Some linguistic correlates of gradients and attention weights in BERT

Matthijs Westera, Universitat Pompeu Fabra

BlackboxNLP 2019

Some findings

In the BERT transformer model:

- More info flows from noun to pronoun if they corefer;
- Open-class tokens more informative than closed-class;

With differences per layer and per measure used.

Measures of interest

I consider 3 notions of 'information flow' in BERT, from token t_1 to t_2:

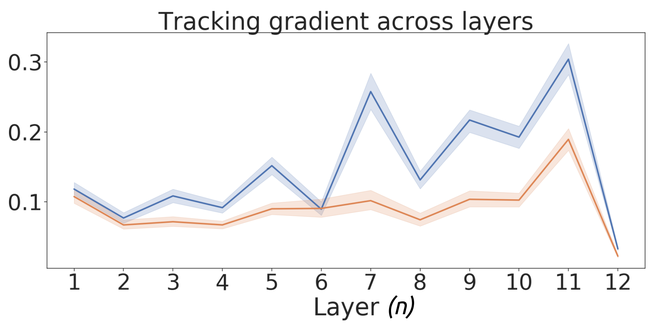
1. $G_n(t_1,t_2)$: magnitude (*2*-norm) of gradient of t_2 at layer n w.r.t. t_1 at layer $n-1$.
2. $G_n*(t_1,t_2)$: likewise, but w.r.t. t_1 at embedding layer.
3. $A_n(t_1,t_2)$: attention weight of t_2 at layer n w.r.t. t_1, averaged over all attention heads.

G_n and A_n are correlated of course: Spearman coef. = .41.

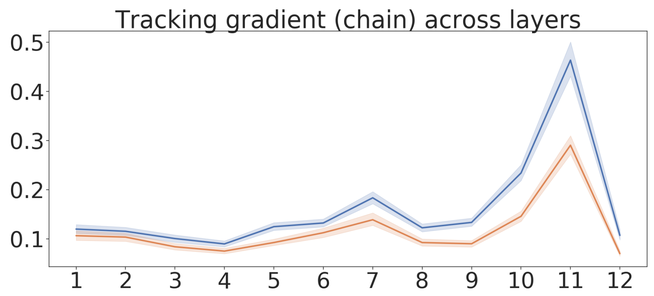
Coreference

t_1 = noun	coreference
t_2 = pronoun	no coreference

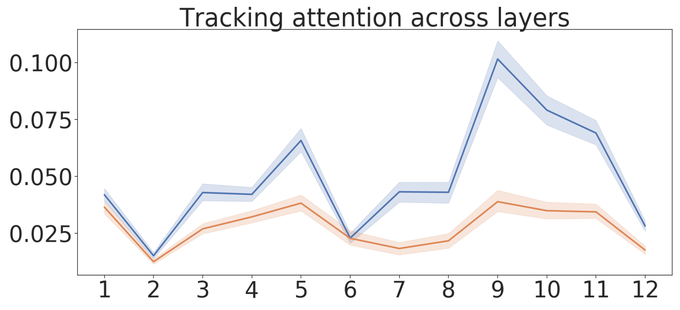
$G_n(t_1,t_2)$



$G_n*(t_1,t_2)$



$A_n(t_1,t_2)$



Some notes:

- Clear effect of coreference.
- Though in different layers for *G* vs. *A*.
- Effect persists if *distance* taken into account.

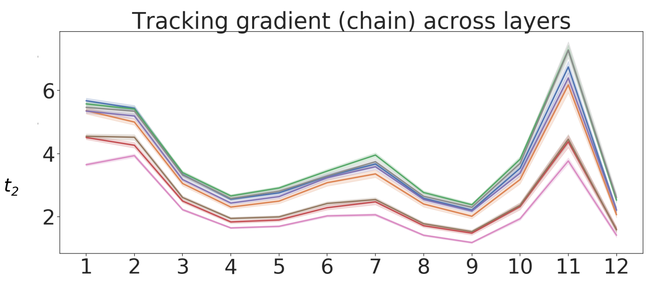
Data used

I apply BERT to random subsets of 500 sentences from:

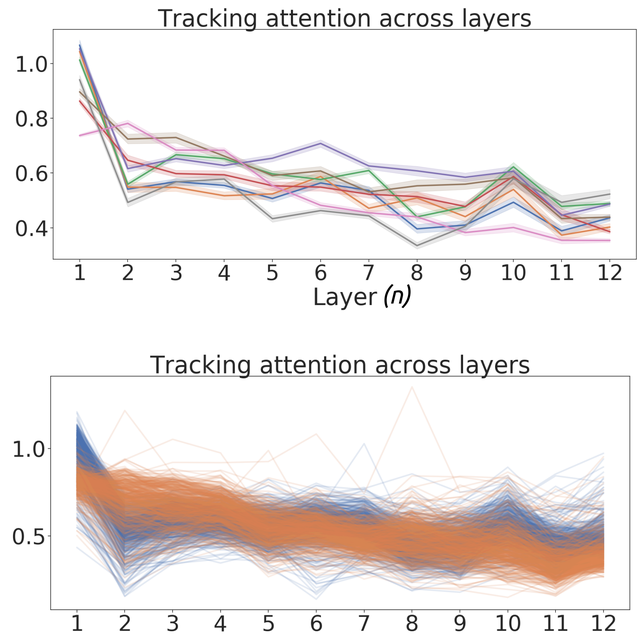
- OntoNotes, for coreference;
- Universal Dependencies (GUM portion) for dependency parses and POS tags.

Parts of speech / open vs. closed

$G_n*(t_1,t_2)$ *averaged over all* t_2



$A_n(t_1,t_2)$ *averaged over all* t_2



Some notes:

- *G** shows persistent effect, but this is almost wholly due to difference in *G* at layer 1.
- *A* is more messy.

Discussion

- Existing work often looks at what information is contained in hidden representations (diagn. classifiers).
- What might be the value of looking instead, or in addition, at how it got there?

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 715154). This paper reflects the authors' view only, and the EU is not responsible for any use that may be made of the information it contains.

Thanks also to the PyTorch developers and to the people behind huggingface/pytorch-pretrained-BERT.