

# Methods for analyzing (current) NLP models

Ana Marasović

AllenNLP @ Allen Institute for AI & University of Washington

Guest lecture in CSE 517 @ University of Washington

March 2020

**Allen**NLP

**Ai2** Allen Institute for AI



# The rise of contextualized word representations

Application of ML to text requires the **conversion of words into a numerical form**

- **Word embeddings:** dense high-dimensional vectors whose vicinity in a vector space correlates with their association similarity (e.g. coffee–cup, car–wheel)
- **Pretrained word embeddings:** taken “off-the-shelf” using a toolkit (e.g. *word2vec*) that derives them from another ML model that is already unsupervisedly trained on large unlabeled corpora

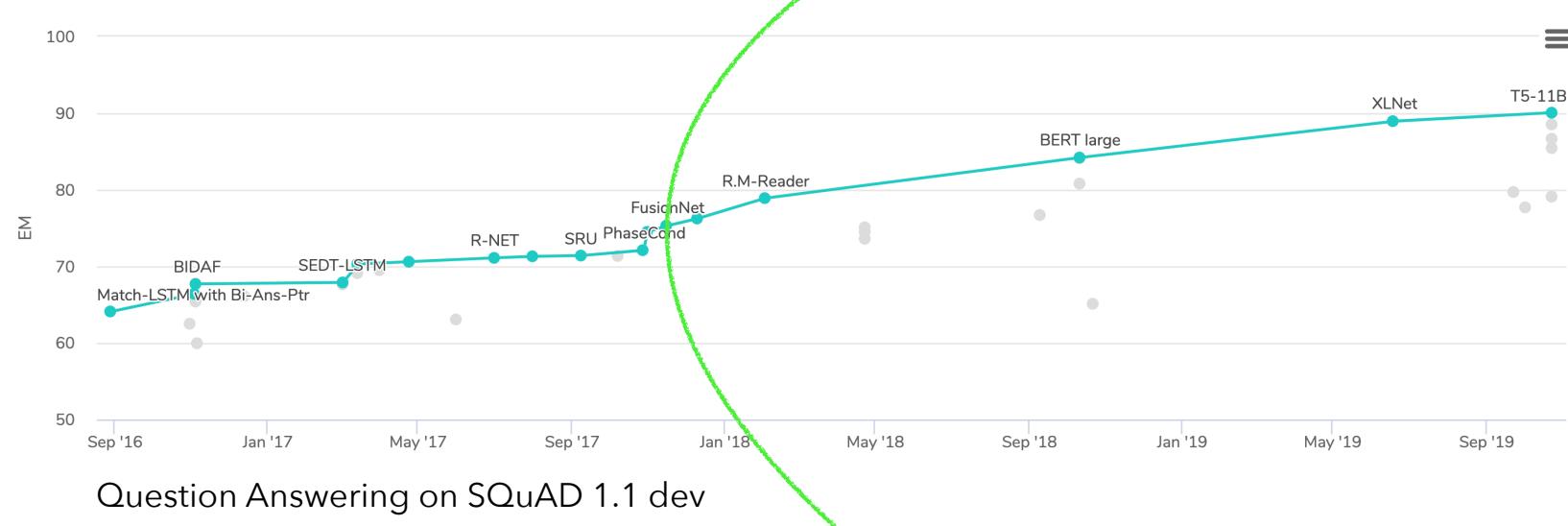
**Static word embeddings:** a single word embedding for a given word, regardless of other words in a given sentence

- A word can contain a wide range of different meanings, depending on the context:
  - An A-to-Z guide on how you can use Google's Bert for binary text classification tasks with Python and Pytorch*
  - In one sketch, Bert reads a book called Boring Stories and chuckles, "Wow! These boring stories are really exciting!"*

# The rise of contextualized word representations

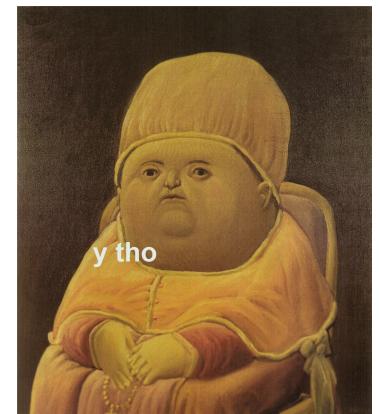
**Bidirectional “masked” language models** represent a given word dependent on the sequence of other words in a given text, i.e. the *context* of the word

The shift to CWRs played a pivotal role in NLP



# New trend in NLP

**examining which linguistic phenomena may or may not  
be captured by pretrained language models**



# 1

## What linguistic knowledge can be taken for granted when we utilize pretrained LMs for downstream tasks?

Are “pretrained language models and massive compute” indeed all we need?

Sutton (2019); Brooks (2019)



Thang Luong  
@lmthang

A new era of NLP has just begun a few days ago: large pretraining models (Transformer 24 layers, 1024 dim, 16 heads) + massive compute is all you need. BERT from @GoogleAI: SOTA results on everything [arxiv.org/abs/1810.04805](https://arxiv.org/abs/1810.04805). Results on SQuAD are just mind-blowing. Fun time ahead!

### SQuAD1.1 Leaderboard

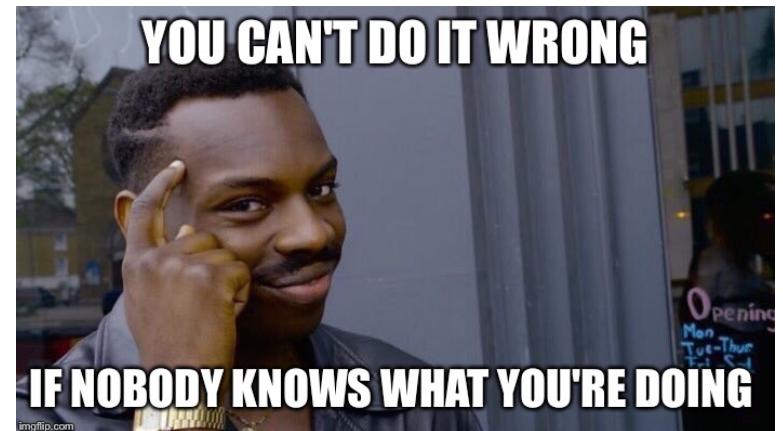
Since the release of SQuAD1.0, the community has made rapid progress, with the best models now rivaling human performance on the task. Here are the ExactMatch (EM) and F1 scores evaluated on the test set of v1.1.

Rank	Model	EM	F1
	Human Performance <i>Stanford University (Rajpurkar et al. '16)</i>	82.304	91.221
1	BERT (ensemble) <i>Google A.I.</i> <small>Oct 05, 2018</small>	87.433	93.160
2	BERT (single model) <i>Google A.I.</i> <small>Oct 05, 2018</small>	85.083	91.835
2	nlNet (ensemble) <i>Microsoft Research Asia</i> <small>Sep 09, 2018</small>	85.356	91.202
2	nlNet (ensemble) <i>Microsoft Research Asia</i> <small>Sep 26, 2018</small>	85.954	91.677
3	QANet (ensemble)	84.454	90.490

2

**What kind of linguistic knowledge available pretrained LMs are not predictive of?**

What are **challenging tasks** and **datasets**?



3

Can we achieve competitive performance with models that derive **word representations that are more transparent about biases they capture?**

Zhao et al (2019); May et al (2019)



Tal Linzen  
@tallinzen

The bitter lesson: if we train them with 100x as many GPUs on 100x as many words, our embeddings won't have any bias!

 Jacob Eisenstein @jacobeisenstein · Mar 15, 2019

Replying to @jacobeisenstein @yoavgo and 2 others

In any case, the conclusion that "our embeddings are fine, they're a reflection of the corpus, let's handle the problem downstream" seems to miss an opportunity to do better, and to ask how far we can really get with frequency-based distributional semantics.

4:37 PM · Mar 15, 2019 · Twitter Web Client

20 Likes



Tal Linzen @tallinzen · Mar 15, 2019

Replying to @tallinzen

Snarkless version: it's a hard problem to figure out what exactly we want statistical learners to do, and it's hard to figure out how put together the training data and task formulation to get them to do that! We need domain knowledge rather than compute for that!

1

2

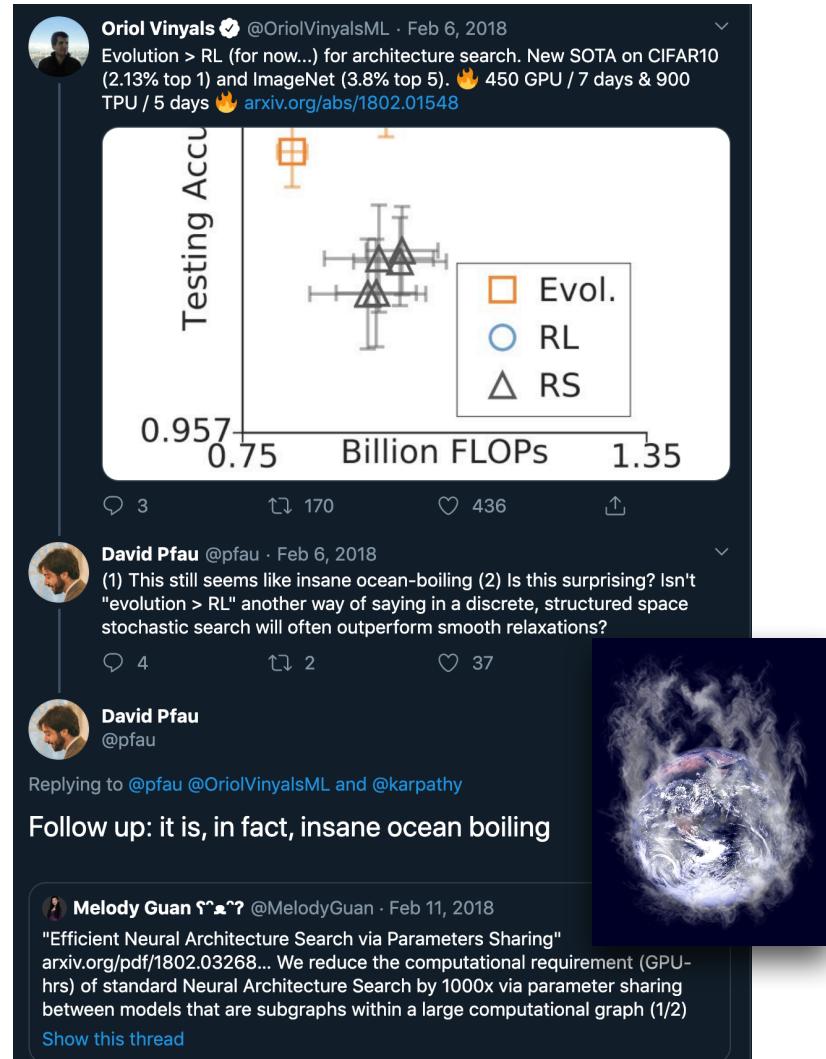
38

↑

4

Can we achieve competitive performance with **smaller, faster, and energy-efficient models?**

Strubell et al (2019); Pham et al (2018); Li and Talwalkar (2018)



# Analysis @ \*ACL 2018–2019

- A. Linguistic probing
  - B. Examining attention weights
  - C. Challenge sets
  - D. Adversarial examples
  - E. Explaining predictions
  - F. ...
- 
- today: an overview of these approaches  
focused on English language***

# Probing



Initial “hypothesis” in the words of John Hewitt:

*“I think my representation learner unsupervisedly developed a notion of linguistic property Y, and encodes this notion in its intermediate representations in order to better perform the task it was trained on (like language modeling).”*

**hard to test directly**

**probing as a proxy**

# Probing workflow

1. **Freeze** an already **pretrained “masked” LM**
2. Specify a **linguistic property** for testing the LM from Step 1
3. Define a **probing task** for assessing “understanding” of this linguistic property
4. Choose a **simple probing model** for solving the probing task
5. Collect **(labelled) data** for this model and make a train/test split
6. Pass the data from the previous step through the LM → CWRs are (some) **LM’s hidden state activations**
7. CWRs from the previous step or their combination are **input features** to the probing model from Step 4
8. Train the probing model → if its test **performance** is:
  - A. “good”, it is likely that **CWRs** are predictive of the linguistic property required for solving the probing task
  - B. **otherwise**, it is hard to draw firm conclusions and **we can not be certain that the linguistic property is not encoded in CWRs**

# Step 1: Which pretrained “masked” LM?

- **BERT** (Coenen et al, 2019; Ettinger 2020; Goldberg 2019; Hewitt and Manning, 2019; Jawahar et al, 2019; Kim et al, 2019; Kober et al, 2019; Lin et al, 2019; Liu et al, 2019; Niven and Kao, 2019; Schwartz and Dagan, 2019; Tenney et al, 2019a, 2019b)
- **ELMo** (Kober et al, 2019; Liu et al, 2019; Peters et al, 2018; Perone et al, 2018; Schwartz and Dagan, 2019; Tenney et al, 2019a)
- **GPT** (Liu et al, 2019; Schwartz and Dagan, 2019; Tenney et al, 2019a)
- **CoVe** (Tenney et al, 2019a; Zhang and Bowman, 2019)

*among others*

## Step 2-3: Which property and task?

**Downstream vs probing tasks:** former are too complex to firmly say what linguistic information captured by CWRs was necessary to solve them

Categories of probing tasks:

- Token labeling
- Segmentation
- Pairwise labeling
- Span labeling
- Structure prediction

# Example: Token labeling

**Property:** basic syntax

$\mathcal{M}_{LM}$  ... frozen LM

$\mathcal{M}_{probe}$  ... trainable probing model

**Task:** POS tagging; label a word in a given sentence with its Part-Of-Speech tag, e.g. noun, verb, adjective, etc.

**Input sentence:** "Vinken, 61 years old"

Vinken, 61 years old →  $\mathcal{M}_{LM}$  → {CWR(Vinken), CWR(61), CWR(years), CWR(old)}

Vinken → CWR(Vinken) →  $\mathcal{M}_{probe}$  → NNP (proper noun, singular)

61 → CWR(61) →  $\mathcal{M}_{probe}$  → CD (cardinal number)

years → CWR (years) →  $\mathcal{M}_{probe}$  → NNS (noun, plural)

old → CWR(old) →  $\mathcal{M}_{probe}$  → JJ (adjective)

# Example: Segmentation

**Property:** spans and boundaries

**Task:** chunking; divide a text in syntactically correlated parts using the BIO notation

**Input sentence:** "He reckons the current account deficit will narrow"

*He reckons the current account deficit will narrow* →  $\mathcal{M}_{LM}$  → {CWR(He), CWR(reckons), ...}

*He* → CWR(He) →  $\mathcal{M}_{probe}$  → B-NP (beginning of a noun phrase)

*reckons* → CWR(reckons) →  $\mathcal{M}_{probe}$  → B-VP (beginning of a verb phrase)

*the* → CWR (the) →  $\mathcal{M}_{probe}$  → B-NP

*current* → CWR(current) →  $\mathcal{M}_{probe}$  → I-NP (inside a noun phrase)

# Example: Pairwise labeling

**Property:** coreferential links between entities in the same sentence

**Task:** intra-sentence coreferential link prediction; given two entities in a given sentence, predict whether they are coreferent

**Input sentence:** "Obama is the former president"

*Obama is the former president* →  $\mathcal{M}_{LM}$  → {CWR(Obama), CWR(president)}

CWR(Obama, president) = concatenate(CWR(Obama); CWR(president); CWR(Obama) ⊕ CWR(president))

CWR(Obama, president) →  $\mathcal{M}_{probe}$  → coreferent

# Example: Span labeling

**Property:** constituent types

**Task:** constituent labeling; label a span of tokens within the phrase-structure parse of the sentence

**Input sentence:** "The important thing about Disney is that it is a global brand."

*The important thing about Disney is that it is a global brand.* →  $\mathcal{M}_{LM}$  → {CWR(The), CWR(important), ...}

CWR(is a global brand) = concat(CWR(is); CWR(brand); CWR(is) ⊕ CWR(brand); CWR(is)-CWR(brand))

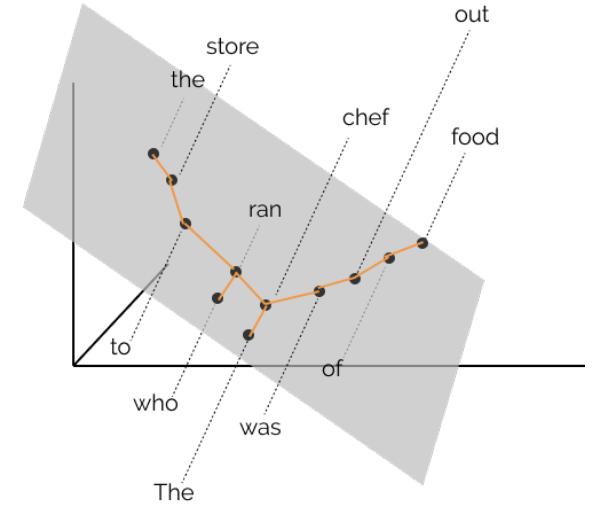
CWR(is a global brand) →  $\mathcal{M}_{probe}$  → VP (verb phrase)

# Example: Structure prediction

**Property:**  $\exists ?$  linear transformation of the CWR-space under which vector distance encodes parse trees

**Task:** reconstructing parse trees

1. Learn a linear transformation of the CWR-space s.t. the squared distance between transformed word vectors correlates with the path length in the dependency parse tree between them
2. Compute the distance between each word pair using their transformed vector representations
3. From the predicted parse tree distances compute the minimum spanning tree



Hewitt and Manning (2019)

## Step 4: Which probing model?

Probing models have to **simple enough** so that we can attribute their performance to CWRs and not to the capacity of the model

**Popular probing models:** linear model (e.g. Liu et al, 2019), one-layer FFNN (e.g. Perone et al, 2018), two-layer FFNN (e.g. Tenney et al, 2019a)

**Linear models** are simple enough, but a non-linear combination of dimensions in a CWR might be predictive of a given linguistic property

Even a **single-layer FFNN** can be potentially have large capacity (hint: universal approximation theorem)

## Step 5: Which probing data?

**Use available annotations:** Penn Treebank (PTB), Universal Dependencies English Web Treebank (UD-EWT), CCGbank, data of CoNLL shared tasks, STREUSLE, UDS,...

⇒ More studies devoted to **syntactic** than **semantic** phenomena due to available data

**Create data:** Diverse Natural Language Inference Collection (DNC; Poliak et al, 2018; Kim et al, 2019)

**Low vs high-resource:** rarely discussed; related to the discussion on the previous slide

## Step 6: Which activations?

**Hidden activations at each layer** (e.g. Peters et al, 2018, Liu et al, 2019)

**Scalar mix of hidden activations at each layer** (e.g. Liu et al, 2019, Shwartz and Dagan, 2019)

**Hidden activations of the final-layer** before the classification layer (e.g. Peters et al, 2018, Shwartz and Dagan, 2019)

# Step 7: What is a “good” performance?

## Current approach

- Compare to the state of the art when possible (e.g. for PoS tagging)
- Otherwise, researchers’ intuition

Do we need a more systematic approach?

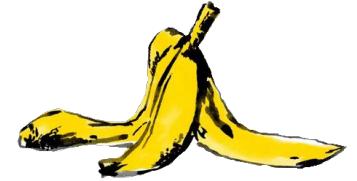
- 90+ → predictive; 80–90 → possibly predictive; <80 → unknown



Fair comparison of different LMs (without re-training) is hard

Comparison with random word representations (Zhang and Bowman, 2018; Conneau et al, 2018)

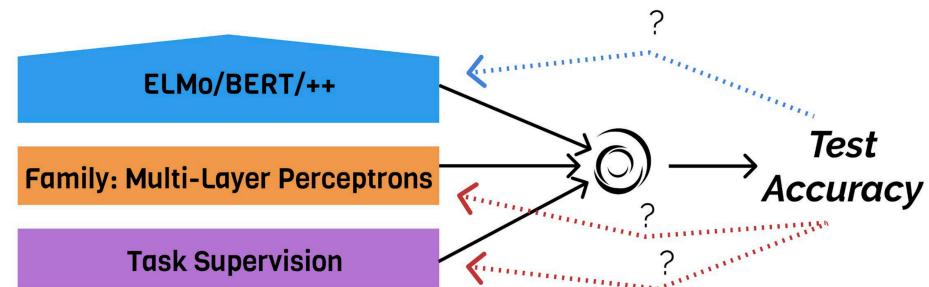
# Probing pitfalls



- What is the definition of encoding/capturing/being predictive of a linguistic property? When is the probing performance satisfying? What can we say about CWRs if the probing classifier does not perform “well”?
- How to firmly attribute probing performance to CWRs and not to the probing model’s capacity? What is the influence of probing train and test data size? (This point is circled in green)
- Predictability of a property does not entail that the end-task model is using it. How to know when certain linguistic knowledge is utilized?
- Should we repeat these analyses every time a new LM is released?

# Toward better probes

- Control tasks (Hewitt and Liang, 2019)



**Control tasks:** tasks that can't have been learned a priori by a representation, but can be learned by the probe through memorization

- **structure:** the output for a word token is a deterministic function of the word type  
→ the probe itself can learn the task
- **randomness:** the output for each word type is sampled independently at random  
→ no representation can have learned the task a priori

Figure from [https://nlp.stanford.edu/~johnhew/2019control\\_slides.pdf](https://nlp.stanford.edu/~johnhew/2019control_slides.pdf)

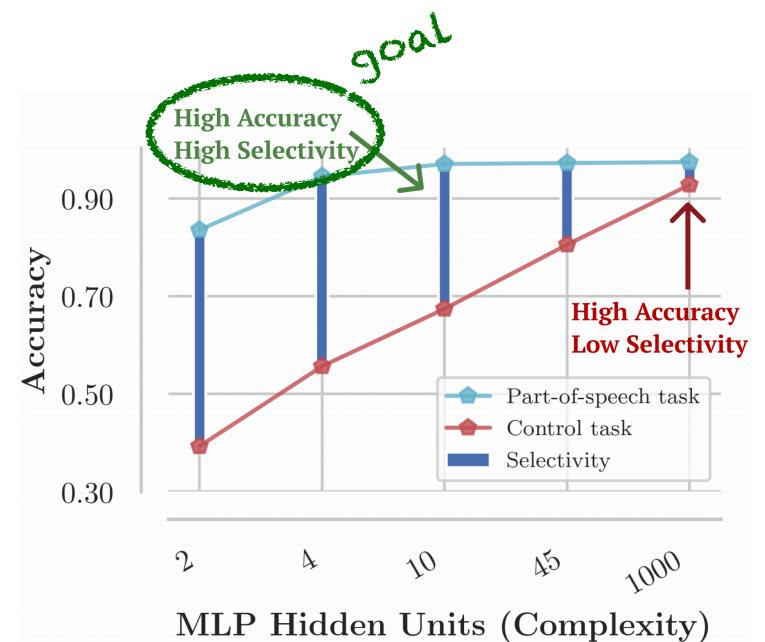
# Toward better probes

- Constructing a control task for POS tagging (Hewitt and Liang, 2019)

$V$  ... vocabulary

$Y$  ... POS tagset

1. **Define control behavior**  $C(\cdot) : V \rightarrow Y$ , that *randomly* partitions vocabulary  $V$  into  $|Y|$  categories, hence *deterministically* labels sentences by looking up a category for each word
2. **Train POS control model**  $f_{control}(\cdot) : X \rightarrow C(V(X))$  using the same architecture as for probing
3. **Compute selectivity:** probe's accuracy on the probing task minus its accuracy on the control task



# What is next for NLP according to probing



Rogers et al, 2019; BERTology primer

1. What linguistic knowledge is guaranteed with pretrained LMs for end-tasks?
2. What kind of linguistic knowledge available pretrained LMs are not predictive of?  
What are challenging tasks and datasets?
3. What biases pretrained LMs capture? Can they be more transparent about them?
- ✗ 4. Can we make pretrained LMs smaller, faster, and more energy-efficient?

# Examining attention



Initial “hypothesis” in the words of John Hewitt:

*“I think my representation learner unsupervisedly developed a notion of linguistic property Y, and encodes this notion in its intermediate representations in order to better perform the task it was trained on (like language modeling).”*

**hard to test directly**

**visualize self-attention  
matrices as a proxy**

**Kovaleva et al. Revealing the Dark Secrets of BERT. EMNLP 2019.**

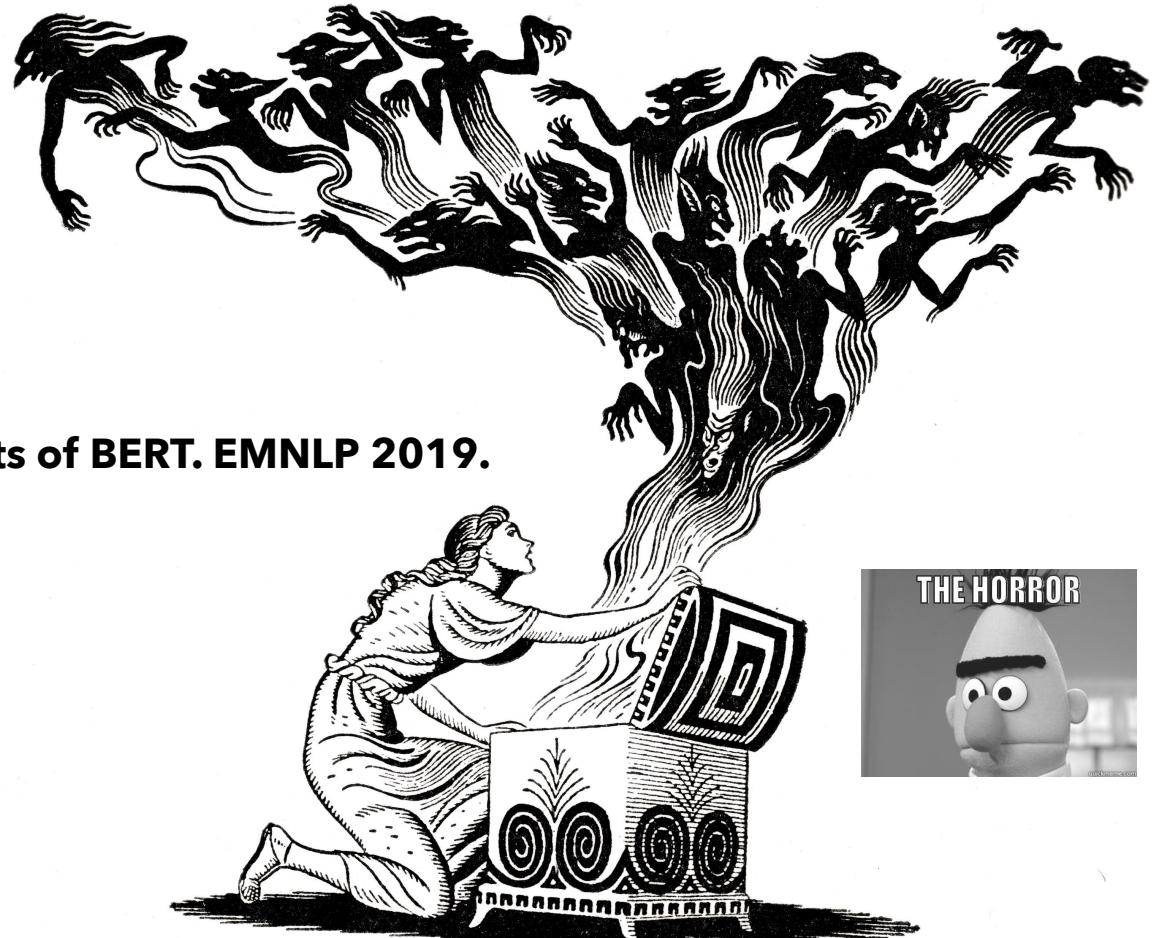


Figure from <https://medium.com/@eruanna317/pandoras-box-2017-278cb0373cb8>

# BERT-Base forward pass

trainable parameters

– Input

$$W_T \in \mathbb{R}^{\text{vocab size} \times d} = \mathbb{R}^{\text{vocab size} \times 768} \dots \text{token embeddings}$$

$$W_P \in \mathbb{R}^{\text{max input length} \times d} = \mathbb{R}^{512 \times 768} \dots \text{positional embeddings}$$

$$I = (i_1, \dots, i_{512}) \in \mathbb{N}_0^{1 \times \text{max input length}} = \mathbb{N}_0^{1 \times 512} \dots \text{input vocab indices}$$

$$T = \text{lookup}(W_T, I) \in \mathbb{R}^{\text{max input length} \times d} = \mathbb{R}^{512 \times 768} \dots \text{input token embeddings}$$

$$X = T + W_P \in \mathbb{R}^{\text{max input length} \times d} = \mathbb{R}^{512 \times 768} \dots \text{input embeddings}$$

$$Z_0 = X$$

# BERT-Base forward pass

– Self-attention layer

trainable parameters  
output of the previous layer

$$Q_{h,l} = Z_{l-1} W_{h,l}^Q \in \mathbb{R}^{\text{max input len} \times d_q} = \mathbb{R}^{512 \times 64} \dots \text{query matrix}$$

$$K_{h,l} = Z_{l-1} W_{h,l}^K \in \mathbb{R}^{\text{max input len} \times d_k} = \mathbb{R}^{512 \times 64} \dots \text{key matrix}$$

$$V_{h,l} = Z_{l-1} W_{h,l}^V \in \mathbb{R}^{\text{max input len} \times d_v} = \mathbb{R}^{512 \times 64} \dots \text{value matrix}$$

$$A_{h,l} = \text{Softmax}\left(\frac{Q_{h,l} K_{h,l}^T}{\sqrt{d_k}}\right) \in \mathbb{R}^{\text{max input len} \times \text{max input len}} = \mathbb{R}^{512 \times 512}$$

$(A_{h,l})_{i,j}$  ... importance of the j-th word for the i-th word

$$Z_{h,l} = A_{h,l} V_{h,l} \in \mathbb{R}^{\text{max input len} \times d_v} = \mathbb{R}^{512 \times 64}$$

$h \in \{1, \dots, n_{\text{heads}}\}, n_{\text{heads}} = 12$

$l \in \{1, \dots, n_{\text{layers}}\}, n_{\text{layers}} = 12$

# BERT-Base forward pass

– Layer normalization

trainable parameters  
output of the previous layer

$$\tilde{Z}_l = \text{concat}(Z_{1,l}, \dots, Z_{n_{\text{heads}},l}) \in \mathbb{R}^{\max \text{ input len} \times (d_v \cdot n \text{heads})} = \mathbb{R}^{512 \times (64 \cdot 12)} = \mathbb{R}^{512 \times 768}$$

$$\bar{Z}_l = \text{LayerNorm}(\textcolor{red}{Z}_{l-1} + \tilde{Z}_l) \in \mathbb{R}^{512 \times 768}$$

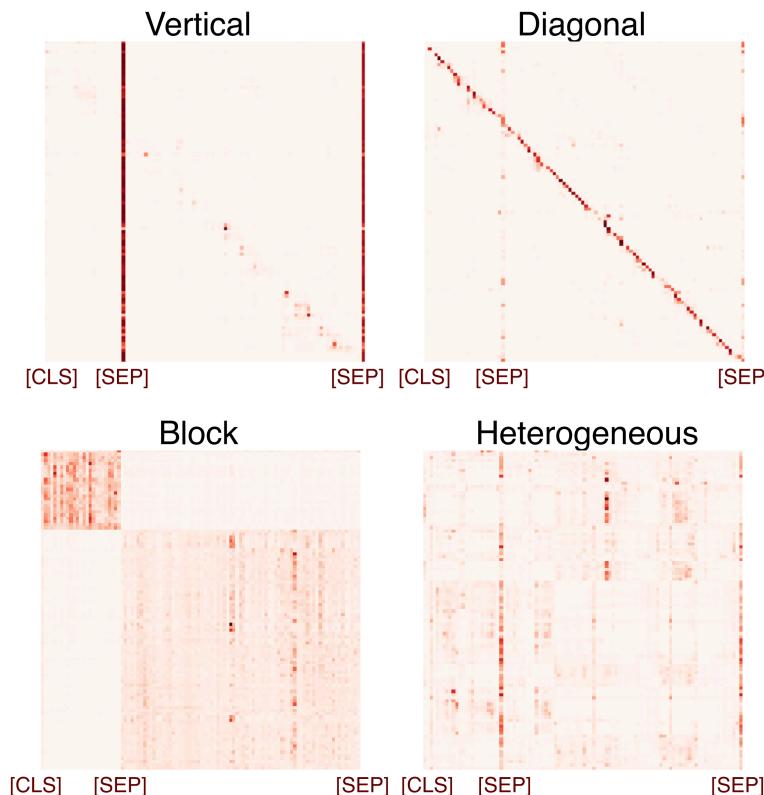
$$Z_l^{ffnn} = \max(0, \bar{Z}_l \textcolor{blue}{W}_l^{ffnn} + \textcolor{blue}{b}_l^{ffnn}) \in \mathbb{R}^{\max \text{ input len} \times d_{ffnn}} = \mathbb{R}^{512 \times 3072}$$

$$Z_l^{out} = Z_l^{ffnn} \textcolor{blue}{W}_l^{out} + \textcolor{blue}{b}_l^{out} \in \mathbb{R}^{\max \text{ input len} \times d} = \mathbb{R}^{512 \times 768}$$

$$Z_l = \text{LayerNorm}(\bar{Z}_l + Z_l^{out}) \in \mathbb{R}^{512 \times 768}$$

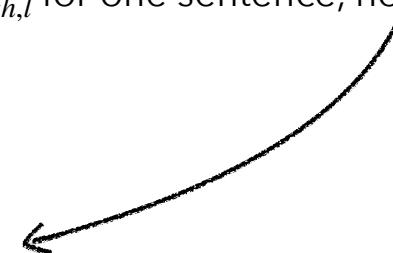
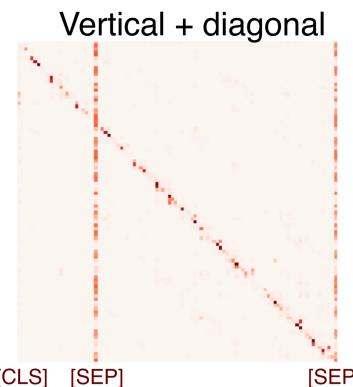
$$l \in \{1, \dots, n_{\text{layers}}\}, n_{\text{layers}} = 12$$

# Self-attention patterns

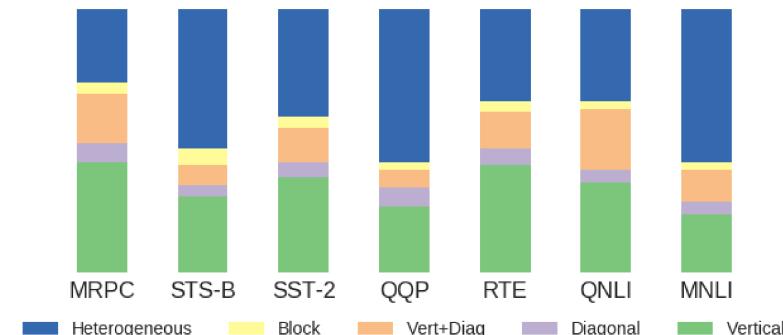


$$A_{h,l} = \text{Softmax}\left(\frac{Q_{h,l}K_{h,l}^T}{\sqrt{d_k}}\right) \in \mathbb{R}^{\text{max input len} \times \text{max input len}} = \mathbb{R}^{512 \times 512}$$

$A_{h,l}$  for one sentence, not average;  $h, l$  unknown



% estimated from a ConvNet that is trained on  
400 manually annotated attention heatmaps



# “Interpretable” attention heads

- Relation-specific heads

## HEAD IMPORTANCE

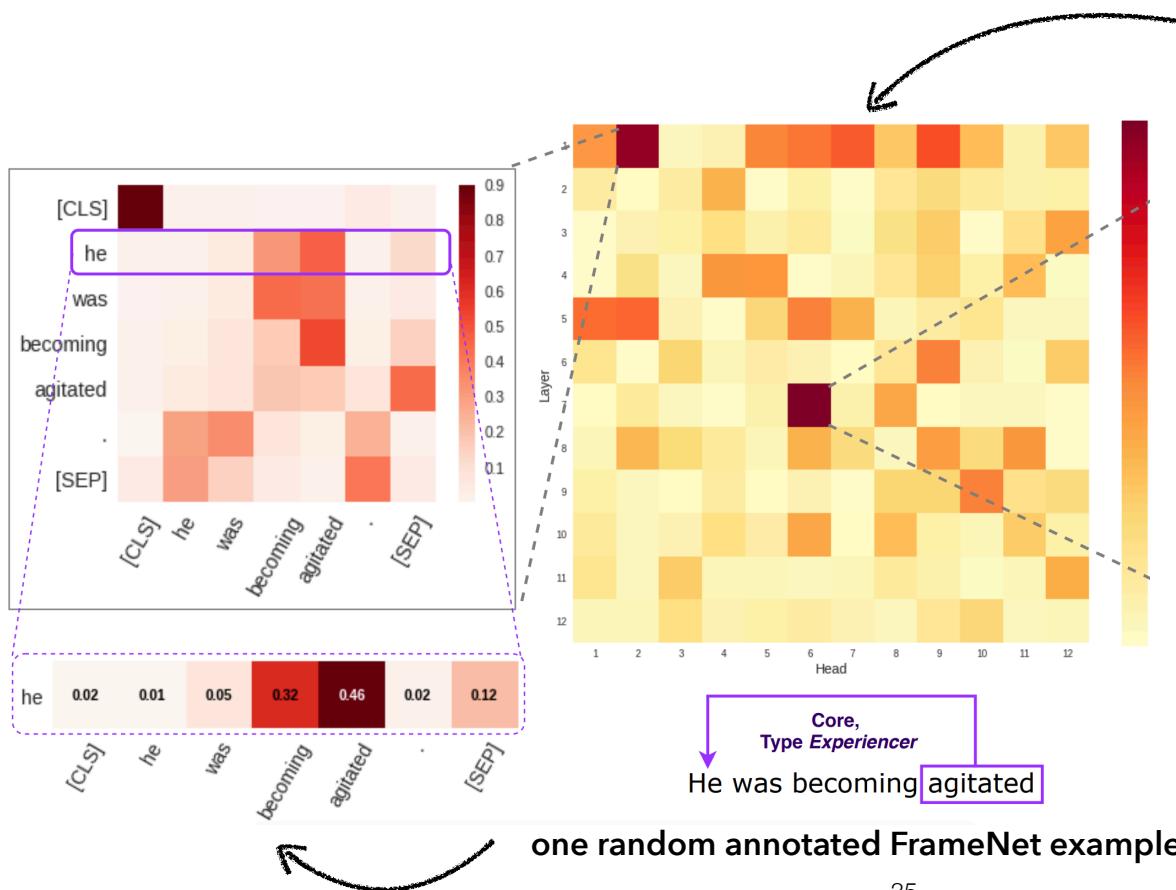
average max absolute attention weight  
among FrameNet-annotated tokens

**Voita et al, 2019:**

Layerwise Relevance Propagation (LRP)

**Michel et al, 2019:**

$$I_h = \mathbb{E}_{x \sim X} |A_{h,l}(x)^T \frac{\partial \mathcal{L}(x)}{\partial A_{h,l}(x)}|$$

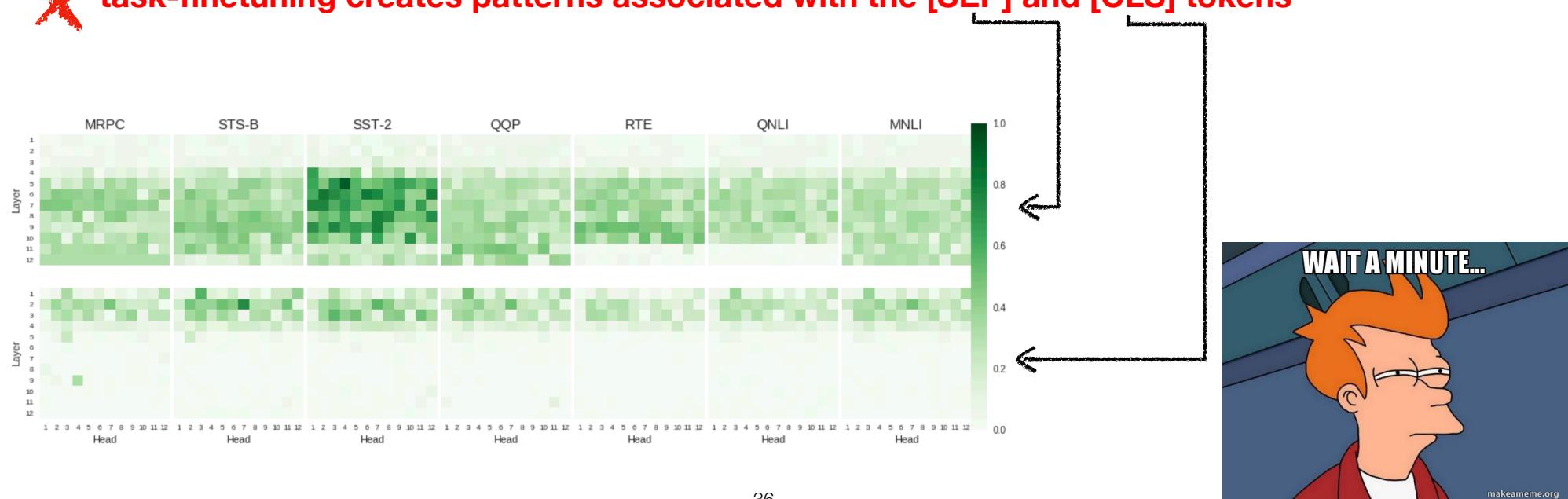


# “Interpretable” attention heads

- Attention to linguistic features

**Hypothesis:** task-finetuning of BERT creates patterns reflecting linguistic features, i.e. specific tokens get higher attention weights, producing **vertical stripes** on the corresponding attention maps

 **task-finetuning creates patterns associated with the [SEP] and [CLS] tokens**



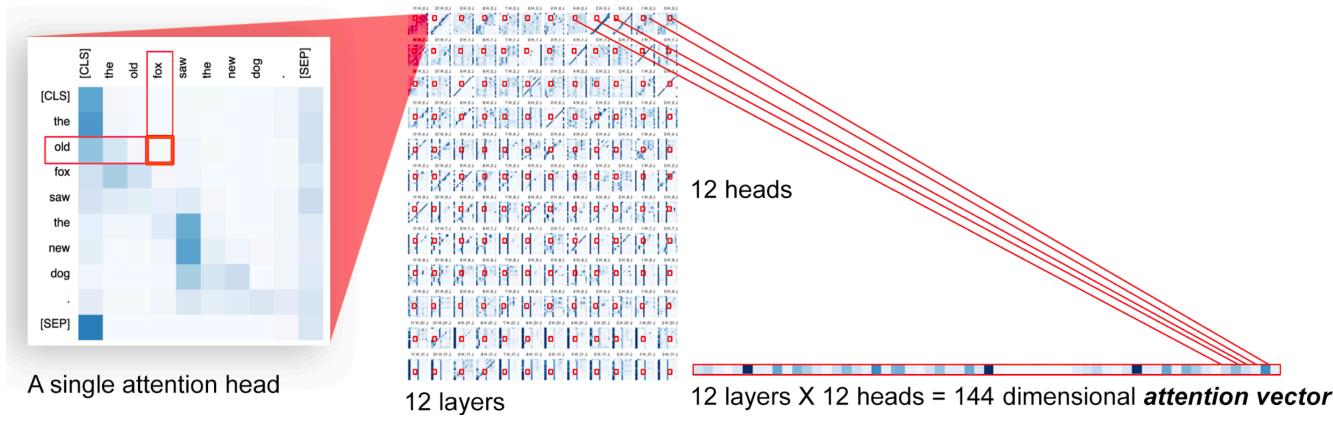
# Attention probes

*Obama is the former president* →  $\mathcal{M}_{LM}$  → {CWR(Obama), CWR(president)}

CWR(Obama, president) = concatenate(CWR(Obama); CWR(president); CWR(Obama) ⊙ CWR(president))

CWR(Obama, president) →  $\mathcal{M}_{probe}$  → coreferent

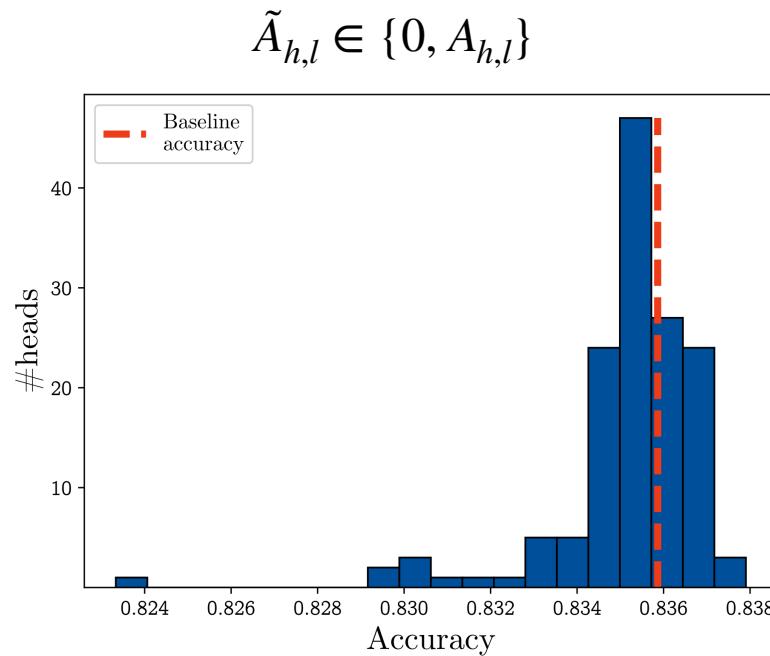
Instead of CWR(Obama, president), combine attention scores  $(A_{h,l})_{i_{Obama}, j_{president}}$



Reif et al, 2019

# Pruning attention heads

– Michel et al, 2019; BERT for NLI



🤗 BERTology

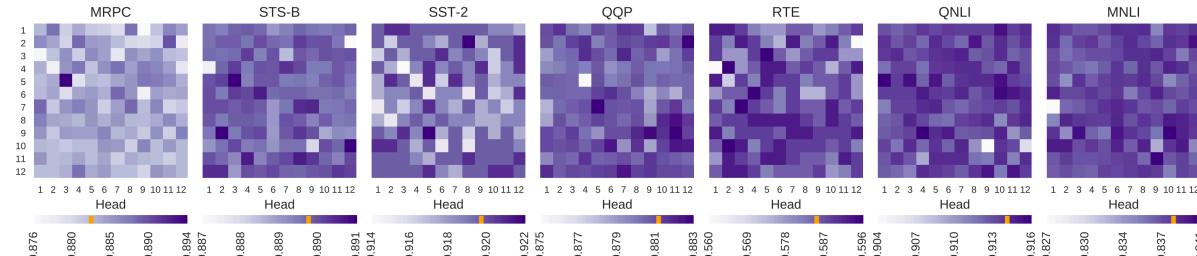
<https://huggingface.co/transformers/bertology.html>

# Pruning attention heads

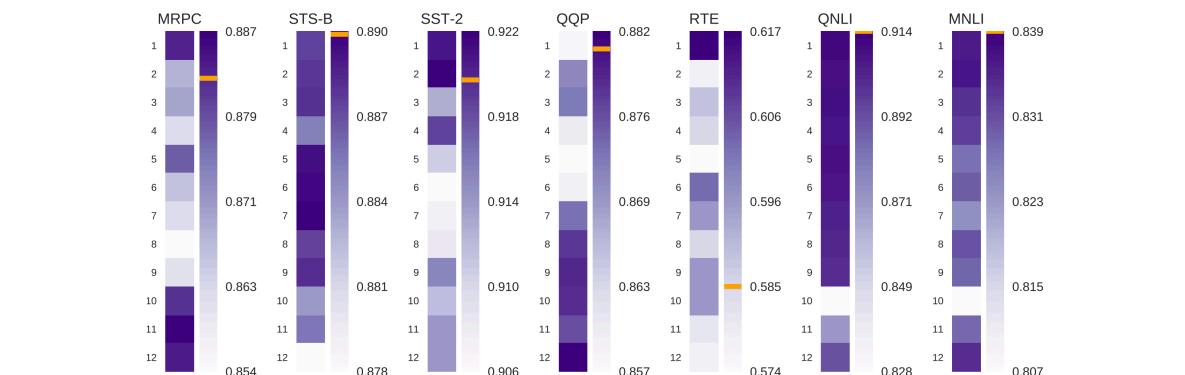
– Kovaleva et al, 2019; BERT for GLUE

$$A_{h,l} = \frac{1}{\text{sentece length}}$$

performance while disabling  
one head at a time



performance while disabling  
one layer (12 heads) at a time



# Pruning attention heads

– Voita et al, 2019; Transformer for NMT

$$\tilde{A}_{h,l} = g_{h,l} \cdot A_{h,l}$$

$g_{h,l} \in \mathbb{R}$  ... a scalar gate independent of the input

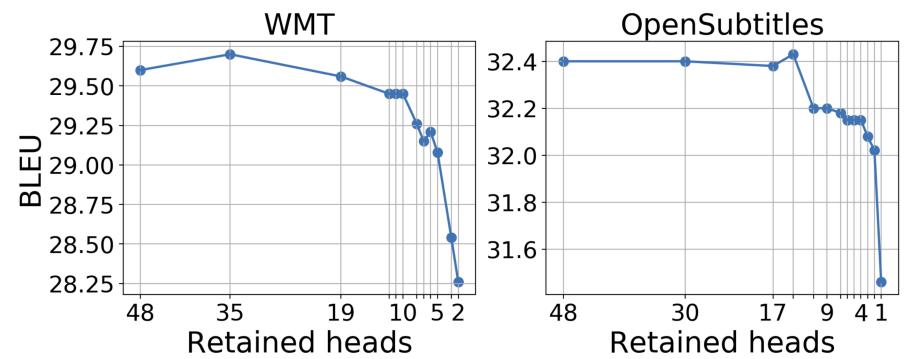
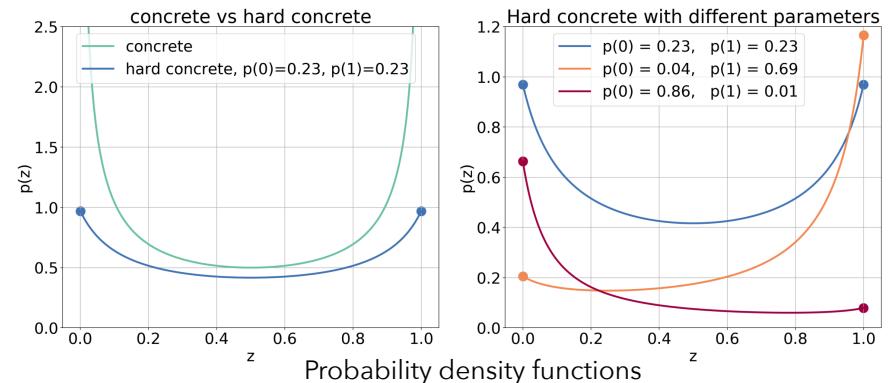
$$L_0(g_{1,1}, \dots, g_{H,L}) = \sum_{l=1}^L \sum_{h=1}^H (1 - \mathbb{I}_{[g_{h,l}=0]}) \dots \text{non-differentiable}$$

$$L_{Concrete} = \sum_{l=1}^L \sum_{h=1}^H (1 - \mathbb{P}(g_{h,l} = 0 | \phi_i))$$

$\phi_i$  ... parameters of the Hard Concrete distribution  
(stretch-and-rectify version of the Gumbel softmax)

$$L(\theta, \phi) = L_{NMT}(\theta, \phi) + \lambda L_{Concrete}(\phi) \text{ & reparametrization trick}$$

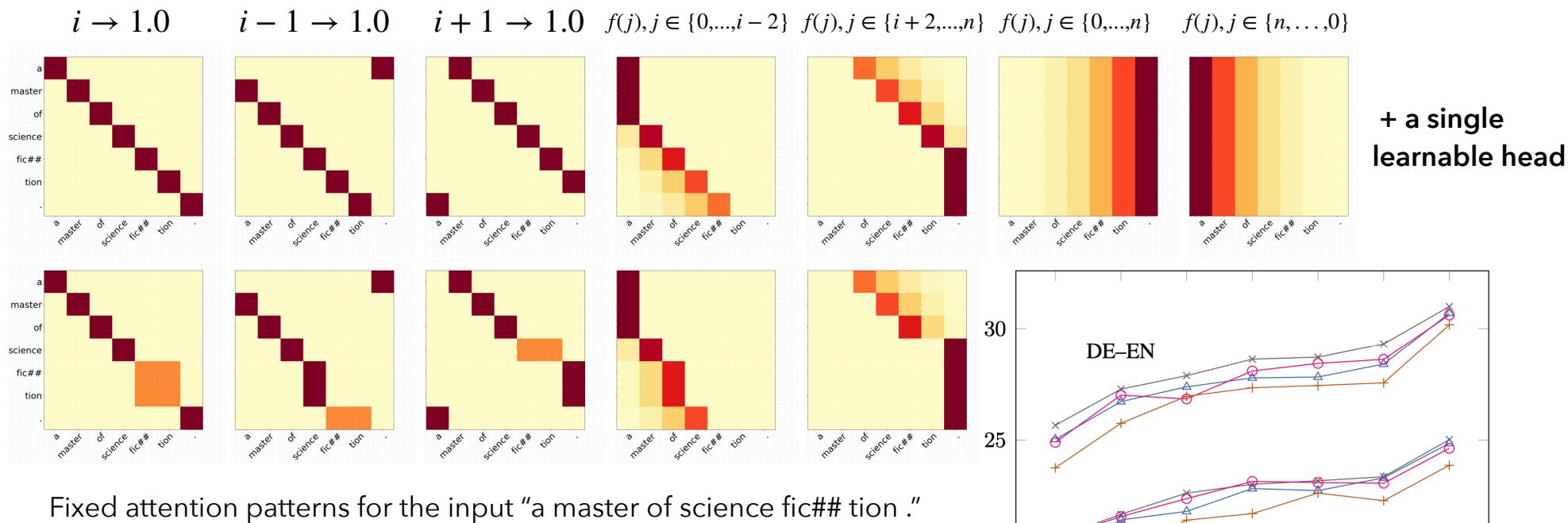
Maddison et al, 2017; Jang et al, 2017; Louizos et al, 2018



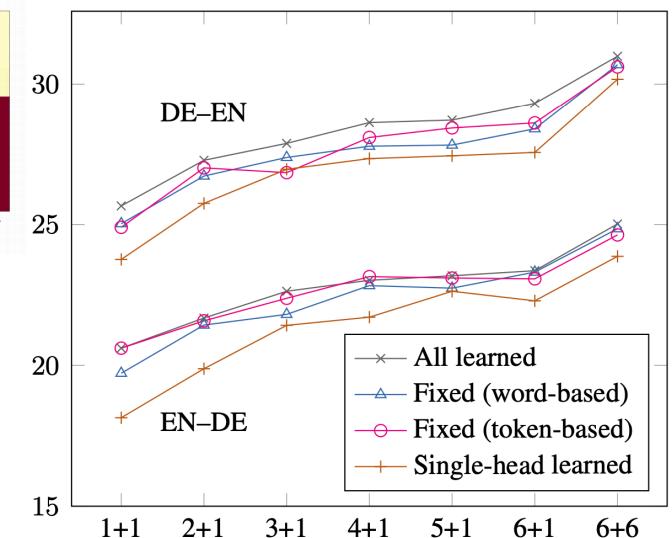
# Fixed attention heads

– Raganato et al, 2019; Transformer for NMT

$$f(i) = \frac{(i+1)^3}{\sum_{i=start}^{end} (i+1)^3}$$



41



# Attention pitfalls



Wiegreffe and Pinter, 2019

## MAYBE

- adversarial attention weights can be learned
- $\exists$  tests to determine when/whether attention can be used as an explanation

Jain and Wallace, 2019

## NO

- attention weights don't correlate with leave-one-out methods
- $\exists$  alternative attention weights with near-identical predictions

Should attention be treated as justification  
for a prediction?

Moradi et al, 2019

## NO

- $\exists$  alternative attention weights with near-identical predictions

Serrano and Smith, 2019

## NO

- analysis based on intermediate representation erasure shows that attention weights often fail to identify representations most important to the model's final decision

Zhong et al, 2019

## MAYBE

- attention was often misaligned with the words that contribute to sentiment
- attention trained with human rationales brings faithful explanations

# Improving attention-based explanations

– Brunner et al, 2020

$E$  ... input embeddings

$W^V$  ... value weights matrix

$V = EW^V$  ... value matrix

$T := EW^V H$

$$A = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

$AT$  ... output of a multi-head attention layer

$\text{rank}(T) \leq \min(d_s, d_v)$

$\text{LN}(T) = \{\tilde{x} \in \mathbb{R}^{1 \times d_s} \mid \tilde{x}^T T = 0\}$

$d_s = 512$  ... input sequence length

$d_v = 64$  ... dimension of value vector

$$\dim(\text{LN}(T)) = d_s - \text{rank}(T) \geq d_s - \min(d_s, d_v) = \begin{cases} d_s - d_v, & d_s > d_v \\ 0, & \text{otherwise} \end{cases}$$

$\forall \tilde{A} \in \{[\tilde{x}_1, \dots, \tilde{x}_{d_s}]\} : \tilde{x}_i^T \in \text{LN}(T) \Rightarrow (\tilde{A} + A)T = AT \Rightarrow \text{attention is not unique}$

# Improving attention-based explanations

– Brunner et al, 2020

$$AT = (A^{\parallel} + A^{\perp})T = A^{\perp}T$$

$$A^{\parallel} \in \text{LN}(T), A^{\perp} \in (\text{LN}(T))^{\perp}$$

$$A^{\perp} = A - \text{Projection}_{\text{LN}(T)}(A) \dots \text{effective attention}$$

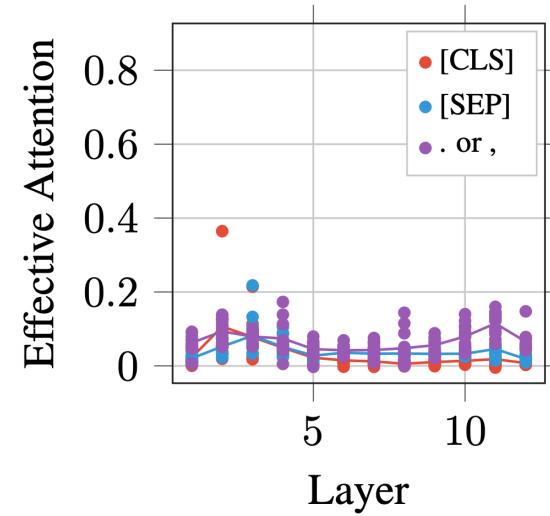
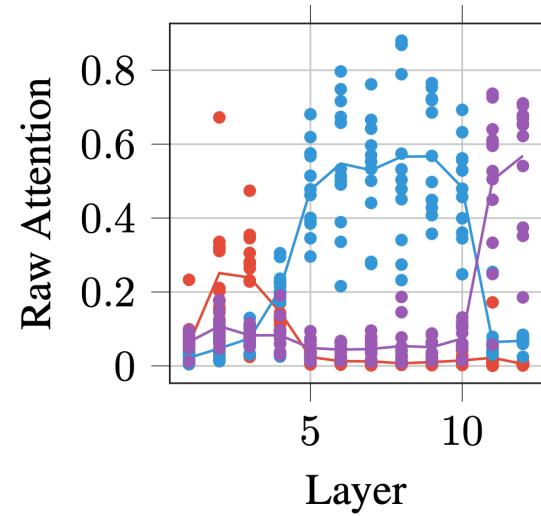
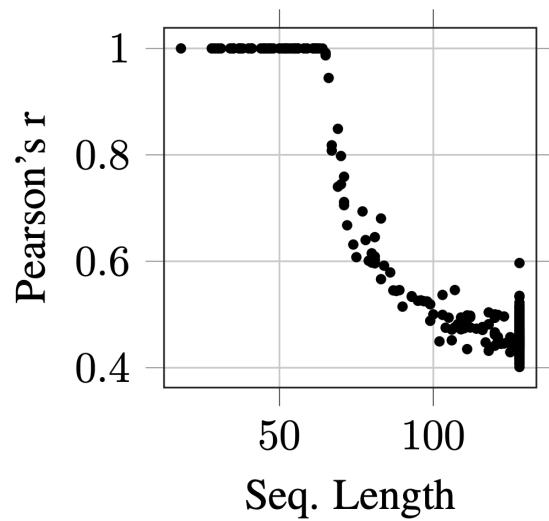


not specified in the paper

1. Make a QR decomposition of T;  $Q \in \mathbb{R}^{d_s \times d_s}$  is an orthogonal matrix,  $R \in \mathbb{R}^{d \times d}$  is an upper triangular matrix
2. An orthonormal set of basis vectors for  $\text{LN}(T^T)$  are the last  $d_s - r$  columns of Q;  $r$  is the rank of Q
3. Project each row  $a_i \Rightarrow P_{\text{LN}(T)}(a_i) = \sum_{i=1}^{d_s-r} \langle a_i, q_{r+i} \rangle q_{r+i}$
4.  $\text{Projection}_{\text{LN}(T)}(A) = [P(a_1), \dots, P(a_{d_s})]^T$

# Improving attention-based explanations

– Brunner et al, 2020



attention patterns associated with the [SEP] and [CLS] tokens are less dominating

# What is next for NLP according to attention



Rogers et al, 2019; BERTology primer

1. What linguistic knowledge is guaranteed with pretrained LMs for end-tasks?
2. What kind of linguistic knowledge available pretrained LMs are not predictive of?  
What are challenging tasks and datasets?
3. What biases pretrained LMs capture? Can they be more transparent about them?
4. Can we make pretrained LMs smaller, faster, and more energy-efficient?

# References 1

- Brooks. A better lesson. 2019. <http://rodneybrooks.com/a-better-lesson/>
- Brunner et al. On Identifiability in Transformers. ICLR 2020.
- Conneau et al. What you can cram into a single \$&!\* vector: Probing sentence embeddings for linguistic properties. ACL 2018.
- Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. TACL 2020.
- Goldberg. Assessing BERT's Syntactic Abilities. Arxiv 2019
- Hewitt and Liang. Designing and Interpreting Probes with Control Tasks. EMNLP 2019.
- Hewitt and Manning. A Structural Probe for Finding Syntax in Word Representations. NAACL 2019.
- Jain and Wallace. Attention is not Explanation. NAACL 2019.
- Jang et al. Categorical Reparameterization with Gumbel-Softmax. ICLR 2017.
- Jawahar et al. What does BERT learn about the structure of language? ACL 2019.
- Kim et al. Probing What Different NLP Tasks Teach Machines about Function Word Comprehension. \*SEM 2019.

# References 2

Kober et al. Temporal and Aspectual Entailment. IWCS 2019.

Kovaleva et al. Revealing the Dark Secrets of BERT. EMNLP 2019.

Li and Talwalkar. Random Search and Reproducibility for Neural Architecture Search. UAI 2019. Louizos et al. Learning Sparse Neural Networks through  $L_0$  regularization. ICLR 2018.

Maddison et al. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. ICLR 2017.

May et al. On Measuring Social Biases in Sentence Encoders. NAACL 2019.

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

Moradi et al. Interrogating the Explanatory Power of Attention in Neural Machine Translation. WNGT @ EMNLP 2019.

Niven and Kao. Probing Neural Network Comprehension of Natural Language Arguments. ACL 2019.

Perone et al. Evaluation Of Sentence Embeddings In Downstream And Linguistic Probing Tasks. Arxiv 2018.

Peters et al. Dissecting Contextual Word Embeddings: Architecture and Representation. EMNLP 2018.

# References 3

- Pham et al. Efficient Neural Architecture Search via Parameter Sharing. ICML 2018.
- Poliak et al. Collecting Diverse Natural Language Inference Problems for Sentence Representation Evaluation. EMNLP 2018.
- Raganato et al. Fixed Encoder Self-Attention Patterns in Transformer-Based Machine Translation. Arxiv 2020
- Reif et al. Visualizing and Measuring the Geometry of BERT. NeurIPS 2019.
- Rogers et al. A Primer in BERTology: What we know about how BERT works. Arxiv 2020.
- Serrano and Smith. Is Attention Interpretable? ACL 2019.
- Shwartz and Dagan. Still a Pain in the Neck: Evaluating Text Representations on Lexical Composition. TACL 2019.
- Strubell et al. Energy and Policy Considerations for Deep Learning in NLP. ACL 2019.
- Sutton. The Bitter Lesson. 2019. <http://www.incompleteideas.net/Incldeas/BitterLesson.html>
- Tenney et al. What Do You Learn From Context? Probing For Sentence Structure In Contextualized Word Representations. ICLR 2019a.
- Tenney et al. BERT Redisovers the Classical NLP Pipeline. ACL 2019b.

# References 4

- Voita et al. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. ACL 2019.
- Zhao et al. Gender Bias in Contextualized Word Embeddings. NAACL 2019.
- Zhang and Bowman. Language Modeling Teaches You More Syntax than Translation Does: Lessons Learned Through Auxiliary Task Analysis. Blackbox @ EMNLP 2018.
- Zhong et al. Fine-grained Sentiment Analysis with Faithful Attention. Arxiv 2019.
- Wiegreffe and Pinter. Attention is not not Explanation. EMNLP 2019.