Mirror-BERT: Transforming Pretrained Language Models into Universal Text Encoders

Fangyu Liu, Ivan Vulić, Anna Korhonen, Nigel Collier

Language Technology Lab, University of Cambridge, UK

1 Motivation

- Off-the-shelf pretrained language models (PLMs) such as BERT/RoBERTa are not effective universal text encoders.
- Downstream task data (e.g. NLI, paraphrasing, sentence similarity) are needed for finetuning a good universal text encoder.

Model	Avg.
Avg. GloVe embeddings	61.32
Avg. BERT embeddings	54.81
BERT CLS-vector	29.19
InferSent - Glove	65.01
Universal Sentence Encoder	71.22
SBERT-NLI-base	74.89
SBERT-NLI-large	76.55
SRoBERTa-NLI-base	74.21
SRoBERTa-NLI-large	**76.68**

off-the-shelf BERT

text encoders tuned on task data (e.g. NLI, paraphrase)

Figure 1: Table from (Reimers & Gurevych, 2019): on the task of Semantic Textual Similarity, off-the-shelf BERT performs worse than text encoders tuned on task data and even GloVe word embeddings.

RQ: How do we close the gap *without labelled data*?

In this work we propose *Mirror-BERT*, which can transform a given PLM into a powerful word, phrase, or sentence encoder, usually matching the performance of supervised encoders.

2 Method: Self-Supervised Learning

- Step 1: given a randomly sampled sequence x_i (e.g. a raw sentence from Wikipedia), we replicate it and get an identical string $\overline{x}_i$.
- Step 2 (optional): randomly replace a span of certain length in $\overline{x}_i$ with `[MASK]`.
- Step 3: send x_i and $\overline{x}_i$ to the same PLM separately and get their representations $f(x_i)$ and $f(\overline{x}_i)$.
- Step 4: Leverage the infoNCE loss (Eq. (1)) to pull $f(x_i)$ and $f(\overline{x}_i)$ together with respect to other features in the mini-batch (i.e. $f(x_j)$ and $f(\overline{x}_j)$ where $j \neq i$).

$$\mathcal{L}_b = -\sum_{i=1}^{|\mathcal{D}_b|} \log \frac{\exp(\cos(f(x_i), f(\overline{x}_i))/\tau)}{\sum_{x_i \in \mathcal{N}_i} \exp(\cos(f(x_i), f(x_j))/\tau)}. \quad (1)$$

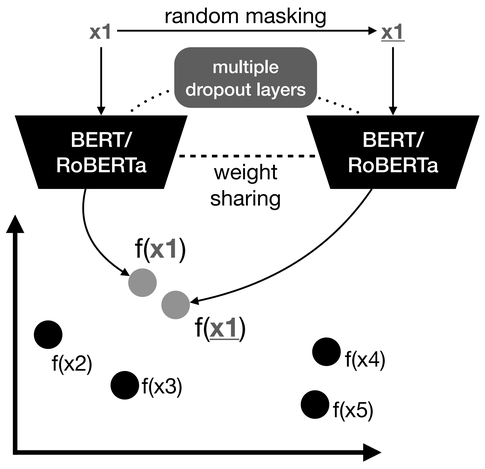

Figure 2: The same text sequence can be observed from two additional "views": 1) by performing random masking in the input space, and/or 2) by applying dropout (inside the BERT/RoBERTa PLM) in the feature space, yielding identity-based (i.e., "mirrored") positive examples for contrastive-fine-tuning.

Intuition: The random span masking and dropout layers inside BERT/RoBERTa serve as data augmentations. Essentially, we inject two inductive biases: (i) masking parts of an input sentence, humans can usually reconstruct its semantics, then so should the models; (ii) dropping a small subset of neurons or representation dimensions, the embeddings should not drift too much.

3 Experiments

Lexical-level Tasks:

lang.→	EN	FR	ET	AR	ZH	RU	ES	PL	avg.
fastText	.434	.560	**.447**	.409	.428	**.435**	**.488**	.396	.450
BERT	.267	.020	.106	.220	.398	.202	.177	.217	.201
+ Mirror	**.556**	**.621**	.308	**.538**	**.639**	.365	.296	**.444**	**.471**

Table 1: Word similarity evaluation on Multi-SimLex.

Sentence-level Tasks:

dataset→	STS12	STS13	STS14	STS15	STS16	STS-b	SICK-R	avg.
SBERT	**.719**	.774	**.742**	.799	.747	.774	**.721**	.754
RoBERTa⋆	.134	.126	.124	.203	.224	.129	.320	.180
+ Mirror	.646	**.818**	.734	**.802**	**.782**	**.787**	.703	.753

Table 2: English Semantic Textual Similarity benchmark results.

4 Discussions

Observation: more data don't help.

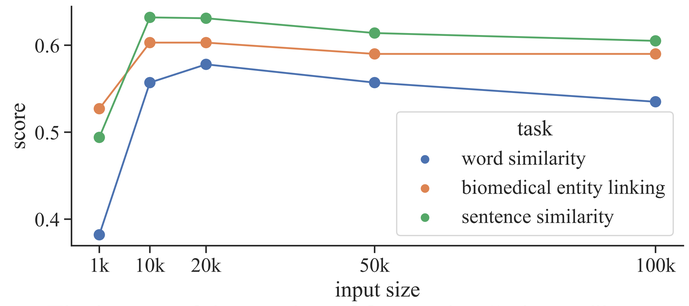

Figure 3: The impact of the number of fine-tuning "mirrored" examples (x-axis) on the task performance (y-axis).

Learning new knowledge or exposing available knowledge? Seems to be the latter.

model	ρ
fastText	.434
BERT	.267
+ Mirror	.556
+ Mirror (random string, lr `5e-5`)	.481

Table 3: Running Mirror-BERT with a set of 'zero-semantics' random strings. Evaluation is conducted on Multi-SimLex (EN).