Some additional experiments extending the tech report ”Assessing BERT’s Syntactic Abilities” by Yoav Goldberg

Thomas Wolf[1]

[1]Huggingface Inc.

January 2019

1 Introduction

This document report a few additional experiments extending Yoav Goldberg’s tech report ”Assessing BERT’s Syntactic Abilities”, which can be found at `http://u.cs.biu.ac.il/~yogo/bert-syntax.pdf`.

The present document is an extension of Yoav Goldberg’s tech report and as such, I won’t go into all the details of the methodology used. The reader is kindly asked to read Yoav Goldberg’s tech report prior to the present document.

The experiments described in Yoav Goldberg’s tech report shed an interesting light on the strong performances of the BERT model [Devlin et al., 2018] on English syntactic phenomena.

However, directly comparing BERT to a classical LSTM language model is complicated by the many differences between these two architectures and training/evaluation setups:

- *Architecture*: Transformer VS. Recurrent Neural Network
- *Training objective*: Masked Language Modeling VS. Language Modeling
- *Evaluation method*: use of sentences prefix + suffix for BERT VS. only prefix for Language Models
- *Training dataset*

In the present document, I report a few additional experiments by evaluating additional settings for the Transformer model on the same evaluation data. More precisely, the additional experiments are:

- evaluating the OpenAI Generative Pre-trained Transformer of Radford et al. [2018][1] which is a Transformer model with an architecture highly

[1] `https://blog.openai.com/language-unsupervised/`

similar to BERT (see discussion below) but has been pre-trained with a Language Modeling objective on the Toronto Book Corpus [Zhu et al., 2015] only, and

- evaluating BERT when it is only supplied with a prefix.

2 Experimental details

In my experiments, I used the PyTorch implementations of the OpenAI GPT and BERT [2] with the pre-trained models respectively supplied by OpenAI and Google AI.

The code to reproduce these additional experiments is available at `https://github.com/huggingface/bert-syntax`

2.1 Experiments with OpenAI GPT

The OpenAI GPT model is trained with a language modeling objective and is thus a uni-directional model in which a token probability is a function of the previous sentence tokens only.

In the experiments, I tried two settings: (i) use only the prefix of a sentence and predict the probability of the focus word and (ii) compute the join probability of {focus-word, postfix} conditioned on the prefix by using the chain-rule and the sequentially computed token probabilities (similarly to what was done in Marvin and Linzen [2018]).

Unlike BERT, OpenAI GPT should be able to predict a missing portion of arbitrary length. To simplify the comparison with the BERT experiments, I filtered the stimuli to keep only the ones that were used in the BERT experiments. I thus discarded in particular the stimuli in which the focus verb or its plural/singular inflection does not appear as a single word in the BERT word-piece-based vocabulary so that the data used for the experiments is the same as in the BERT settings.

In all experiments, when a missing word is split in several sub-words by OpenAI GPT's Byte-Pair-Encoding tokenizer, I compute the join probability of the sequence of sub-words (conditioned on the prefix of the sentence)[3]. I also tested discarding the sentences in which at least one of the focus words was split by the BPE process. This additional filtering didn't have any significant impact on the results.

2.2 Experiments on BERT with prefix-only

I also tried to supply BERT with the prefix only, cutting the postfix of each stimuli after the masked token.

[2] both available at `https://github.com/huggingface/pytorch-pretrained-BERT`

[3] As usual, the conditional probability is obtained by applying a softmax function over the output vocabulary to the output logits of the OpenAI GPT model. To prevent underflow I worked with the log of the softmax output.

Note that in this case there is some discrepancy between the evaluation setup and the training setup since the model was supplied with well-formed sentences during the training and is supplied with only prefixes during testing (in all the cases in which the focus word is not the last word). However, note also that, given the pretraining scheme of BERT in which a random choice of tokens is masked, the last token will have also been masked for at least some training instances so that the general setup of using only the left context to predict a word is not fully unknown to the BERT model.

3 Comparing the LSTM, BERT and OpenAI GPT architectures

The OpenAI GPT model is trained as a language model and thus uses the same training objective as the LSTMs described in Linzen et al. [2016], Gulordava et al. [2018] and Marvin and Linzen [2018].

Its training objective differs from the masked-language modeling objective used to train BERT.

The dataset used for pre-training OpenAI GPT is the Toronto Book Corpus (TBC) [Zhu et al., 2015] and is thus different from BERT which is trained on both TBC and Wikipedia. OpenAI GPT training corpus also differs from the LSTM experiments reported in Linzen et al. [2016], Gulordava et al. [2018] and Marvin and Linzen [2018].

Apart from the training objective and dataset, the OpenAI GPT model is very similar to the BERT-Base model, having the same general architecture, the same number of layers and size of hidden-states, and a similar number of parameters (about 110 million parameters).

The main[4] architectural differences are:

- the use of causal-masked attention heads in OpenAI GPT, and
- the use a Byte-Pair-Encoding vocabulary in OpenAI GPT (instead of WordPieces).

Regarding the second difference which can seems anectodal, it should be emphasized that several works (at least in Machine Translation and Language Modeling) have noted how the sub-word vocabulary used by a model can have a strong impact on its performances (see [Sennrich et al., 2016, Baevski and Auli, 2018] for instance).

4 Results

The results can be found on tables 1, 2 and 3.

[4] Two small additional differences I noted when re-implementing both models in PyTorch are (i) differences in the way the geLU activation function is computed and (ii) an additional LayerNorm layer after the embeddings layer

On the Linzen et al. [2016] and Gulordava et al. [2018] stimuli, BERT outperform the OpenAI GPT by a strong margin.

However, on the more granular stimuli from Marvin and Linzen [2018], OpenAI GPT surprinsingly obtain scores higher than the BERT model on several tested conditions (indicated in bold on table 3 while the underlined scores are the top scores in the sub-categories) indicative of a more subtle interaction between architecture, training objective and training dataset.

One interesting trend that can be observed on Table 3, is that the OpenAI GPT scores seems generally correlated with the LSTM scores (in particular performing worse on the "Across XX" settings) albeit with higher scores.

5 Discussion

These additional experiments seems to indicate, first, that the OpenAI GPT model is more sensitive than BERT to distracting nouns ("attractors").

Another interesting pattern is the similarity of the OpenAI GPT results and the LSTM results. This may indicate that, more than the self-attention architecture, it could well be the training objective (LM vs. Masked-LM) or the resultant bi-directionality of the BERT architecture that provide the basis of BERT's strong results on the sentences that contains distractors [5].

	prefix + suffix			only prefix			
	BERT†	BERT†	OpenAI	BERT	BERT	OpenAI	#
Attractors	Base	Large	GPT	Base	Large	GPT	sents
1	0.97	0.97	0.82	0.82	0.78	0.82	24031
2	0.97	0.97	0.76	0.82	0.78	0.77	4414
3	0.96	0.96	0.69	0.82	0.76	0.71	946
4	0.97	0.96	0.70	0.82	0.78	0.70	254

Table 1: Results on the Linzen et al. [2016] stimuli. 'Only prefix': BERT is only provided with the pre-fix of the stimuli up the the (masked) target word. † are the results reported in Yoav Goldberg's tech report "Assessing BERT's Syntactic Abilities".

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, October 2018. URL `http://arxiv.org/abs/1810.04805`. arXiv: 1810.04805.

[5] Two other less likely options would be (i) the use of wikipedia as additional training data and (ii) the use of WordPieces sub-unit instead of BPE.

	prefix + suffix			only prefix			
	BERT†	BERT†	OpenAI	BERT	BERT	OpenAI	#
Attractors	Base	Large	GPT	Base	Large	GPT	sents
all	0.83	0.80	0.68	0.54	0.55	0.73	383
0	0.84	0.80	0.68	0.49	0.50	0.74	311
1	0.81	0.75	0.71	0.75	0.75	0.75	63
2	0.89	0.89	0.44	0.89	0.89	0.44	9

Table 2: Results on the EN NONCE Gulordava et al. [2018] stimuli. ’Only prefix’: BERT is only provided with the pre-fix of the stimuli up the the (masked) target word. † are the results reported in Yoav Goldberg’s tech report ”Assessing BERT’s Syntactic Abilities”.

	prefix + suffix			only prefix					
	BERT†	BERT†	OpenAI	BERT	BERT	OpenAI	LSTM	Humans	# Pairs
	Base	Large	GPT	Base	Large	GPT	(M&L)	(M&L)	(# M&L Pairs)
SUBJECT-VERB AGREMENT:									
Simple	**1.00**	**1.00**	0.94	0.83	0.83	0.96	0.94	0.96	120 (140)
In a sentential complement	0.83	0.86	**0.90**	0.84	0.62	0.88	0.99	0.93	1440 (1680)
Short VP coordination	0.89	0.86	**0.99**	0.94	0.85	**0.99**	0.90	0.82	720 (840)
Long VP coordination	**0.98**	0.97	0.91	0.74	0.63	0.93	0.61	0.82	400 (400)
Across a prepositional phrase	**0.85**	**0.85**	0.75	0.78	0.70	0.76	0.57	0.85	19440 (22400)
Across a subject relative clause	0.84	**0.85**	0.69	0.83	0.82	0.73	0.56	0.88	9600 (11200)
Across an object relative clause	**0.89**	0.85	0.75	0.85	0.73	0.73	0.50	0.85	19680 (22400)
Across an object relative (no *that*)	**0.86**	0.81	0.73	0.83	0.71	0.70	0.52	0.82	19680 (22400)
In an object relative clause	0.95	**0.99**	0.88	0.61	0.69	0.88	0.84	0.78	15960 (22400)
In an object relative (no *that*)	0.79	0.82	**0.86**	0.53	0.50	0.85	0.71	0.79	15960 (22400)
REFLEXIVE ANAPHORA:									
Simple	0.94	0.92	**1.00**	0.92	0.86	**1.00**	0.83	0.96	280 (280)
In a sentential complement	0.89	0.86	**0.94**	0.78	0.68	**0.94**	0.86	0.91	3360 (3360)
Across a relative clause	0.80	0.76	0.66	**0.92**	0.82	0.66	0.55	0. 87	22400 (22400)

Table 3: Results on the Marvin and Linzen [2018] stimuli. In the ”only prefix” setting, BERT is only provided with the prefix of the stimuli and the (masked) target word (see text). † are the results reported in Yoav Goldberg’s tech report ”Assessing BERT’s Syntactic Abilities”.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. page 12, 2018.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *arXiv:1506.06724 [cs]*, June 2015. URL `http://arxiv.org/abs/1506.06724`. arXiv: 1506.06724.

Rebecca Marvin and Tal Linzen. Targeted Syntactic Evaluation of Language Models. *arXiv:1808.09031 [cs]*, August 2018. URL `http://arxiv.org/abs/1808.09031`. arXiv: 1808.09031.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *arXiv:1611.01368 [cs]*, November 2016. URL `http://arxiv.org/abs/1611.01368`. arXiv: 1611.01368.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. *arXiv:1803.11138 [cs]*, March 2018. URL `http://arxiv.org/abs/1803.11138`. arXiv: 1803.11138.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. pages 1715–1725. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1162. URL `http://aclweb.org/anthology/P16-1162`.

Alexei Baevski and Michael Auli. Adaptive Input Representations for Neural Language Modeling. *arXiv:1809.10853 [cs]*, September 2018. URL `http://arxiv.org/abs/1809.10853`. arXiv: 1809.10853.