What does BERT look at? An analysis of BERT's attention.

Stanford + Facebook AI, ACL Workshop on Black-box NLP 2019

Mathieu Ravaut

Wednesday, April 8th 2020

layer6

Motivation

Sesame Street now omnipresent in NLP:

- ELMo (Allen AI, 2018)
- **BERT (Google, 2018)**
- XLNet (Google + CMU, 2019)
- RoBERTa (Facebook, 2019)
- AlBERT (Google, 2019)
- BART (Facebook, 2020)
- ELECTRA (Google + Stanford, 2020)
- ...

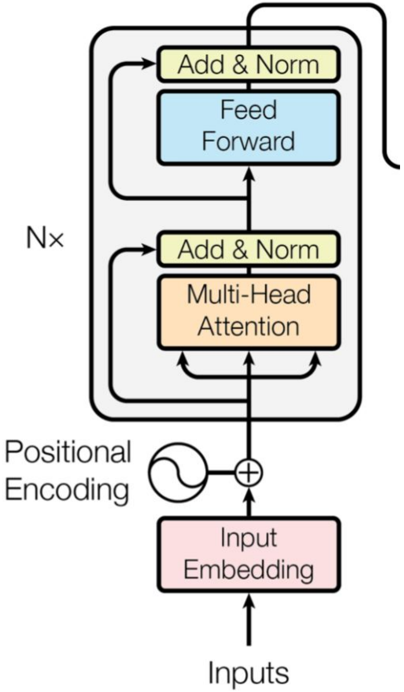
All are **Transformer**-based architectures.
All rely on **attention.**

Motivation

SQUAD 2.0
Leaderboard (April 2020):

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Mar 12, 2020	ALBERT + DAAF + Verifier (ensemble) *PINGAN Omni-Sinitic*	**90.386**	**92.777**
2 Jan 10, 2020	Retro-Reader on ALBERT (ensemble) *Shanghai Jiao Tong University* http://arxiv.org/abs/2001.09694	90.115	92.580
3 Nov 06, 2019	ALBERT + DAAF + Verifier (ensemble) *PINGAN Omni-Sinitic*	90.002	92.425
4 Sep 18, 2019	ALBERT (ensemble model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	89.731	92.215
4 Feb 25, 2020	Albert_Verifier_AA_Net (ensemble) *QIANXIN*	89.743	92.180
5 Jan 23, 2020	albert+transform+verify (ensemble) *qianxin*	89.528	92.059
6 Mar 06, 2020	ALBert-LSTM (ensemble) *oppo.tensorlab*	89.269	91.777
7 Dec 08, 2019	ALBERT+Entailment DA (ensemble) *CloudWalk*	88.761	91.745
8 Mar 06, 2020	ELECTRA (single model) *Google Brain & Stanford*	88.716	91.365

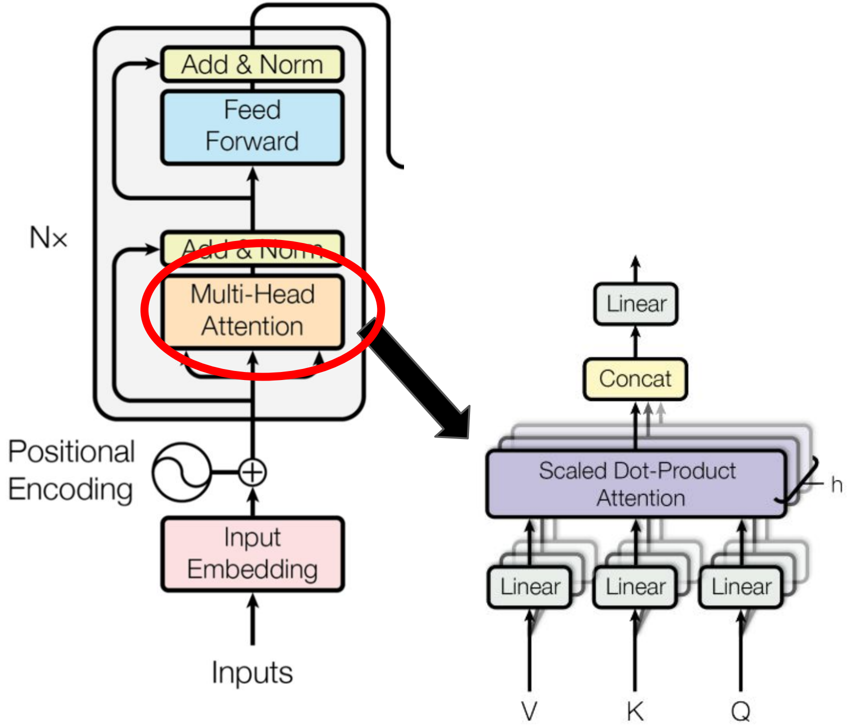
BERT



Encoder block

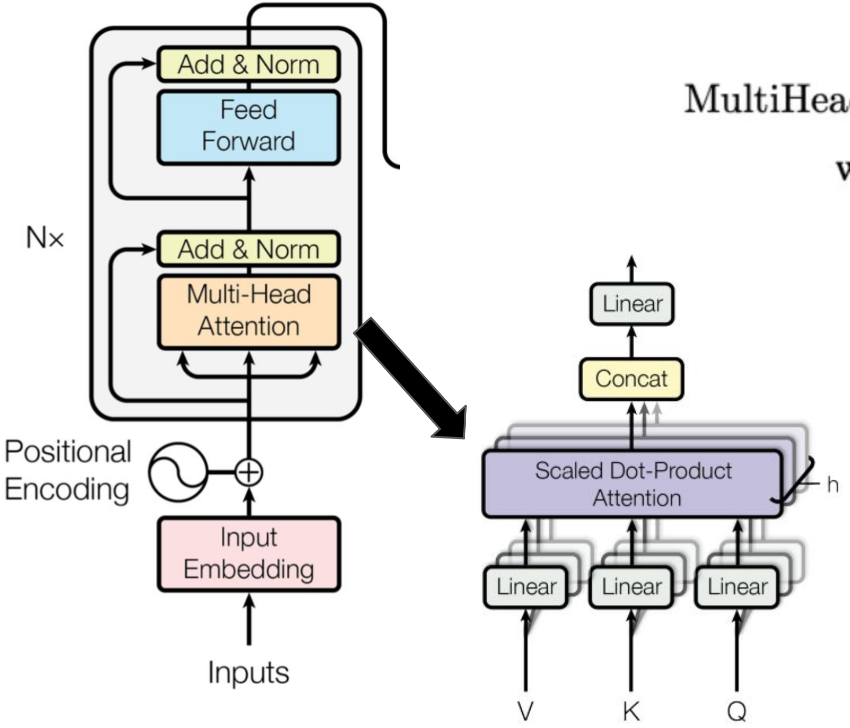
layer6

BERT



Encoder block

BERT



$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_\mathrm{h})W^O$$
$$\text{where } \mathrm{head}_\mathrm{i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

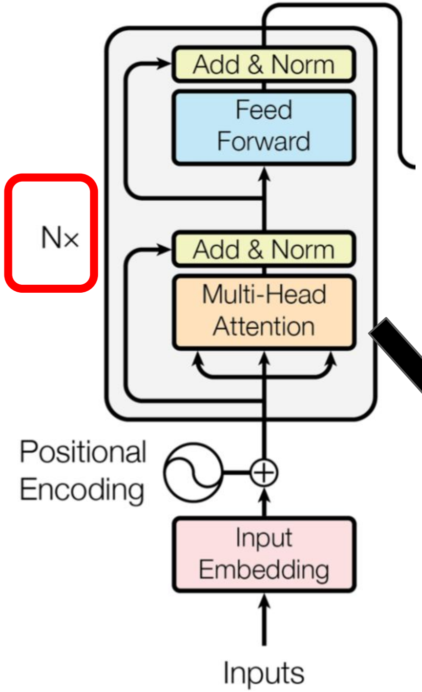
Self-attention: take Q=K=V=input

Encoder block

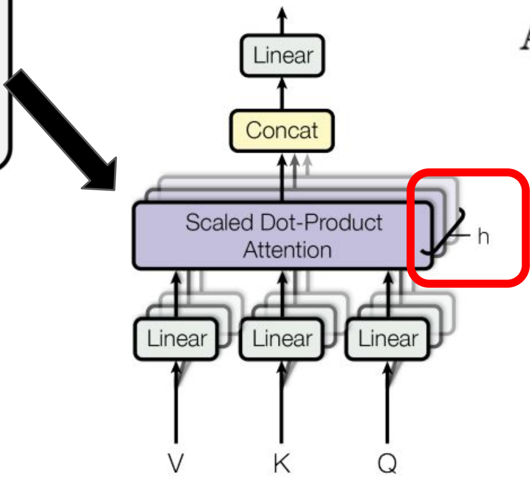
layer6

BERT

12 layers



Encoder block



$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, \ldots, \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Self-attention: take Q=K=V=input

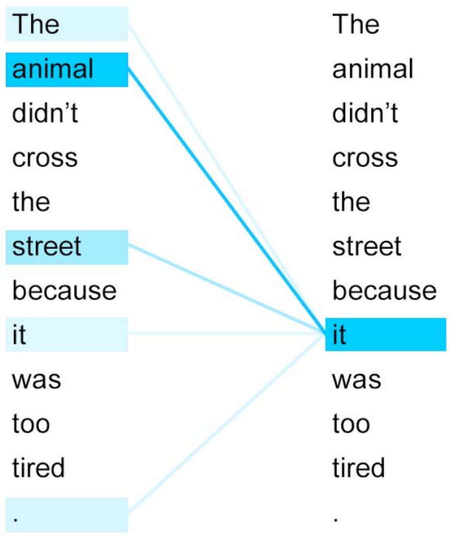
12 heads per layer

144 heads / attention distributions

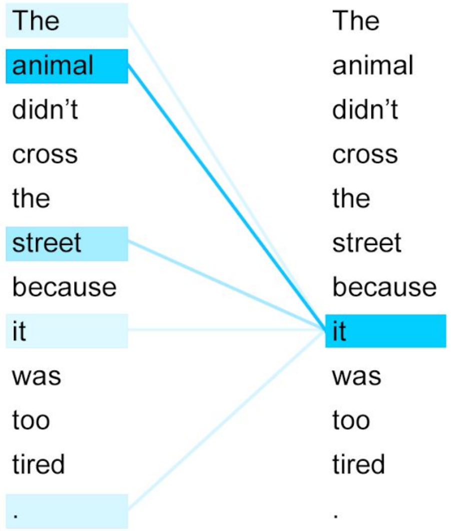
layer6

Self-attention weights

Looking at the self-attention distribution for “it” from 5th to 6th layer of the Transformer:



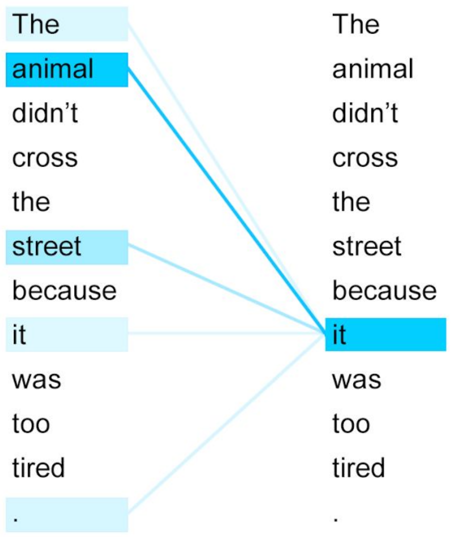
Self-attention weights



The weight between each pair of words (i,j) is written:

$$\alpha_{ij} = \frac{\exp\left(q_i^T k_j\right)}{\sum_{l=1}^{n} \exp\left(q_i^T k_l\right)}$$

Self-attention weights



The weight between each pair of words (i,j) is written:

$$\alpha_{ij} = \frac{\exp\left(q_i^T k_j\right)}{\sum_{l=1}^{n} \exp\left(q_i^T k_l\right)}$$

- Are these connections **meaningful**?
- Can we use these attention heads as **already trained classifiers**?

layer 6

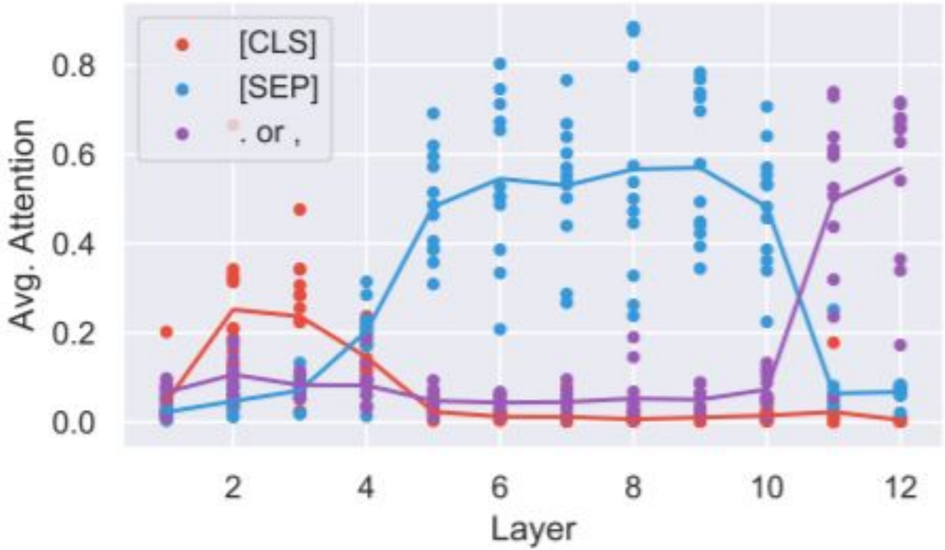
High-level patterns

How much do attention heads attend to **previous / current / next** token?

- Most heads put little weight on the current token.
- In layers 2,4,7,8: **4 heads** put > 50% of attention weight on **previous** token.
- In layers 1,2,3,6: **5 heads** put > 50% on the **next token**.

High-level patterns

What about the always present [SEP] and [CLS], or [.] tokens?

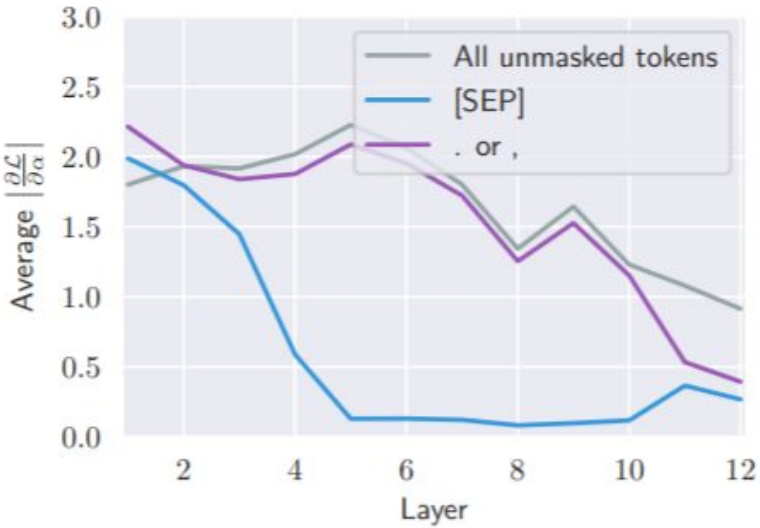


layer 6

High-level patterns

Do heads attend so much to [SEP] because it's the "default" to attend to when the head's function is not called for (ex: noun in a head focused on verb-direct-obj links)?

Norm of gradient with respect to attention weight.



Small magnitude for [SEP]: changing the attention weights to [SEP] does not change much the BERT outputs.

layer6

High-level patterns

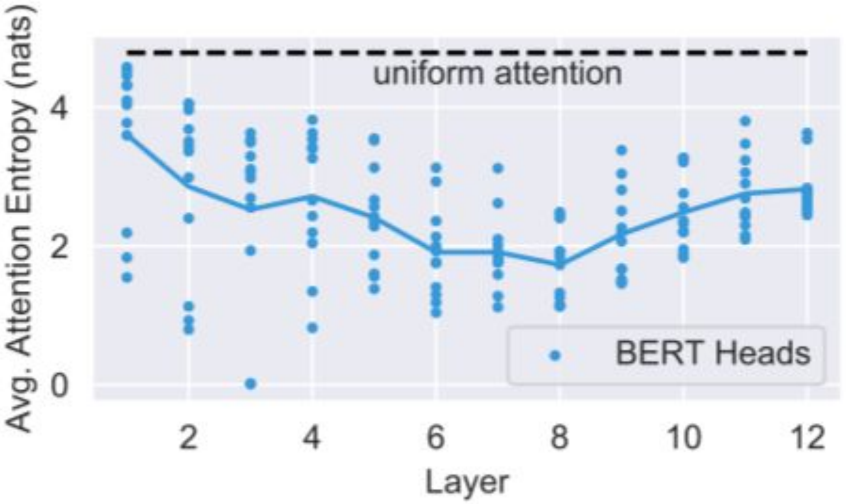


Figure 4: Entropies of attention distributions. In the first layer there are particularly high-entropy heads that produce bag-of-vector-like representations.

Entropy over the attention distribution for **[CLS] token only**.

High-level patterns

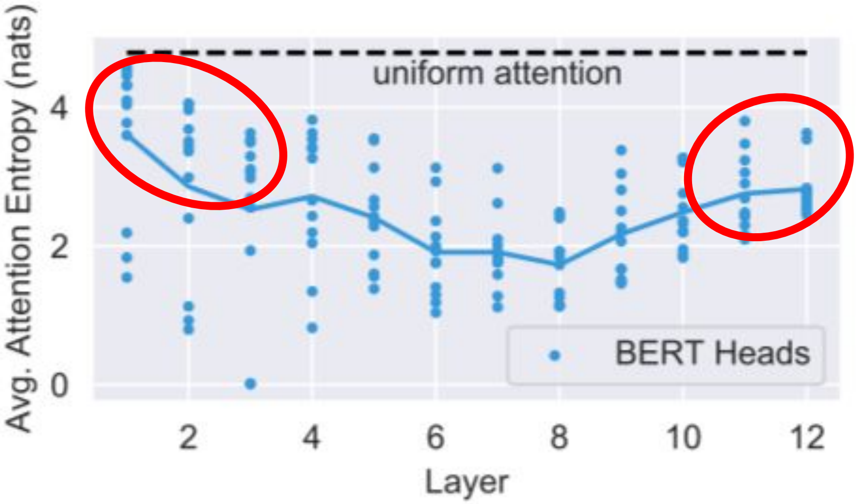
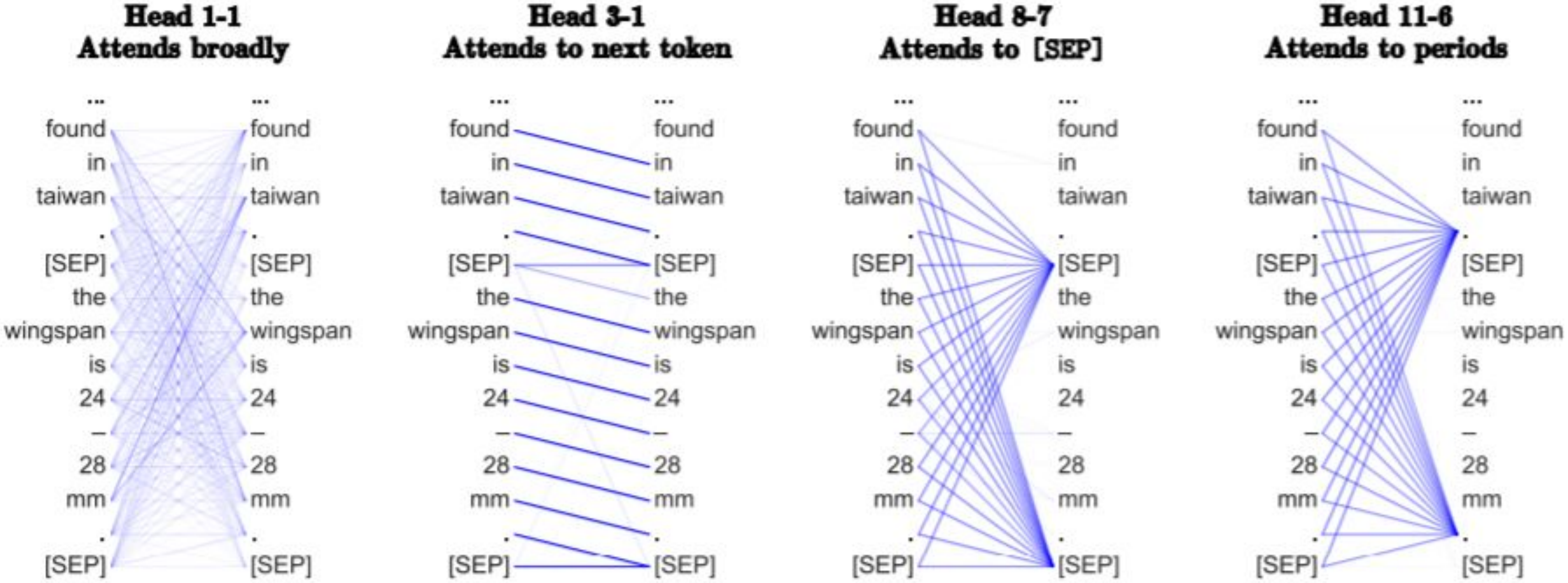


Figure 4: Entropies of attention distributions. In the first layer there are particularly high-entropy heads that produce bag-of-vector-like representations.

High-level patterns



layer 6

Building models with attention heads

Goal is to evaluate attention heads at word-level tasks.

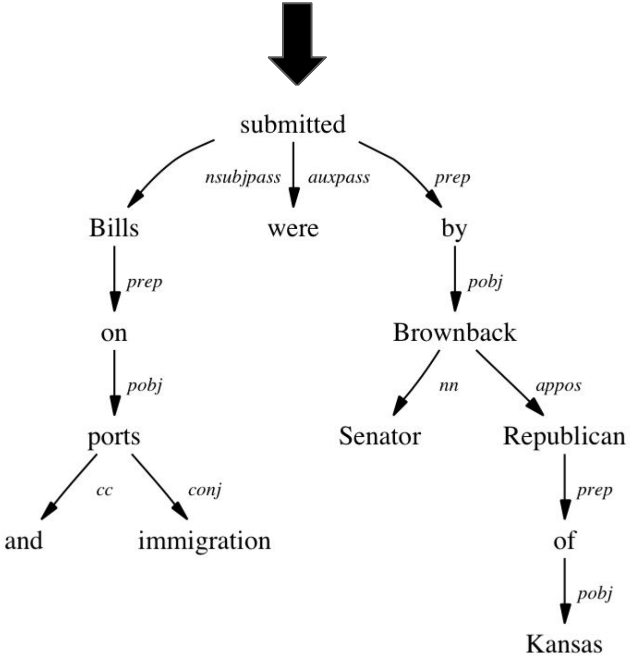
BERT uses byte-pair tokenization: ~8% of ***words are split into multiple tokens***.

- Attention **to** word = [$token_1$, … , $token_k$] : **sum**(weight to each token)
- Attention **from** word = [$token_1$, … , $token_k$] : **mean**(weight from each token)

layer 6

Dependency syntax

Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas.



Dependency syntax

- At most one connection between 2 words.

- Different types of dependencies:
  - *prep*: preposition
  - *aux*: auxiliary
  - *advmod*: adverb modifier

- Evaluation with **accuracy**.

- Dataset is the **PennTreebank** (WSJ) with Stanford Dependencies labels.

Dependency syntax with attention heads

10 most common dependencies

Other dependencies where heads do well.

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
prep	7-4	66.7	61.8 (-1)
pobj	9-6	**76.3**	34.6 (-2)
det	8-11	**94.3**	51.7 (1)
nn	4-10	70.4	70.2 (1)
nsubj	8-2	58.5	45.5 (1)
amod	4-10	75.6	68.3 (1)
dobj	8-10	**86.8**	40.0 (-2)
advmod	7-6	48.8	40.2 (1)
aux	4-10	81.1	71.5 (1)
poss	7-6	**80.5**	47.7 (1)
auxpass	4-10	**82.5**	40.5 (1)
ccomp	8-1	**48.8**	12.4 (-2)
mark	8-2	**50.7**	14.5 (2)
prt	6-7	**99.1**	91.4 (-1)

Dependency syntax with attention heads

10 most common dependencies

Other dependencies where heads do well.

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
prep	7-4	66.7	61.8 (-1)
pobj	9-6	**76.3**	34.6 (-2)
det	8-11	**94.3**	51.7 (1)
nn	4-10	70.4	70.2 (1)
nsubj	8-2	58.5	45.5 (1)
amod	4-10	75.6	68.3 (1)
dobj	8-10	**86.8**	40.0 (-2)
advmod	7-6	48.8	40.2 (1)
aux	4-10	81.1	71.5 (1)
poss	7-6	**80.5**	47.7 (1)
auxpass	4-10	**82.5**	40.5 (1)
ccomp	8-1	**48.8**	12.4 (-2)
mark	8-2	**50.7**	14.5 (2)
prt	6-7	**99.1**	91.4 (-1)

Heads are evaluated in both directions (to/from the current word) and we take the highest weight.

layer6

Dependency syntax with attention heads

10 most common dependencies

Other dependencies where heads do well.

Layer number - head number

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
prep	7-4	66.7	61.8 (-1)
pobj	9-6	**76.3**	34.6 (-2)
det	8-11	**94.3**	51.7 (1)
nn	4-10	70.4	70.2 (1)
nsubj	8-2	58.5	45.5 (1)
amod	4-10	75.6	68.3 (1)
dobj	8-10	**86.8**	40.0 (-2)
advmod	7-6	48.8	40.2 (1)
aux	4-10	81.1	71.5 (1)
poss	7-6	**80.5**	47.7 (1)
auxpass	4-10	**82.5**	40.5 (1)
ccomp	8-1	**48.8**	12.4 (-2)
mark	8-2	**50.7**	14.5 (2)
prt	6-7	**99.1**	91.4 (-1)

layer6

Dependency syntax with attention heads

10 most common dependencies

Other dependencies where heads do well.

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
prep	7-4	66.7	61.8 (-1)
pobj	9-6	**76.3**	34.6 (-2)
det	8-11	**94.3**	51.7 (1)
nn	4-10	70.4	70.2 (1)
nsubj	8-2	58.5	45.5 (1)
amod	4-10	75.6	68.3 (1)
dobj	8-10	**86.8**	40.0 (-2)
advmod	7-6	48.8	40.2 (1)
aux	4-10	81.1	71.5 (1)
poss	7-6	**80.5**	47.7 (1)
auxpass	4-10	**82.5**	40.5 (1)
ccomp	8-1	**48.8**	12.4 (-2)
mark	8-2	**50.7**	14.5 (2)
prt	6-7	**99.1**	91.4 (-1)

Simple baseline: predict the word at fixed distance *offset* from the current word

layer6

Dependency syntax with attention heads

10 most common dependencies

Other dependencies where heads do well.

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
prep	7-4	66.7	61.8 (-1)
pobj	9-6	**76.3**	34.6 (-2)
det	8-11	**94.3**	51.7 (1)
nn	4-10	70.4	70.2 (1)
nsubj	8-2	58.5	45.5 (1)
amod	4-10	75.6	68.3 (1)
dobj	8-10	**86.8**	40.0 (-2)
advmod	7-6	48.8	40.2 (1)
aux	4-10	81.1	71.5 (1)
poss	7-6	**80.5**	47.7 (1)
auxpass	4-10	**82.5**	40.5 (1)
ccomp	8-1	**48.8**	12.4 (-2)
mark	8-2	**50.7**	14.5 (2)
prt	6-7	**99.1**	91.4 (-1)

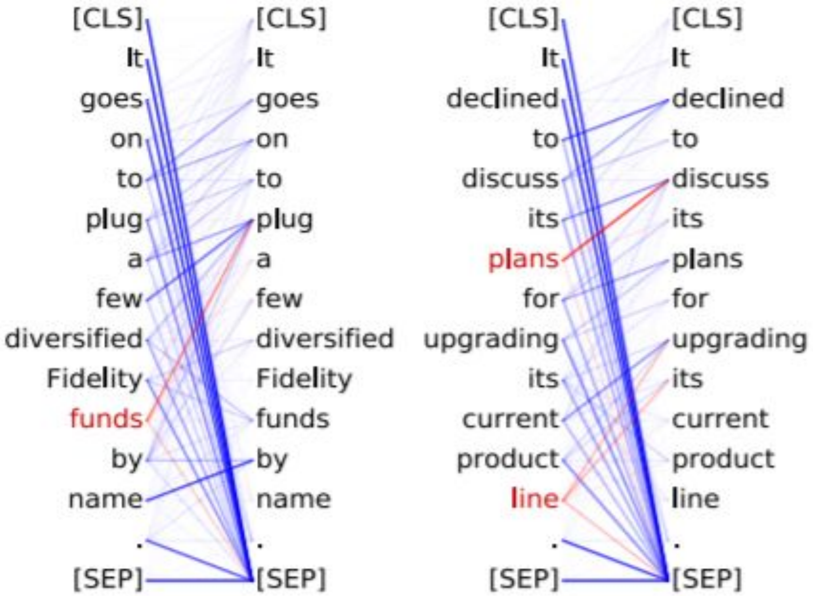
Best offset value (predict the word 2 positions before the current word).

layer6

Dependency syntax with attention heads

Head 8-10

- **Direct objects** attend to their verbs
- 86.8% accuracy at the `dobj` relation



Dependency syntax with attention heads

10 most common dependencies

Other dependencies where heads do well.

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
prep	7-4	66.7	61.8 (-1)
pobj	9-6	**76.3**	34.6 (-2)
det	8-11	**94.3**	51.7 (1)
nn	4-10	70.4	70.2 (1)
nsubj	8-2	58.5	45.5 (1)
amod	4-10	75.6	68.3 (1)
dobj	8-10	**86.8**	40.0 (-2)
advmod	7-6	48.8	40.2 (1)
aux	4-10	81.1	71.5 (1)
poss	7-6	**80.5**	47.7 (1)
auxpass	4-10	**82.5**	40.5 (1)
ccomp	8-1	**48.8**	12.4 (-2)
mark	8-2	**50.7**	14.5 (2)
prt	6-7	**99.1**	91.4 (-1)

- Attention heads are bad overall (34.5 acc)
- Some heads are very good for one or a few dependencies (ex: 4-10).

layer 6

Coreference resolution

- *Bill said he would come.*
  he here refers to *Bill*.

- Does the head word of a coreferent mention most attend to the head of one of that mention antecedent?

- Evaluate the fraction of time this is the case (**accuracy**).

- Dataset is **CoNLL-2012**.

Coreference resolution

Model	All	Pronoun	Proper	Nominal
Nearest	27	29	29	19
Head match	52	47	67	40
Rule-based	69	70	77	60
Neural coref	83*	–	–	–
Head 5-4	65	64	73	58

*Only roughly comparable because on non-truncated documents and with different mention detection.

Table 2: Accuracies (%) for systems at selecting a correct antecedent given a coreferent mention in the CoNLL-2012 data. One of BERT's attention heads performs fairly well at coreference.

- Display results from BERT's best head (head 5-4).
- BERT is better than two simple baselines but not a rule-based method nor a neural method.

layer6

Dependency syntax with multiple heads

- Idea: build a simple classifier on top of the attention maps.

- Use fixed Glove embeddings for words.

$$p(i|j) \propto \exp\left(\sum_{k=1}^{n} W_{k,:}(v_i \oplus v_j)\alpha_{ij}^{k} + U_{k,:}(v_i \oplus v_j)\alpha_{ji}^{k}\right)$$

- W, U are learned matrices of weights
  v are the Glove embeddings
  k is the head index

layer 6

Dependency syntax with multiple heads

Model	UAS
Structural probe	80 UUAS*
Right-branching	26
Distances + GloVe	58
Random Init Attn + GloVe	30
Attn	61
Attn + GloVe	77

Table 3: Results of attention-based probing classifiers on dependency parsing. A simple model taking BERT attention maps and GloVe embeddings as input performs quite well. *Not directly comparable to our numbers; see text.

Clustering heads

- Are heads in the same layer **similar** to each other?
- We can compare attention distributions with the Jensen-Shannon divergence:

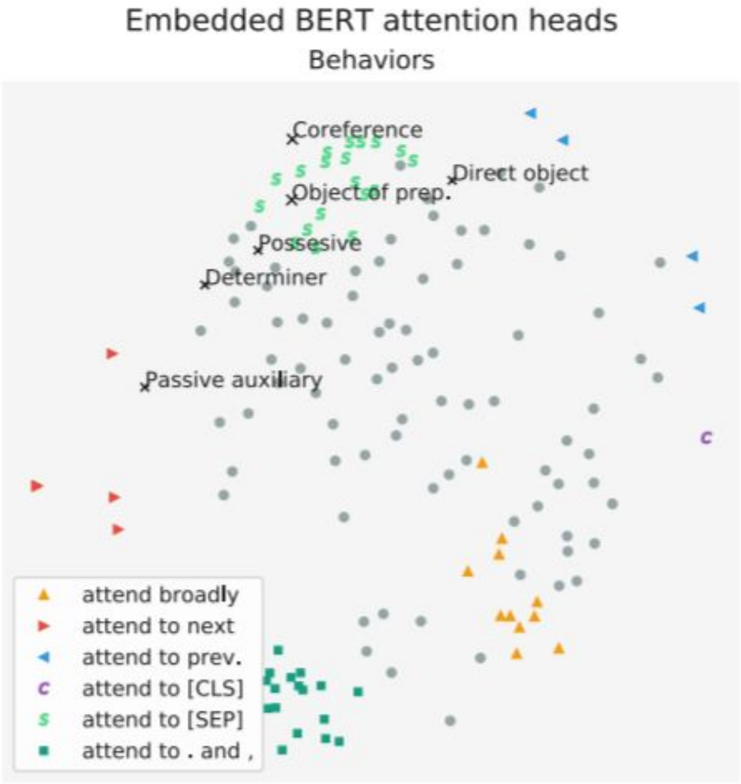
$$\sum_{\text{token}\in\text{data}} JS(\mathrm{H}_i(\text{token}), \mathrm{H}_j(\text{token}))$$

- Visualization in 2D by applying multi-dimensional scaling.

layer 6

Clustering heads

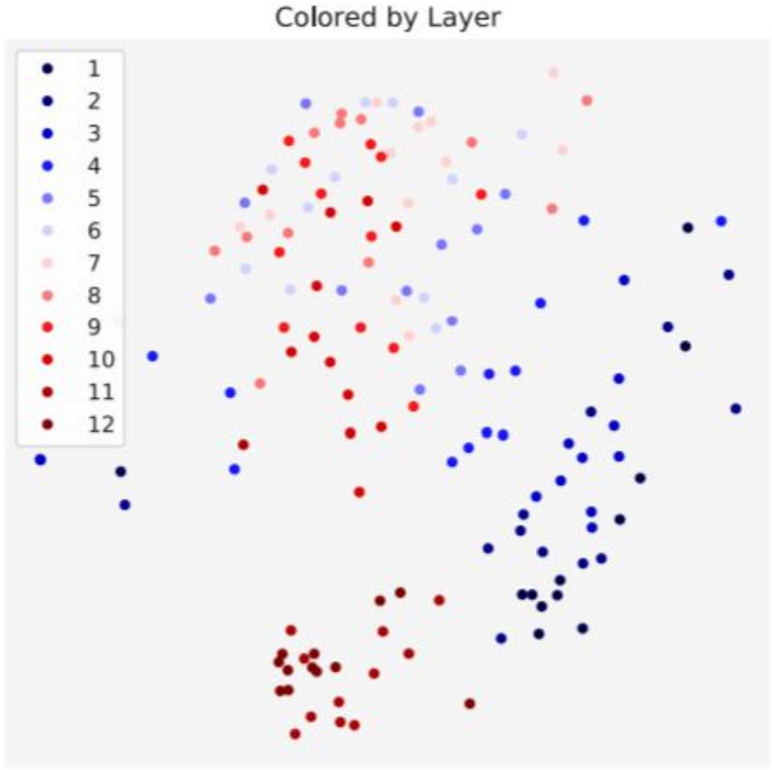
Heads tagged with their **function**.



layer 6

Clustering heads

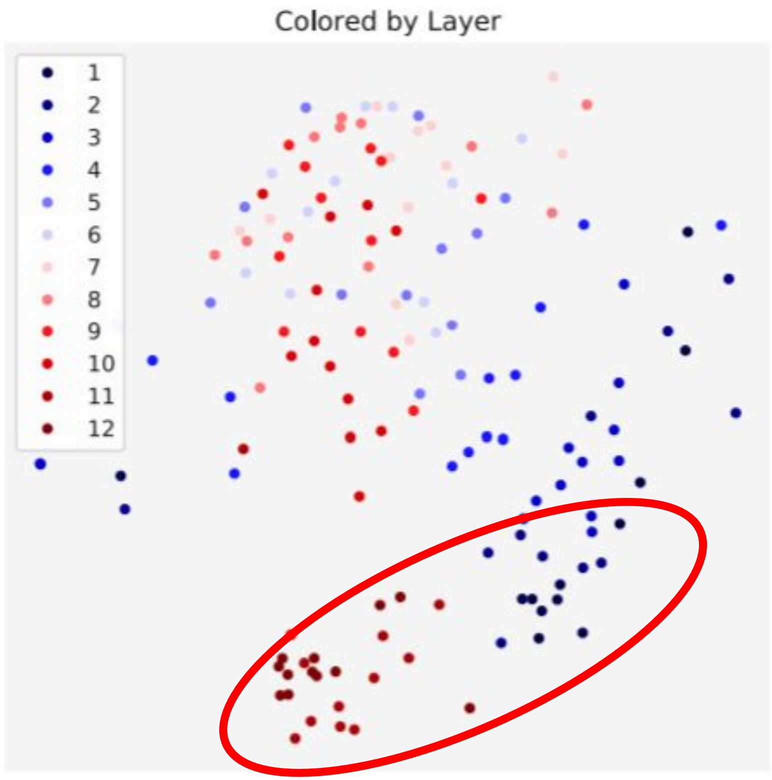
Heads colored by **layer**.



layer6

Clustering heads

Heads colored by **layer**.



First and last layers similar?

layer6

Thank you!