Inexpensive Domain Adaptation of Pretrained Language Models (Appendix)

Word2Vec training

We downloaded the PubMed, PMC and CORD-19 corpora from:

- https://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_bulk/ [20 January 2020, 68GB raw text]
- https://ftp.ncbi.nlm.nih.gov/pubmed/baseline/ [20 January 2020, 24GB raw text]
- https://pages.semanticscholar.org/coronavirus-research [17 April 2020, 2GB raw text]

We extract all abstracts and text bodies and apply the BERT basic tokenizer (a rule-based word tokenizer that standard BERT uses before wordpiece tokenization). Then, we train CBOW Word2Vec[5] with negative sampling. We use default parameters except for the vector size (which we set to $d_{\mathrm{W2V}} = d_{\mathrm{LM}}$).

Experiment 1: Biomedical NER

Pretrained models

General-domain BERT and BioBERTv1.0 were downloaded from:

- www.storage.googleapis.com/bert_models/2018_10_18/cased_L-12_H-768_A-12.zip
- www.github.com/naver/biobert-pretrained

Data

We downloaded the NER datasets by following instructions on www.github.com/dmis-lab/biobert#Datasets. For detailed dataset statistics, see Lee et al. (2020).

Preprocessing

We use Lee et al. (2020)'s preprocessing strategy: We cut all sentences into chunks of 30 or fewer whitespace-tokenized words (without splitting inside labeled spans). Then, we tokenize every chunk S with $\mathcal{T} = \mathcal{T}_{\mathrm{LM}}$ or $\mathcal{T} = \hat{\mathcal{T}}_{\mathrm{LM}}$ and add special tokens:

$$X = \textit{[CLS]}\ \mathcal{T}(S)\ \textit{[SEP]}$$

Word-initial wordpieces in $\mathcal{T}(S)$ are labeled as *B(egin)*, *I(nside)* or *O(utside)*, while non-word-initial wordpieces are labeled as *X(ignore)*.

Modeling, training and inference

We follow Lee et al. (2020)'s implementation (www.github.com/dmis-lab/biobert): We add a randomly initialized softmax classifier on top of the last BERT layer to predict the labels. We finetune the entire model to minimize negative log likelihood, with the AdamW optimizer (Loshchilov and Hutter, 2018) and a linear learning rate scheduler (10% warmup). All finetuning runs were done on a GeForce Titan X GPU (12GB).

At inference time, we gather the output logits of word-initial wordpieces only. Since the number of word-initial wordpieces is the same for $\mathcal{T}_{\mathrm{LM}}(S)$ and $\hat{\mathcal{T}}_{\mathrm{LM}}(S)$, this makes mean-pooling the logits straightforward.

Hyperparameters

We tune the batch size and peak learning rate on the development set (metric: F1), using the same hyperparameter space as Lee et al. (2020):

Batch size: $[10, 16, 32, 64]$[6]

Learning rate: $[1 \cdot 10^{-5}, 3 \cdot 10^{-5}, 5 \cdot 10^{-5}]$

We train for 100 epochs, which is the upper end of the 50–100 range recommended by the original authors. After selecting the best configuration for every task and model (see Table 7), we train the final model on the concatenation of training and development set, as was done by Lee et al. (2020). See Figure 2 for expected maximum development set F1 as a function of the number of evaluated hyperparameter configurations (Dodge et al., 2019).

Experiment 2: Covid-19 QA

Pretrained model

We downloaded the SQuADBERT baseline from:

- www.huggingface.co/bert-large-uncased-whole-word-masking-finetuned-squad

Data

We downloaded the Deepset-AI Covid-QA dataset from:

- www.github.com/deepset-ai/COVID-QA/blob/master/data/question-answering/COVID-QA.json [24 June 2020]

[5] www.github.com/tmikolov/word2vec

[6] Since LINNAEUS and BC4CHEM have longer maximum tokenized chunk lengths than the other datasets, our hardware was insufficient to evaluate batch size 64 on them.

At the time of writing, the dataset contains 2019 questions and gold answer spans. Every question is associated with one of 147 research papers (contexts) from CORD-19.[7] Since we do not do target-domain finetuning, we treat the entire dataset as a test set.

Preprocessing

We tokenize every question-context pair (Q, C) with $\mathcal{T} = \mathcal{T}_{\text{LM}}$ or $\mathcal{T} = \hat{\mathcal{T}}_{\text{LM}}$, which yields $(\mathcal{T}(Q), \mathcal{T}(C))$. Since $\mathcal{T}(C)$ is usually too long to be digested in a single forward pass, we define a sliding window with width and stride $N = \text{floor}(\frac{509-|\mathcal{T}(Q)|}{2})$. At step n, the "active" window is between $a_n^{(l)} = (n-1)N + 1$ and $a_n^{(r)} = \min(|C|, nN)$. The input is defined as:

$$X^{(n)} = \textit{[CLS]}\ \mathcal{T}(Q)\ \textit{[SEP]}$$
$$\mathcal{T}(C)_{a_n^{(l)}-p_n^{(l)}:a_n^{(r)}+p_n^{(r)}}\ \textit{[SEP]}$$

$p_n^{(l)}$ and $p_n^{(r)}$ are chosen such that $|X^{(n)}| = 512$, and such that the active window is in the center of the input (if possible).

Modeling and inference

Feeding $X^{(n)}$ into the QA model yields start logits $\mathbf{h}'^{(\text{start},n)} \in \mathbb{R}^{|X^{(n)}|}$ and end logits $\mathbf{h}'^{(\text{end},n)} \in \mathbb{R}^{|X^{(n)}|}$. We extract and concatenate the slices that correspond to the active windows of all steps:

$$\mathbf{h}^{(*)} \in \mathbb{R}^{|\mathcal{T}(C)|}$$
$$\mathbf{h}^{(*)} = [\mathbf{h}'^{(*,1)}_{a_1^{(l)}:a_1^{(r)}}; \ldots; \mathbf{h}'^{(*,n)}_{a_n^{(l)}:a_n^{(r)}}; \ldots]$$

Next, we map the logits from the wordpiece level to the word level. This allows us to mean-pool the outputs of $\mathcal{T}_{\text{LM}}$ and $\hat{\mathcal{T}}_{\text{LM}}$ even when $|\mathcal{T}_{\text{LM}}(C)| \neq |\hat{\mathcal{T}}_{\text{LM}}(C)|$.

Let c_i be a word in C and let $\mathcal{T}(C)_{j:j+|\mathcal{T}(c_i)|}$ be the corresponding wordpieces. The start and end logits of c_i are:

$$o_i^{(*)} = \max_{j \leq j' \leq j+|\mathcal{T}(c_i)|}[h_{j'}^{(*)}]$$

Finally, we return the answer span $C_{k:k'}$ that maximizes $o_k^{(\text{start})} + o_{k'}^{(\text{end})}$, subject to the constraints that k' does not precede k and the answer contains no more than 500 characters.

Notes on Covid-QA

There are some important differences between Covid-QA and SQuAD, which make the task challenging:

- The Covid-QA contexts are full documents rather than single paragraphs. Thus, the correct answer may appear several times, often with slightly different wordings. But only a single occurrence is annotated as correct, e.g.:

  Question: What was the prevalence of Coronavirus OC43 in community samples in Ilorin, Nigeria?

  Correct: 13.3% (95% CI 6.9-23.6%) *# from main text*

  Predicted: 13.3%, 10/75 *# from abstract*

- SQuAD gold answers are defined as the "shortest span in the paragraph that answered the question" (Rajpurkar et al., 2016, p. 4), but many Covid-QA gold answers are longer and contain non-essential context, e.g.:

  Question: When was the Middle East Respiratory Syndrome Coronavirus isolated first?

  Correct: (MERS-CoV) was first isolated in 2012, in a 60-year-old man who died in Jeddah, KSA due to severe acute pneumonia and multiple organ failure

  Predicted: 2012

These differences are part of the reason why the exact match score is lower than the word-level F1 score and the substring score (see Table 6, bottom, main paper).

[7] www.github.com/deepset-ai/COVID-QA/issues/103

		BERT (repro)		BioBERTv1.0 (repro)		GreenBioBERT	
Biomedical NER task	(ID)	hyperparams	dev set F1	hyperparams	dev set F1	hyperparams	dev set F1
BC5CDR-disease	(1)	$32, 3 \cdot 10^{-5}$	82.12	$10, 1 \cdot 10^{-5}$	85.15	$32, 1 \cdot 10^{-5}$	83.90
NCBI-disease	(2)	$32, 3 \cdot 10^{-5}$	87.52	$32, 1 \cdot 10^{-5}$	87.99	$10, 3 \cdot 10^{-5}$	88.43
BC5CDR-chem	(3)	$64, 3 \cdot 10^{-5}$	91.00	$32, 1 \cdot 10^{-5}$	93.36	$10, 1 \cdot 10^{-5}$	92.59
BC4CHEMD	(4)	$16, 1 \cdot 10^{-5}$	88.02	$32, 1 \cdot 10^{-5}$	89.35	$16, 1 \cdot 10^{-5}$	88.53
BC2GM	(5)	$32, 1 \cdot 10^{-5}$	83.91	$64, 3 \cdot 10^{-5}$	85.54	$64, 3 \cdot 10^{-5}$	84.25
JNLPBA	(6)	$32, 5 \cdot 10^{-5}$	85.18	$32, 5 \cdot 10^{-5}$	85.30	$10, 3 \cdot 10^{-5}$	85.10
LINNAEUS	(7)	$16, 1 \cdot 10^{-5}$	96.67	$32, 1 \cdot 10^{-5}$	97.22	$10, 1 \cdot 10^{-5}$	96.49
Species-800	(8)	$32, 1 \cdot 10^{-5}$	72.70	$32, 1 \cdot 10^{-5}$	77.34	$16, 1 \cdot 10^{-5}$	75.93

Table 7: Best hyperparameters (batch size, peak learning rate) and best dev set F1 per NER task and model. BERT (repro) and BioBERTv1.0 (repro) refer to our reproduction experiments.

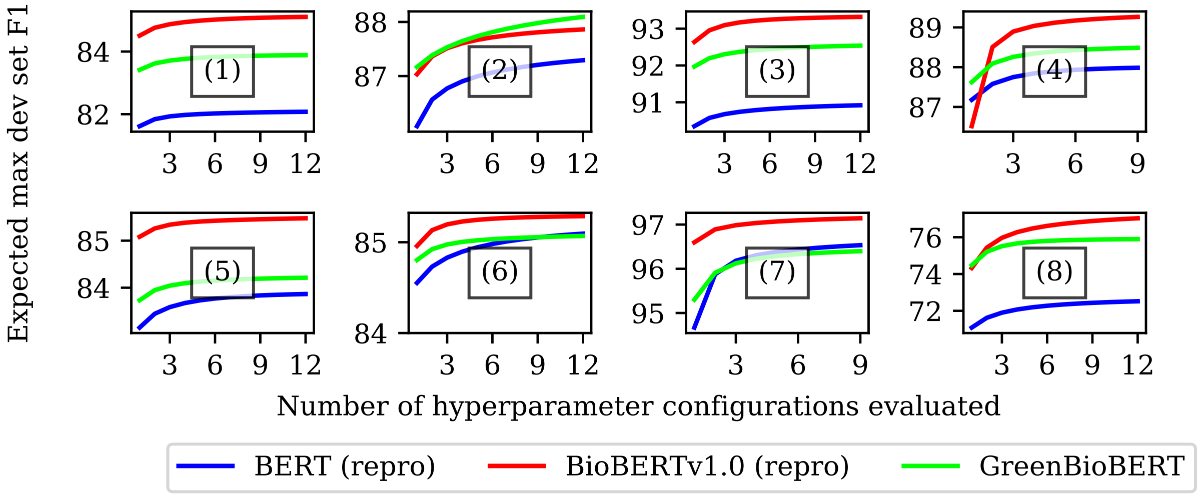

Figure 2: Expected maximum F1 on NER development sets as a function of the number of evaluated hyperparameter configurations. Numbers in brackets are NER task IDs (see Table 7).