Learning from Mistakes: Using Mis-predictions as Harm Alerts in Language Pre-Training

Chen Xing, Wenhao Liu and Caiming Xiong
Salesforce Research, Palo Alto, USA

Motivation

`He went back to his bedroom to continue his draft design of a new bed.`

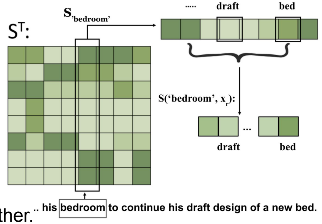
- The ground-truth word is ``study", which is mis-predicted as ``bedroom". It is probably because **the frequently co-occuring pattern** between "bedroom" and "bed" that is easy to fit for the model, **dominates pre-training and outruns the hard-to-fit semantics in the context.**
- We believe that **mis-predictions can help locate such dominating patterns** the model has fitted that harm language understanding. **When a mis-prediction occurs, there are likely to be some dominating patterns related to the mis-prediction in the context** fitted by the model that cause this mis-prediction, for example, the frequently co-occurring word "bed" with the mis-prediction "bedroom".
- **If we can add regularization to train the model to rely less on these dominating patterns such as word co-occurrences when a mis-prediction occurs, thus focusing more on the rest more subtle patterns, more information can be efficiently fitted at pre-training.**

Method:Using Mis-predictions as Harm Alerts (MPA)

- **Building a context matrix S with word co-occurrence information**

$$\mathbf{C}_{i,j}^{normed} = \frac{\mathbf{C}_{i,j}}{(\sum_{z=1}^{V}\mathbf{C}_{i,z})\cdot(\sum_{z=1}^{V}\mathbf{C}_{j,z})},$$

$$\mathbf{S}_i = \frac{\mathbf{C}_i^{normed} - \min(\mathbf{C}_i^{normed})}{\max(\mathbf{C}_i^{normed}) - \min(\mathbf{C}_i^{normed})}.$$



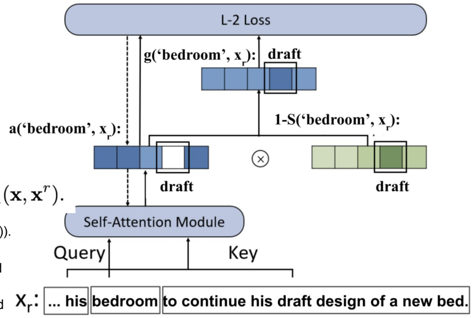
Ci,j records the total number of times that token wi and wj occur together.

- **Pre-Training in MPA**

$$\mathbf{g}(\mathbf{q}_t,\mathbf{K}) = \frac{\mathbf{q}_t\mathbf{K}^T}{\sqrt{d_K}}\cdot(1-\mathbf{S}(t,\mathbf{x}^r)).$$

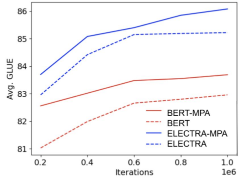
$$\mathcal{L}_A = \frac{1}{N_M}\sum_{t=0}^{N_M}\left(\frac{\mathbf{q}_t\mathbf{K}^T}{\sqrt{d}} - \mathbf{g}(\mathbf{q}_t,\mathbf{K})\right)^2$$

$$\mathcal{L}(\mathbf{x},\mathbf{x}^m,\mathbf{x}^r) = \mathcal{L}_G(\mathbf{x},\mathbf{x}^m) + \lambda\mathcal{L}_D(\mathbf{x},\mathbf{x}^r) + \gamma\mathcal{L}_A(\mathbf{x},\mathbf{x}^r).$$



We multiply the original pre-softmax attention co-efficients with (1-S(t,x^r)). Through this way, keys ignored by the attention module could be set a larger weight if their context coefficients in S(t, x^r) are smaller compared with other tokens in the sentence.
Similarly, keys at positions of frequent context of the mis-prediction would be set smaller weights.

Experiment Results



	Params	Avg. GLUE	Avg. SuperGLUE	SQuAD2.0 F1	SQuAD2.0 EM
GPT-2	117 M	78.8	-	-	-
BERT	110 M	82.2	66.1	76.4	79.6
SpanBERT	110 M	83.9	-	77.1	80.3
ELECTRA	110 M	85.1	-	80.5	83.3
BERT (Ours)	110 M	83.0	66.3	76.9	80.1
BERT-MPA	110 M	**83.7**	**67.4**	**77.5**	**80.7**
ELECTRA (Ours)	110 M	85.2	70.1	80.2	83.1
ELECTRA-MPA	110 M	**86.0**	**72.2**	**83.1**	**86.1**

	MNLI	QNLI	QQP	SST	CoLA	MRPC	RTE	STS
BERT (Ours)	84.9±0.09	**91.3**±0.17	91.0±0.07	92.9±0.15	55.2±0.63	88.3±0.94	68.6±0.74	**89.4**±0.42
BERT-MPA	**84.9**±0.11	90.9±0.27	**91.1**±0.16	**93.4**±0.09	**61.2**±0.51	**89.7**±0.88	**70.6**±0.81	87.8±0.59
ELECTRA(Ours)	86.9±0.07	92.5±0.11	91.6±0.17	93.0±0.08	**67.6**±0.28	90.3±0.59	70.1±0.98	90.0±0.36
ELECTRA-MPA	**87.1**±0.13	**92.8**±0.35	**91.7**±0.19	**93.8**±0.07	67.4±0.24	**91.8**±0.41	**73.2**±0.56	**90.2**±0.29