Sentence-BERT: Sentence Embeddings using Siamese BERT-Network

Nils Reimers and Iryna Gurevych

2019

http://github.com/UKPLab/sentence-transformers

임동주

목차

Introduction

- 기존 BERT를 활용한 STS/NLI
  - Cross-Encoder
    - 많은 양의 문장들의 유사도를 비교할 때 시간이 많이 소요됨
    - Cross Attention 연산 때문

    Ex) 10,000개 문장의 유사도를 각각 비교

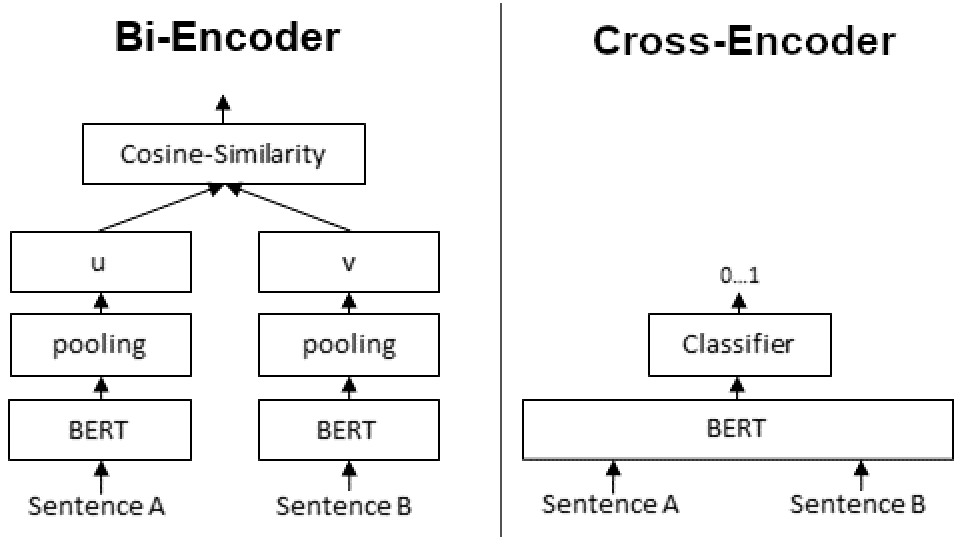
    > $_{10000}C_2$ =49,995,000회의 연산 필요

    > V100 GPU 기준 65시간 소요

- SBERT를 활용한 STS/NLI
  - Bi-Encoder
    - Cross-Encoder에 비해 짧은 시간 소요

    Ex) 10,000개 문장의 유사도를 각각 비교

    > 10,000번의 임베딩 + 코사인 유사도



Model

- Pooling Strategy
  - 고정된 크기의 벡터를 생성하기 위해 BERT/RoBERTa의 output 에 pooling 연산을 추가함
    1. CLS-token의 output 사용
    2. 모든 output vector의 평균 산출 (Mean pooling)
    3. 모든 output vector중 최대값 산출 (Max pooling)

- 적용 가능한 Task 예시
  - NLI
  - STS (Supervised/Unsupervised)

Model

- Classification
  - Objective Function
    - 임베딩된 벡터인 u와 v, 그 둘의 element-wise 차인 $|u-v|$를 훈련된 가중치인 W_t를 각각 곱한 값에 softmax를 취함

$$\begin{cases} o = softmax(W_t(u, v, |u - v|)) \\ W_t \in \mathbb{R}^{3n \times k} \end{cases}$$

  - Loss Function
    - Cross-entropy Loss

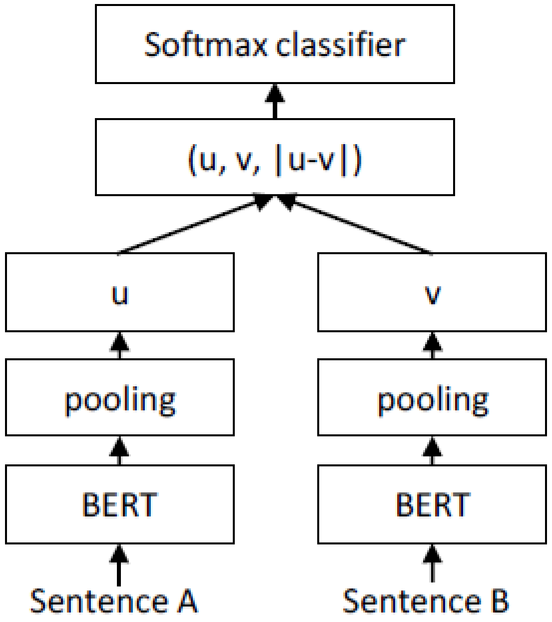


Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

Model

- Regression
  - Objective Function
    - 임베딩된 두 벡터인 *u*와 *v*의 코사인 유사도를 직접 산출
  - Loss Function
    - Mean Squared Error Loss

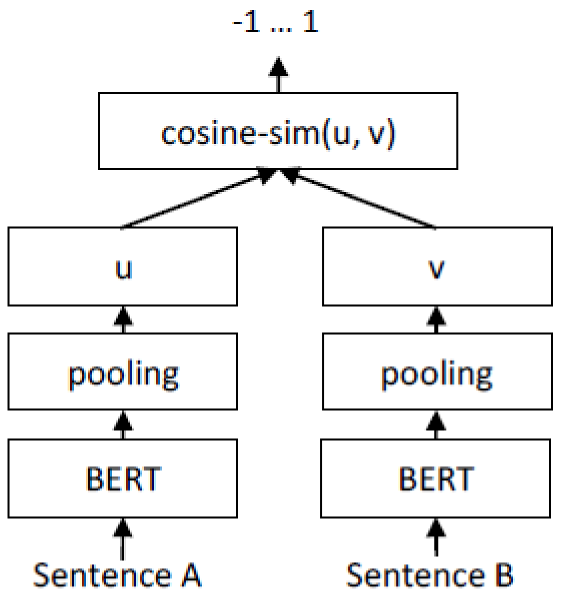
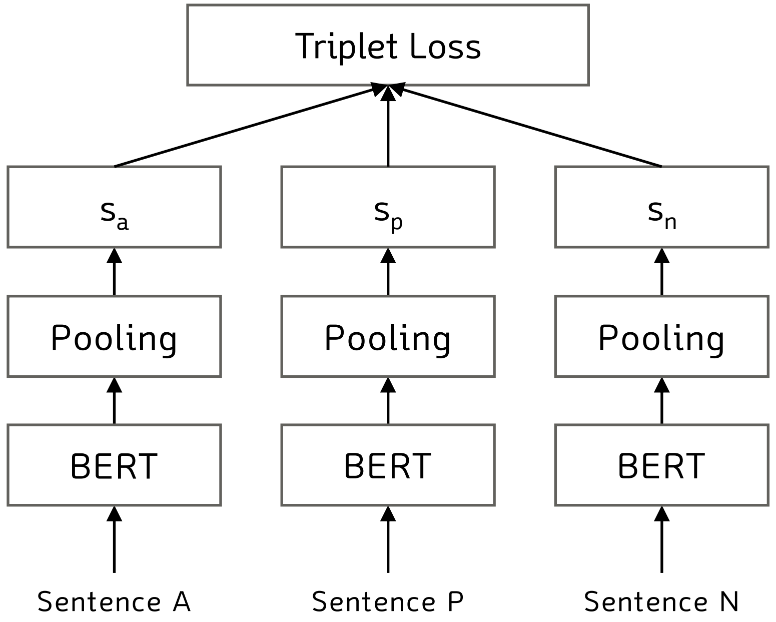


Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

Model

- Triplet Network
  - Loss Function
    - s_x : 문장 x에 대한 임베딩 결과
    - ϵ : margin

$$max(\| s_a - s_p \| - \| s_a - s_n \| + \epsilon, 0)$$



Evaluation-STS

- Unsupervised STS

Model	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg.
Avg. GloVe embeddings	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
Avg. BERT embeddings	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
BERT CLS-vector	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
InferSent - Glove	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder	64.49	67.80	64.61	76.83	73.18	74.92	**76.69**	71.22
SBERT-NLI-base	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT-NLI-large	72.27	**78.46**	**74.90**	80.99	76.25	**79.23**	73.75	76.55
SRoBERTa-NLI-base	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa-NLI-large	**74.53**	77.00	73.18	**81.85**	**76.82**	79.10	74.29	**76.68**

Table 1: Spearman rank correlation ρ between the cosine similarity of sentence representations and the gold labels for various Textual Similarity (STS) tasks. Performance is reported by convention as $\rho \times 100$. STS12-STS16: SemEval 2012-2016, STSb: STSbenchmark, SICK-R: SICK relatedness dataset.

- Pretrain Dataset
  - Wikipedia
  - NLI Dataset
- Test Dataset
  - STS tasks 2012-2016
  - STS benchmark
  - SICK-Relatedness dataset
- Objective
  - Regression (Cosine Similarity)

Evaluation-STS

- Supervised STS
  - Pretrain Dataset
    - STS benchmark (train: 5,749 / valid: 1,500)
  - Test Dataset
    - STS benchmark (test: 1,379)
  - Objective
    - Regression (Cosine Similarity)

Model	Spearman
Not trained for STS	
Avg. GloVe embeddings	58.02
Avg. BERT embeddings	46.35
InferSent - GloVe	68.03
Universal Sentence Encoder	74.92
SBERT-NLI-base	77.03
SBERT-NLI-large	79.23
Trained on STS benchmark dataset	
BERT-STSb-base	84.30 ± 0.76
SBERT-STSb-base	84.67 ± 0.19
SRoBERTa-STSb-base	**84.92** ± 0.34
BERT-STSb-large	**85.64** ± 0.81
SBERT-STSb-large	84.45 ± 0.43
SRoBERTa-STSb-large	85.02 ± 0.76
Trained on NLI data + STS benchmark data	
BERT-NLI-STSb-base	**88.33** ± 0.19
SBERT-NLI-STSb-base	85.35 ± 0.17
SRoBERTa-NLI-STSb-base	84.79 ± 0.38
BERT-NLI-STSb-large	**88.77** ± 0.46
SBERT-NLI-STSb-large	86.10 ± 0.13
SRoBERTa-NLI-STSb-large	86.15 ± 0.35

Table 2: Evaluation on the STS benchmark test set. BERT systems were trained with 10 random seeds and 4 epochs. SBERT was fine-tuned on the STSb dataset, SBERT-NLI was pretrained on the NLI datasets, then fine-tuned on the STSb dataset.

Evaluation-STS

- Argument Facet Similarity
  - Pretrain Dataset
    - AFS corpus
- Wikipedia Sections Distinction
  - Pretrain Dataset
    - Wikipedia
  - Loss Function
    - Triplet Loss

Model	r	ρ
Unsupervised methods		
tf-idf	46.77	42.95
Avg. GloVe embeddings	32.40	34.00
InferSent - GloVe	27.08	26.63
10-fold Cross-Validation		
SVR (Misra et al., 2016)	63.33	-
BERT-AFS-base	77.20	74.84
SBERT-AFS-base	76.57	74.13
BERT-AFS-large	78.68	76.38
SBERT-AFS-large	77.85	75.93
Cross-Topic Evaluation		
BERT-AFS-base	58.49	57.23
SBERT-AFS-base	52.34	50.65
BERT-AFS-large	62.02	60.34
SBERT-AFS-large	53.82	53.10

Table 3: Average Pearson correlation r and average Spearman's rank correlation ρ on the Argument Facet Similarity (AFS) corpus (Misra et al., 2016). Misra et al. proposes 10-fold cross-validation. We additionally evaluate in a cross-topic scenario: Methods are trained on two topics, and are evaluated on the third topic.

Model	Accuracy
mean-vectors	0.65
skip-thoughts-CS	0.62
Dor et al.	0.74
SBERT-WikiSec-base	0.8042
SBERT-WikiSec-large	**0.8078**
SRoBERTa-WikiSec-base	0.7945
SRoBERTa-WikiSec-large	0.7973

Table 4: Evaluation on the Wikipedia section triplets dataset (Dor et al., 2018). SBERT trained with triplet loss for one epoch.

Evaluation-SentEval

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Avg. GloVe embeddings	77.25	78.30	91.17	87.85	80.18	83.0	72.87	81.52
Avg. fast-text embeddings	77.96	79.23	91.68	87.81	82.15	83.6	74.49	82.42
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.8	69.45	84.94
BERT CLS-vector	78.68	84.85	94.21	88.23	84.13	91.4	71.13	84.66
InferSent - GloVe	81.57	86.54	92.50	**90.38**	84.18	88.2	75.77	85.59
Universal Sentence Encoder	80.09	85.19	93.98	86.70	86.38	**93.2**	70.14	85.10
SBERT-NLI-base	83.64	89.43	94.39	89.86	88.96	89.6	**76.00**	87.41
SBERT-NLI-large	**84.88**	**90.07**	**94.52**	90.33	**90.66**	87.4	75.94	**87.69**

Table 5: Evaluation of SBERT sentence embeddings using the SentEval toolkit. SentEval evaluates sentence embeddings on different sentence classification tasks by training a logistic regression classifier using the sentence embeddings as features. Scores are based on a 10-fold cross-validation.

Ablation Study

- Pooling Strategy
  - Mean Pooling…!

- Concatenation
  - Element-wise Difference

	NLI	STSb
Pooling Strategy		
MEAN	**80.78**	**87.44**
MAX	79.07	69.92
CLS	79.80	86.62
Concatenation		
(u, v)	66.04	-
$(\|u - v\|)$	69.78	-
$(u * v)$	70.54	-
$(\|u - v\|, u * v)$	78.37	-
$(u, v, u * v)$	77.44	-
$(u, v, \|u - v\|)$	**80.78**	-
$(u, v, \|u - v\|, u * v)$	80.44	-

Table 6: SBERT trained on NLI data with the classification objective function, on the STS benchmark (STSb) with the regression objective function. Configurations are evaluated on the development set of the STSb using cosine-similarity and Spearman's rank correlation. For the concatenation methods, we only report scores with MEAN pooling strategy.

Computational Efficiency

Model	CPU	GPU
Avg. GloVe embeddings	6469	-
InferSent	137	1876
Universal Sentence Encoder	67	1318
SBERT-base	44	1378
SBERT-base - smart batching	83	2042

Table 7: Computation speed (sentences per second) of sentence embedding methods. Higher is better.

Q&A

[Github]
http://github.com/UKPLab/sentence-transformers