Paper Reading: No Training Required: Exploring Random Encoders For Sentence Classification

Anonymous Authors

ICLR 2019 score: 7, 5, 8

Motivation

- Explore various methods for computing sentence representations from pretrained word embeddings without any training
- look at how much sentence embeddings gain over random methods
- provide a field with more appropriate baselines going forward

Methods for Embedding

- Word embedding: Skip-gram, CBOW, GloVe, ELMO, BERT
- Sentence embeddings:
  - Supervised Learning: Task-specific representation, Such as InferSent, Deep average Network, ...
  - Unsupervised Learning: Such as Skip-thought vector, Average BOW, Weighted BOW, ...
  - Auto-encoder: predict itself. Such as Recursive Auto-encoder, ...
  - Multi-task learning: Generalization for the related supervised/unsupervised methods. Such as Universal sentence encoder, ...
- Phrase/Paragraph/Document embeddings

Approach

Random Sentence Encoders: Produce sentence embeddings from pre-trained word embeddings without requiring any training of the encoder itself.

Sentence representation h is computed using some function f parameterized by θ over pre-trained input word embeddings $e \in L$: $h = f_\theta(e_1, ..., e_n)$

1. Bag of Random Embedding Projections(BOREP)
2. Random LSTMs
3. Echo State Networks

Bag of Random Embedding Projections(BOREP)

$h = f_{pool}(We_i)$,

randomly initialize matrix $W \in R^{D*d}$, the value for the matrix are sampled uniformly from $[-d^{-0.5}, d^{-0.5}]$.

f_{pool} is some pooling function, e.g. $f_{pool}(x) = \sum(x)$,
$f_{pool}(x) = max(x)$...

Random LSTMs

$h = f_{pool}(BiLSTM(e_1, ..., e_n))$.

The LSTM weight matrices and their corresponding biases are initialized uniformly at random from $[-d^{-0.5}, d^{-0.5}]$, d is the hidden size of the LSTM. The architecture here is the same as that of InferSent model.

Echo State Networks

Echo State Networks were designed for sequence prediction problems, where given a sequence X, predict a label y for each step in the sequence.

$$\tilde{\mathbf{h}}_i = f_{pool}(W^i \mathbf{e}_i + W^h \mathbf{h}_{i-1} + b^i)$$
$$\mathbf{h}_i = (1-\alpha)\mathbf{h}_{i-1} + \alpha \tilde{\mathbf{h}}_i,$$

where W^i, W^r and b^i are randomly initialized and are not updated during training.

$$\hat{y}_i = W^o[\mathbf{e}_i; \mathbf{h}_i] + b^o.$$

Only W^o and b^o are learnable.

$$\mathbf{h} = \max(\mathrm{ESN}(\mathbf{e}_1, \ldots, \mathbf{e}_n)).$$

Results: Sentence Classification Tasks

Model	Dim	MR	CR	MPQA	SUBJ	SST2	TREC	SICK-R	SICK-E	MRPC	STSB
BOE	300	77.3	78.9	87.7	91.0	79.7	83.0	80.4	78.6	72.9	70.7
BOREP	4096	77.3	79.0	88.4	92.0	81.2	88.0	85.5	81.6	73.2	68.4
RandLSTM	4096	77.2	78.8	87.9	91.9	82.0	86.9	85.5	81.9	73.7	72.5
ESN	4096	78.4	80.3	88.6	92.4	83.5	88.8	86.1	82.3	73.7	74.6
InferSent-1 = paper version, glove	4096	81.1	86.3	90.2	92.4	84.6	88.2	88.3	86.3	76.2	75.6
InferSent-2 = fixed padding, fasttext	4096	79.7	84.2	89.4	92.7	84.3	90.8	88.8	86.3	76.0	78.4
InferSent-3 = fixed padding, glove	4096	79.7	83.4	88.9	92.6	83.5	90.8	88.5	84.1	76.4	77.3
Δ InferSent-3, BestRand	-	*1.3*	*3.1*	*0.3*	*0.2*	*0.0*	*1.0*	*2.4*	*1.8*	*2.7*	*2.7*
ST-LN	4800	79.4	83.1	89.3	93.7	82.9	88.4	85.8	79.5	73.2	68.9
Δ ST-LN, BestRand	-	*1.0*	*2.8*	*1.3*	*1.3*	*-0.6*	*-0.4*	*-0.3*	*2.8*	*-0.5*	*-5.7*

Table 1: Performance (accuracy for all tasks except SICK-R and STSB, for which we report Pearson's r). All models have 4096 dimensions with the exception of BOE (300) and ST-LN (4800). The last two rows show the performance difference between InferSent-3 and the best performing random architecture for each task. The average performance difference between the best random architecture and InferSent-3 and ST-LN is 1.6 and 0.2 respectively.

Results: Taking Cover To The Max

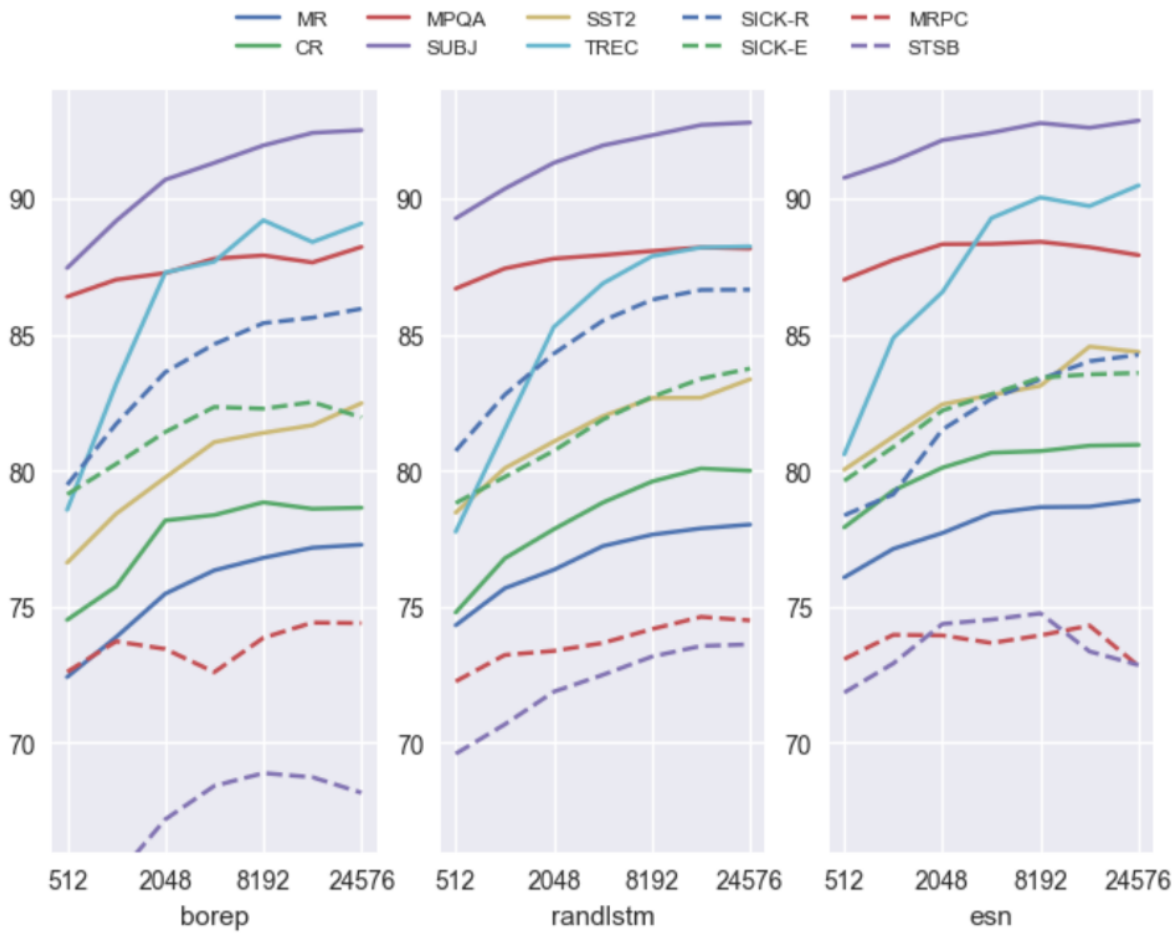

Figure 1: Performance while varying dimensionality, for the three random sentence encoders over the 10 downstream tasks.

Results: Taking Cover To The Max

Model	MR	CR	MPQA	SUBJ	SST2	TREC	SICK-R	SICK-E	MRPC	STSB
BOE	77.3	78.9	87.7	91.0	79.7	83.0	80.4	78.6	72.9	70.7
BOREP	78.6	80.0	88.7	92.9	82.5	89.4	86.0	84.1	73.9	68.2
RandLSTM	78.0	80.0	88.2	92.8	83.4	88.2	86.6	83.7	74.5	73.6
ESN	78.5	80.2	88.9	93.1	84.2	92.1	87.1	85.0	74.8	72.9
InferSent-3 4096×6	79.7	83.9	89.1	92.8	82.4	90.6	79.5	85.9	75.1	75.0
ST-LN 4096×6	75.2	80.8	86.8	92.7	80.6	88.4	82.9	81.3	71.5	67.0

Table 2: Performance (accuracy for all tasks except SICK-R and STSB, for which we report Pearson's r). All models have 4096×6 dimensions. ST-LN and InferSent-3 were projected to this dimension with a random projection.

Results: Probing Tasks

Model	SentLen	WC	TreeDepth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
BOE (300d, class.)	60.5	87.5	32.0	62.7	50.0	83.7	78.0	76.6	50.5	53.8
BOREP (4096d, class.)	64.4	97.1	33.0	71.3	49.8	86.3	81.5	79.3	49.5	54.1
RandLSTM (4096d, class.)	72.8	94.1	35.6	76.2	55.2	86.6	84.0	79.5	49.7	63.1
ESN (4096d, class.)	78.8	92.4	36.9	76.2	62.9	86.6	82.3	79.7	49.7	60.3
Infersent-3	80.6	93.5	37.1	78.2	57.3	86.8	84.8	80.5	53.0	65.8
ST-LN	79.9	79.9	39.5	82.1	69.4	90.2	86.2	83.4	54.5	68.9

Table 3: Performance on a set of probing tasks as defined in (Conneau et al., 2018). All random architecture models are 4096 dimensions and were selected by tuning over validation performance on the classification tasks.

Discussion and Conclusion

- InferSent: requires large amounts of expensive annotation.
  SkipThought: takes a very long time to train.
  Random Sentence Encoders: easy, fast and not bad.
- More dimensions in the encoder is usually better.
- How much training do we need? Different in order task and complex task(linguistic information).
- Why random non-linear features are able to perform well on these tasks?