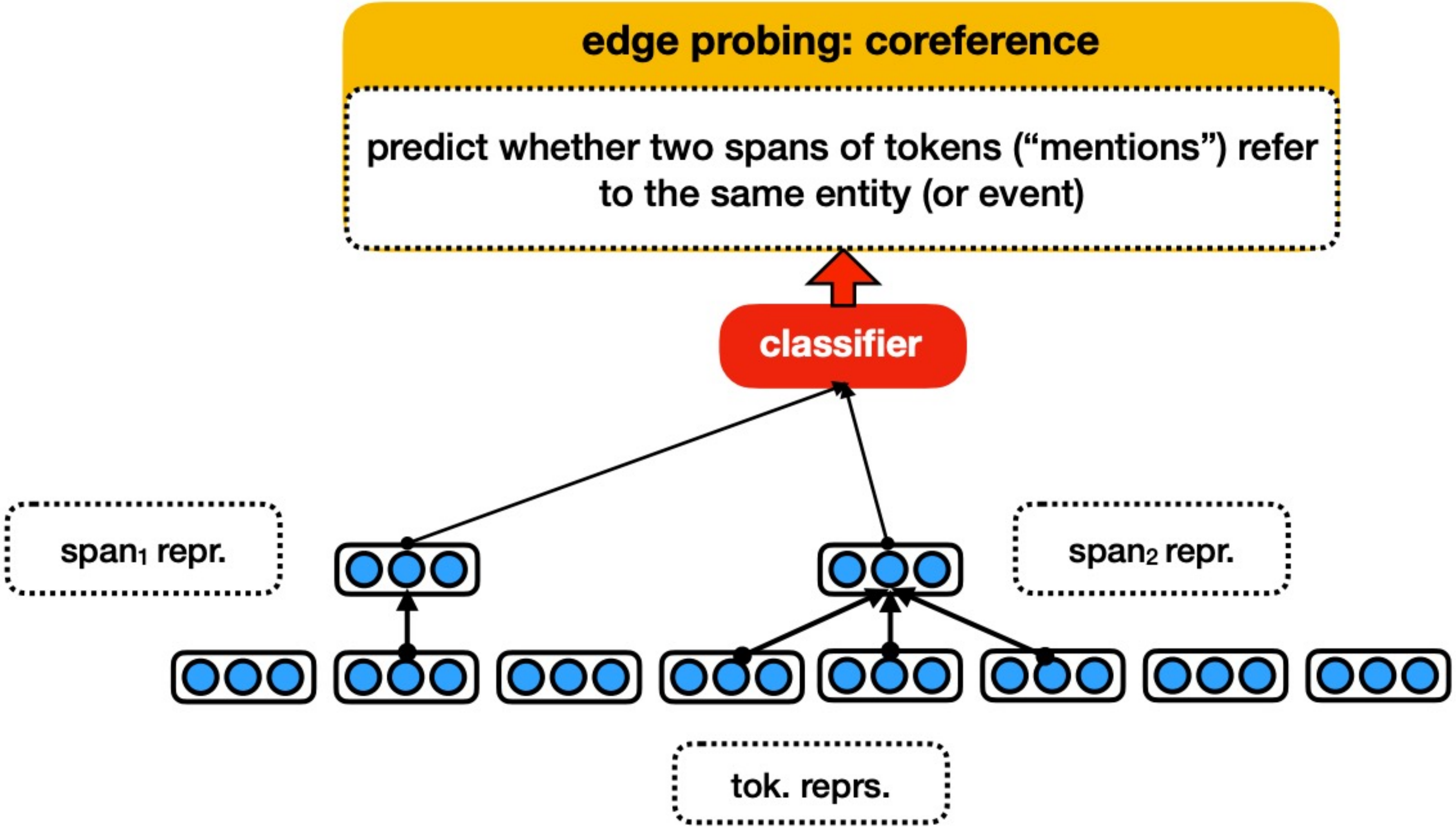
(Adi et al., 2017)

Probing



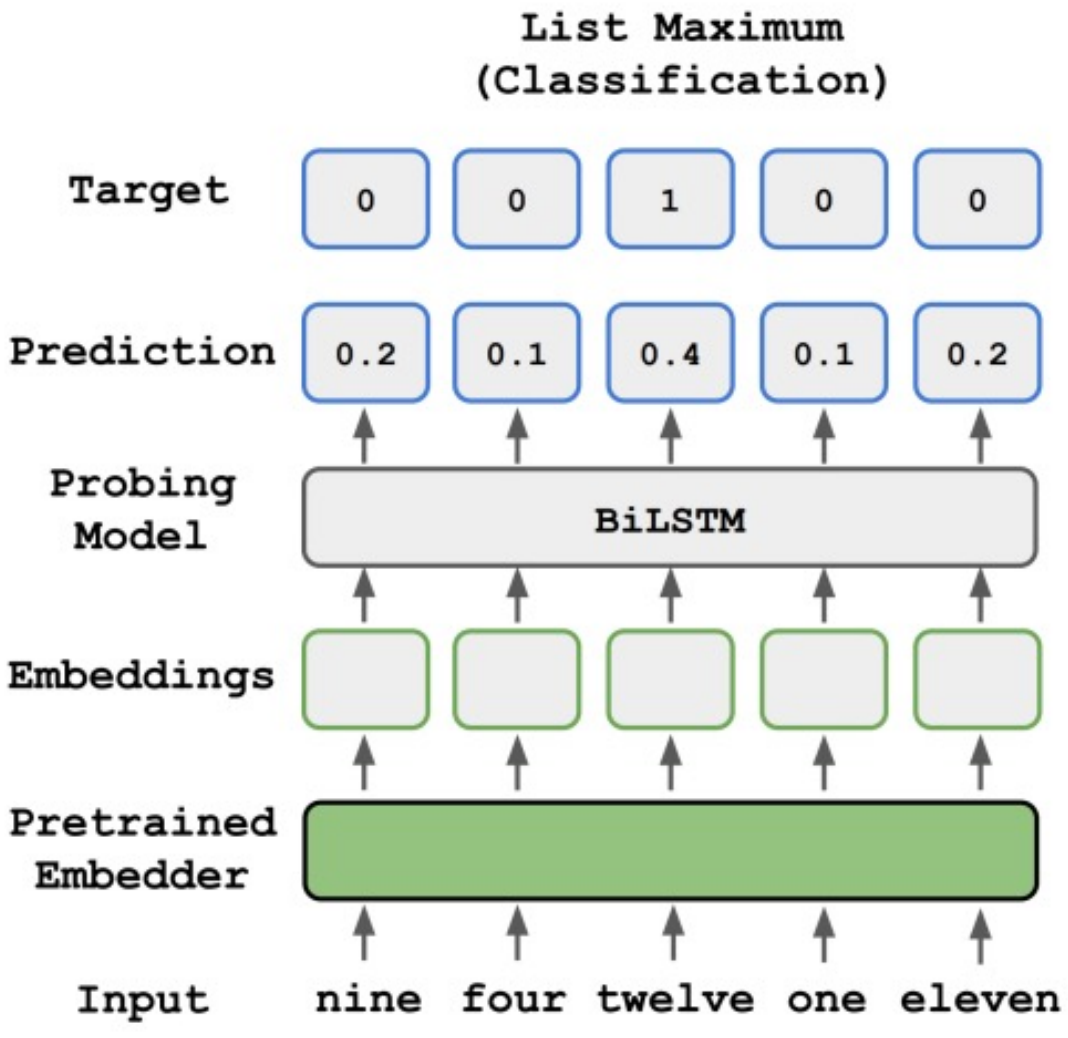
(Liu et al., 2019)

Probing

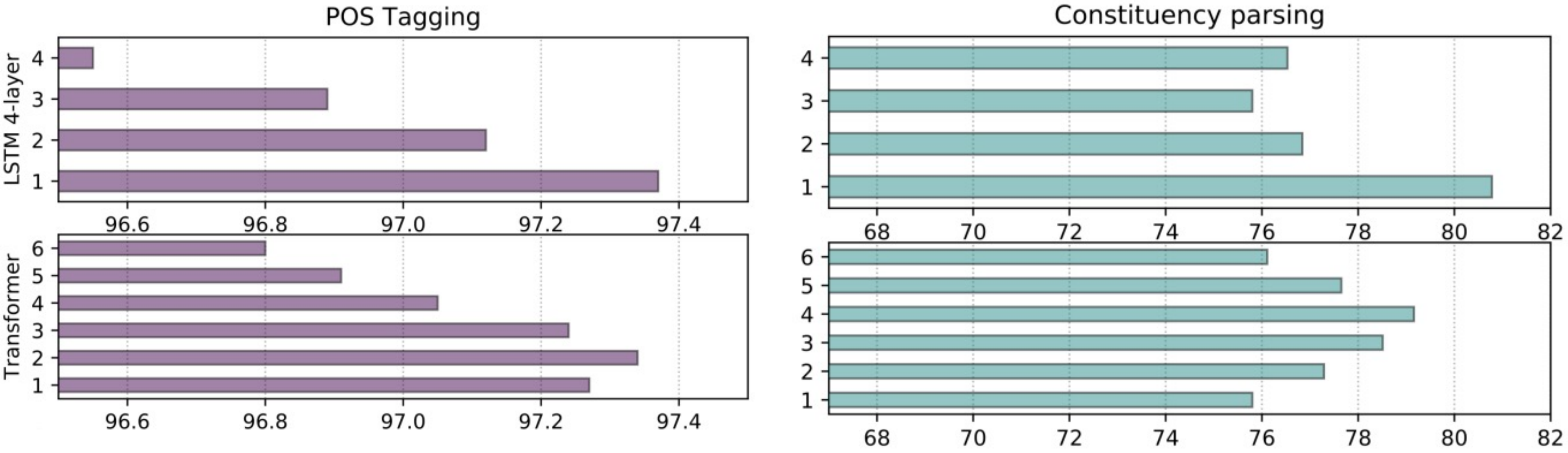


(Tenney et al., 2019)

Probing


List Maximum
(Classification)
Decoding
(Regression)
Addition
(Regression)
Target
0
0
1
0
0
9.0
9.0
Prediction
0.2
0.1
0.4
0.1
0.2
8.5
9.1
Probing
Model
BiLSTM
MLP
MLP
Embeddings
Pretrained
Embedder
Input
nine
four
twelve
one
eleven
nine
seven
two

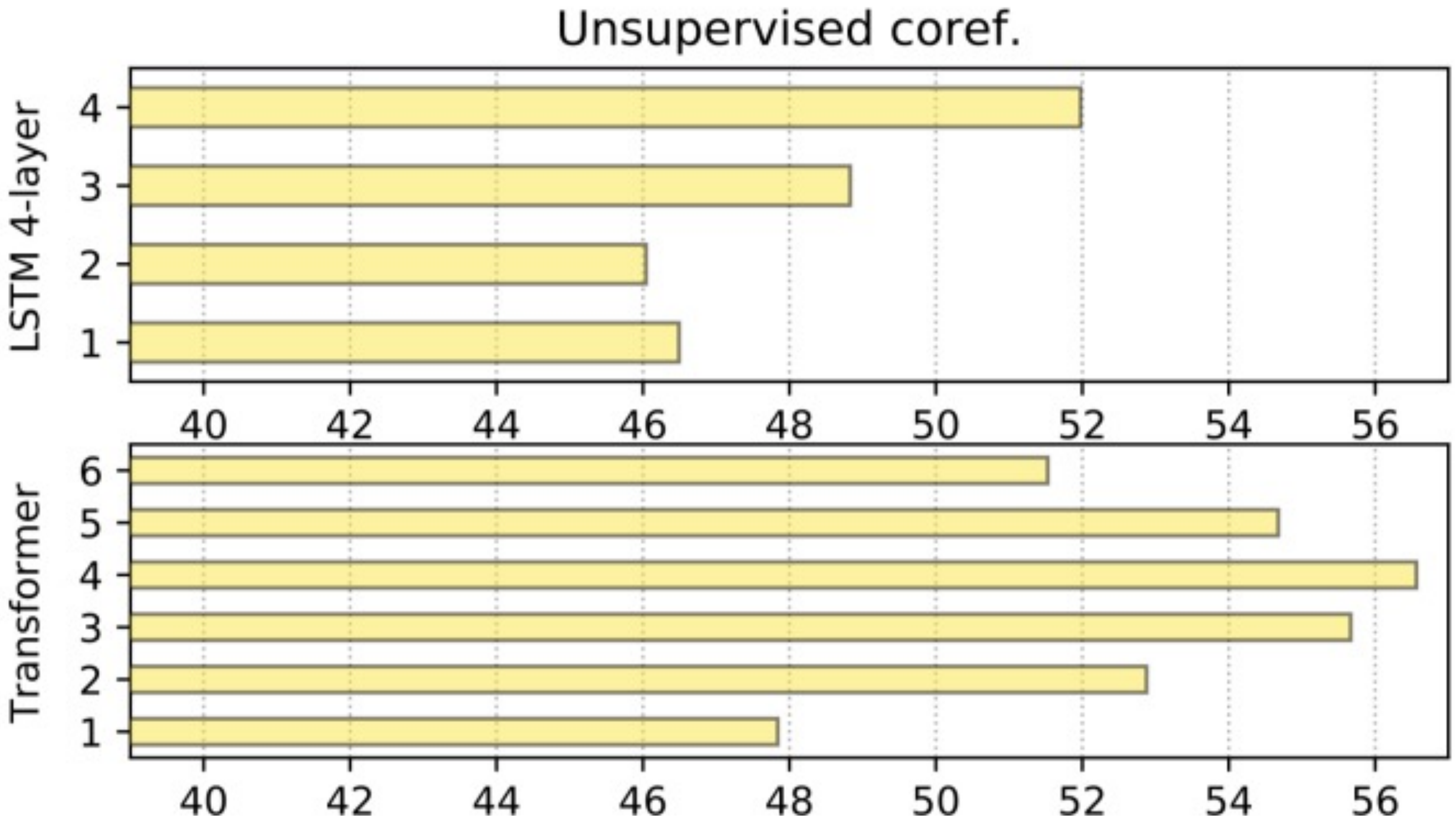
Probing



Lower layers capture more on syntax, upper layers capture more semantics

(Peters et al., 2018)

Probing



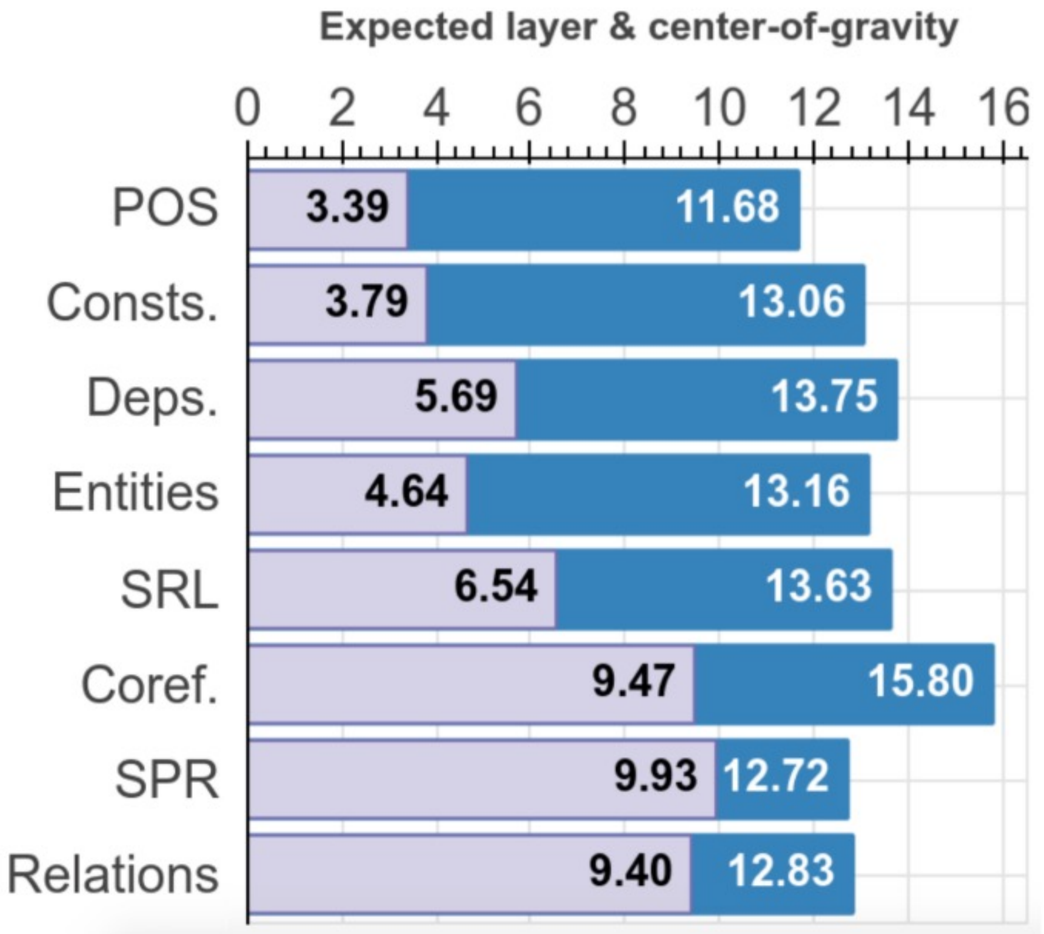
Lower layers capture more on syntax, upper layers capture more semantics

(Peters et al., 2018)

Probing

the expected layer at which the probing model correctly labels an example

a higher center-of-gravity means that the information needed for that task is captured by higher layers



Lower layers capture more on syntax, upper layers capture more semantics

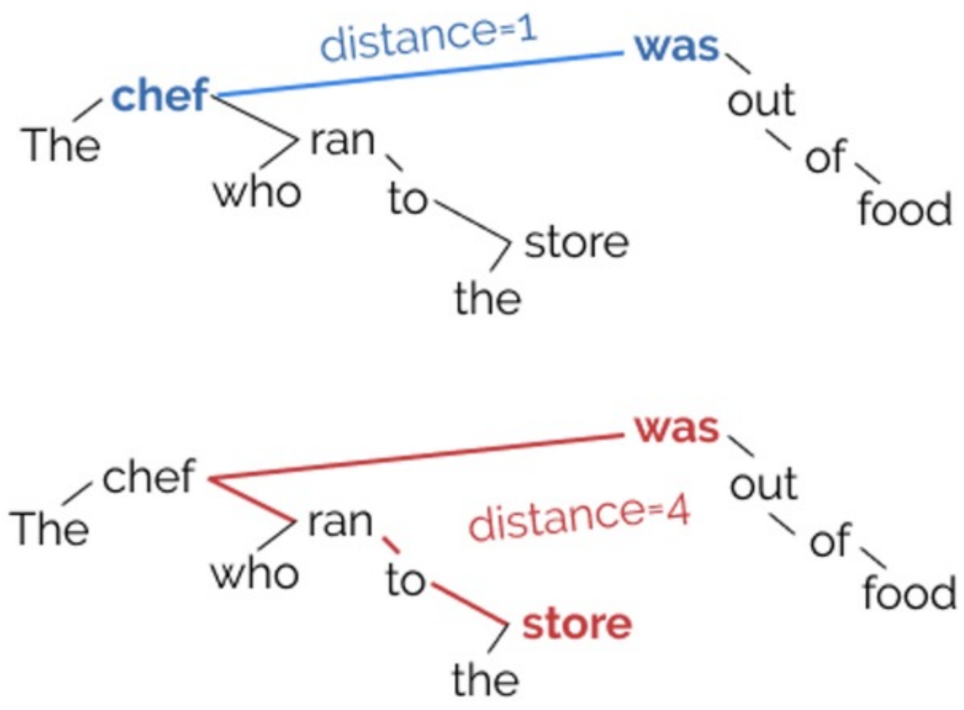
(Tenney et al., 2019)

Probing

The chef who ran to the store was out of food.

1. Because there was no food to be found, the chef went to the next store.

2. After stocking up on ingredients, the chef returned to the restaurant.



Does BERT encode syntactic structure?

(Hewitt and Manning et al., 2019)

Probing

trees as distances and norms

the distance metric—the path length between each pair of words—recovers the tree *T* simply by identifying that nodes *u*, *v* with distance $d_{T\,(u,\,v)} = 1$ are neighbors

the node with greater norm—depth in the tree—is the child

Does BERT encode syntactic structure?

(Hewitt and Manning et al., 2019)

Probing

- probe task 1 — distance:
  predict the path length between each given pair of words

- probe task 2 — depth/norm:
  predict the depth of a given word in the parse tree

Does BERT encode syntactic structure?

(Hewitt and Manning et al., 2019)

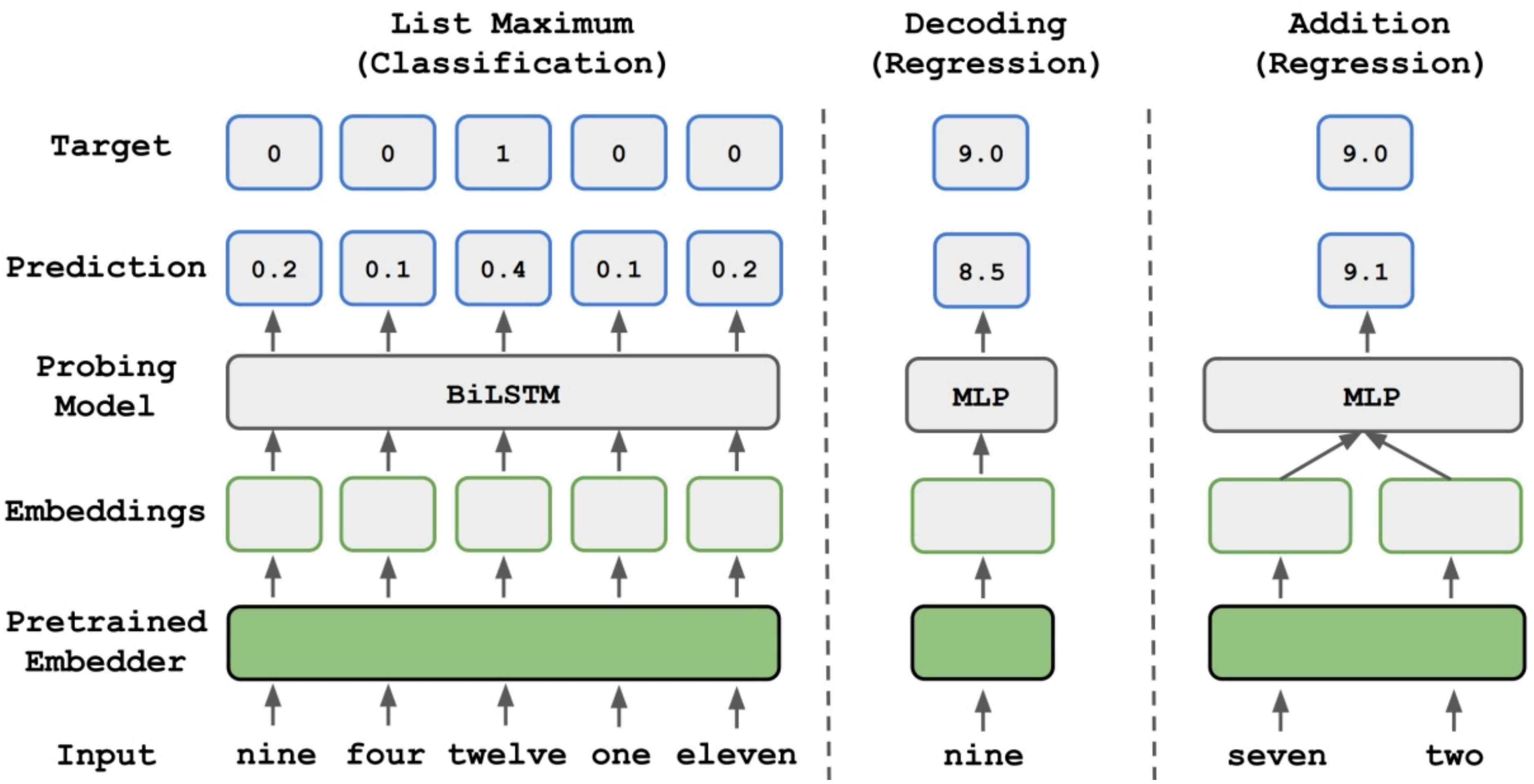
Probing

Method	Distance		Depth	
	UUAS	DSpr.	Root%	NSpr.
ELMo1	77.0	0.83	86.5	0.87
BERTBASE7	79.8	0.85	88.0	0.87
BERTLARGE15	**82.5**	0.86	89.4	0.88
BERTLARGE16	81.7	**0.87**	**90.1**	**0.89**

Does BERT encode syntactic structure?

(Hewitt and Manning et al., 2019)

Probing



Does BERT know numbers?

(Wallace et al., 2019)

Probing

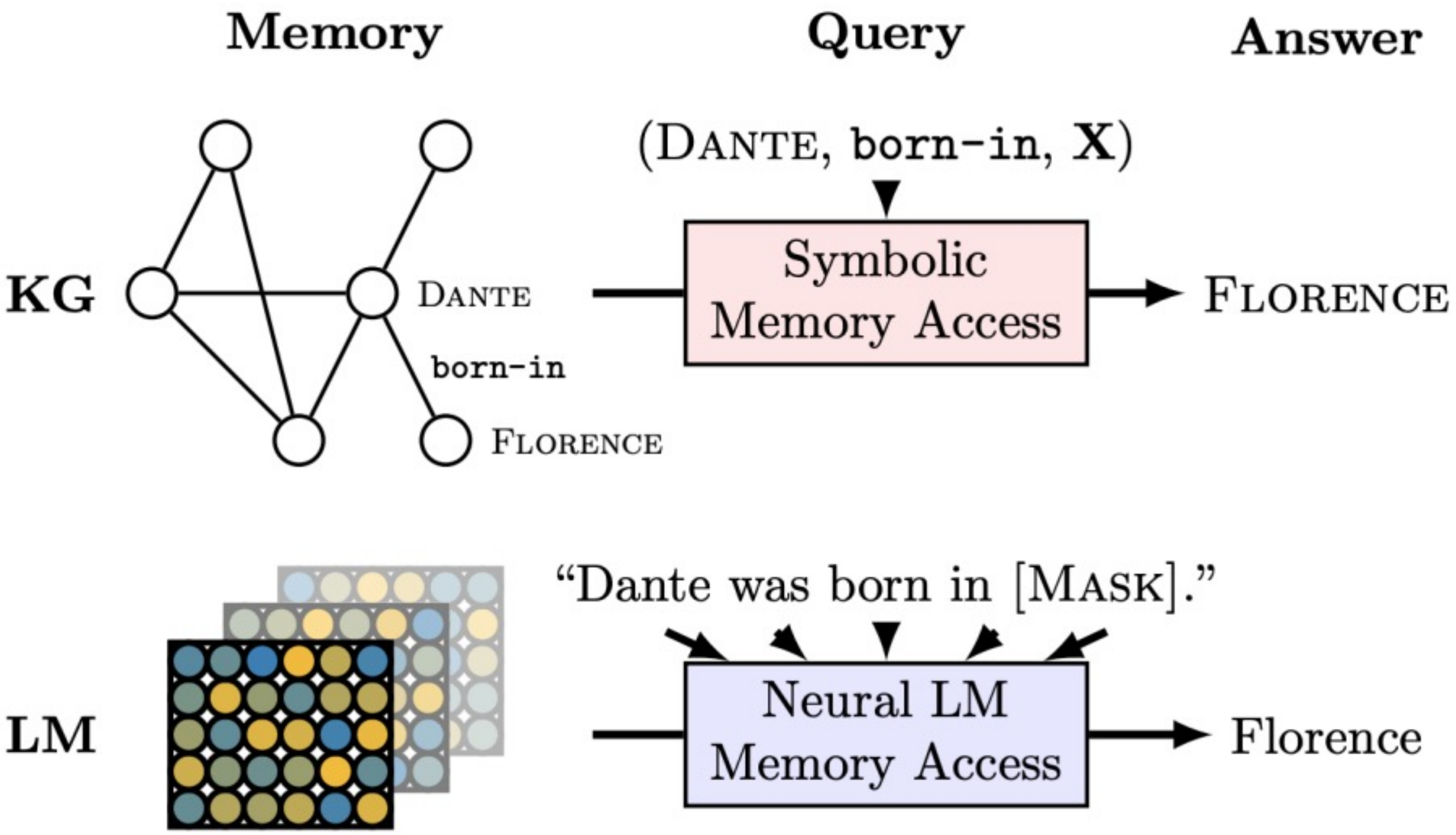
Interpolation	List Maximum (5-classes)			Decoding (RMSE)			Addition (RMSE)		
Integer Range	[0,99]	[0,999]	[0,9999]	[0,99]	[0,999]	[0,9999]	[0,99]	[0,999]	[0,9999]
Random Vectors	0.16	0.23	0.21	29.86	292.88	2882.62	42.03	410.33	4389.39
Untrained CNN	0.97	0.87	0.84	2.64	9.67	44.40	1.41	14.43	69.14
Untrained LSTM	0.70	0.66	0.55	7.61	46.5	210.34	5.11	45.69	510.19
Pre-trained									
Word2Vec	0.90	0.78	0.71	2.34	18.77	333.47	0.75	21.23	210.07
GloVe	0.90	0.78	0.72	2.23	13.77	174.21	0.80	16.51	180.31
ELMo	0.98	0.88	0.76	2.35	13.48	62.20	0.94	15.50	45.71
BERT	0.95	0.62	0.52	3.21	29.00	431.78	4.56	67.81	454.78

(Wallace et al., 2019)

Does BERT know numbers?

Probing

LAMA (LAnguage Model Analysis) probe



Does BERT know world knowledge?

(Petroni et al., 2019)

Probing

LAMA (LAnguage Model Analysis) probe

- manually define templates for considered relations, e.g., "[S] was born in [O]" for "place of birth"
- find sentences that contain both the subject and the object, then mask the object within the sentences and use them as templates for querying
- create cloze-style questions, e.g., rewriting "Who developed the theory of relativity?" as "The theory of relativity was developed by [MASK]"

(Petroni et al., 2019)

Does BERT know world knowledge?

Probing

LAMA (LAnguage Model Analysis) probe

	Relation	Query	Answer	Generation
T-Rex	P54	Dani Alves plays with ____ .	Barcelona	Santos [-2.4] , Porto [-2.5] , Sporting [-3.1] , Brazil [-3.3] , Portugal [-3.7]
	P106	Paul Toungui is a ____ by profession .	politician	lawyer [-1.1] , journalist [-2.4] , teacher [-2.7] , doctor [-3.0] , physician [-3.7]
	P527	Sodium sulfide consists of ____.	sodium	water [-1.2] , sulfur [-1.7] , **sodium [-2.5]** , zinc [-2.8] , salt [-2.9]
	P102	Gordon Scholes is a member of the ____ political party.	Labor	Labour [-1.3] , Conservative [-1.6] , Green [-2.4] , Liberal [-2.9] , **Labor [-2.9]**
	P530	Kenya maintains diplomatic relations with ____.	Uganda	India [-3.0] , **Uganda [-3.2]** , Tanzania [-3.5] , China [-3.6] , Pakistan [-3.6]
	P176	iPod Touch is produced by ____.	Apple	**Apple [-1.6]** , Nokia [-1.7] , Sony [-2.0] , Samsung [-2.6] , Intel [-3.1]
	P30	Bailey Peninsula is located in ____.	Antarctica	**Antarctica [-1.4]** , Bermuda [-2.2] , Newfoundland [-2.5] , Alaska [-2.7] , Canada [-3.1]
	P178	JDK is developed by ____.	Oracle	IBM [-2.0] , Intel [-2.3] , Microsoft [-2.5] , HP [-3.4] , Nokia [-3.5]
	P1412	Carl III used to communicate in ____.	Swedish	German [-1.6] , Latin [-1.9] , French [-2.4] , English [-3.0] , Spanish [-3.0]
	P17	Sunshine Coast, British Columbia is located in ____.	Canada	**Canada [-1.2]** , Alberta [-2.8] , Yukon [-2.9] , Labrador [-3.4] , Victoria [-3.4]
ConceptNet	AtLocation	You are likely to find a overflow in a ____.	drain	sewer [-3.1] , canal [-3.2] , toilet [-3.3] , stream [-3.6] , **drain [-3.6]**
	CapableOf	Ravens can ____.	fly	**fly [-1.5]** , fight [-1.8] , kill [-2.2] , die [-3.2] , hunt [-3.4]
	CausesDesire	Joke would make you want to ____.	laugh	cry [-1.7] , die [-1.7] , **laugh [-2.0]** , vomit [-2.6] , scream [-2.6]
	Causes	Sometimes virus causes ____.	infection	disease [-1.2] , cancer [-2.0] , **infection [-2.6]** , plague [-3.3] , fever [-3.4]
	HasA	Birds have ____.	feathers	wings [-1.8] , nests [-3.1] , **feathers [-3.2]** , died [-3.7] , eggs [-3.9]
	HasPrerequisite	Typing requires ____.	speed	patience [-3.5] , precision [-3.6] , registration [-3.8] , accuracy [-4.0] , **speed [-4.1]**
	HasProperty	Time is ____.	finite	short [-1.7] , passing [-1.8] , precious [-2.9] , irrelevant [-3.2] , gone [-4.0]
	MotivatedByGoal	You would celebrate because you are ____.	alive	happy [-2.4] , human [-3.3] , **alive [-3.3]** , young [-3.6] , free [-3.9]
	ReceivesAction	Skills can be ____.	taught	acquired [-2.5] , useful [-2.5] , learned [-2.8] , combined [-3.9] , varied [-3.9]
	UsedFor	A pond is for ____.	fish	swimming [-1.3] , fishing [-1.4] , bathing [-2.0] , **fish [-2.8]** , recreation [-3.1]

(Petroni et al., 2019)

Does BERT know world knowledge?

Probing

- usually classification problems that focus on simple linguistic properties
- ask simple questions, minimizing interpretability problems
- because of their simplicity, it is easier to control for biases in probing tasks than in downstream tasks
- the probing task methodology is agnostic with respect to the encoder architecture, as long as it produces a vector representation of input text

(Conneau et al., 2018)

Probing

Probing seems great.

Any negatives?

Probing

- Does not necessarily correlate with downstream performance
- Probe may simply learn the task

(Conneau et al., 2018)

Probing

arguments for "simple" probes

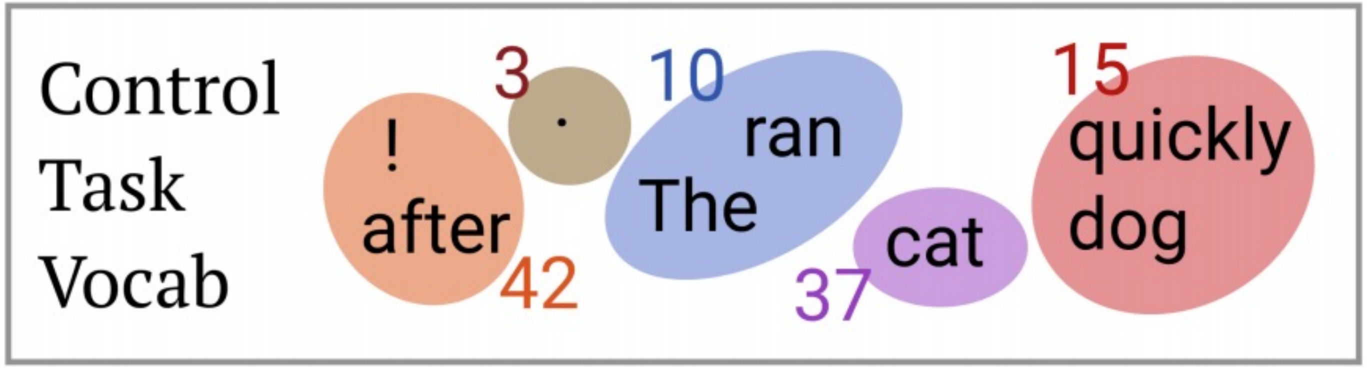
we want to find easily accessible information in a representation

arguments for "complex" probes

useful properties might be encoded non-linearly

(Hewitt et al., 2019)

Probing with Control Tasks



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**

(Hewitt et al., 2019)

Probing with Control Tasks

- independently sample a control behavior *C(v)* for each word type *v* in the vocabulary
- specifies how to define $y_i \in Y$ for a word token x_i with word type *v*
- *control task is a function that maps each token* x_i *to the label specified by the behavior* $C(x_i)$

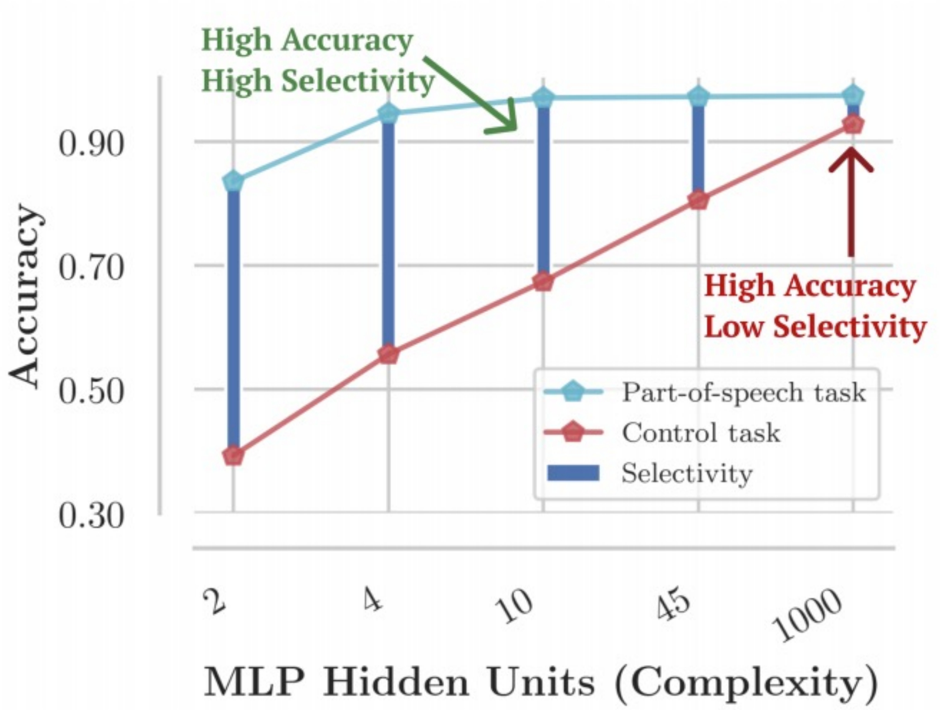
$$f_{\text{control}}(\mathbf{x}_{1:T}) = f(C(x_1), C(x_2), \ldots C(x_T))$$

(Hewitt et al., 2019)

Probing with Control Tasks

selectivity: high linguistic task accuracy + low control task accuracy

measures the probe model's ability to make output decisions independently of linguistic properties of the representation



(Hewitt et al., 2019)

Slide based on, inspired by, or directly from Mohit Iyyer

Probing with Control Tasks

Be careful about probe accuracies!

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

(Hewitt et al., 2019)

Slide based on, inspired by, or directly from Mohit Iyyer

Conclusions

- Just like with all of Machine Learning and Data Science, scrutinize your data and results
- Inspect your model, generally, to ensure it's correctly implemented and sound
- Inspect your predictions to ensure your evaluations are sound
- Inspect your model's internals to understand why it makes its predictions
- Dissect your method above to ensure it's fair and accurate