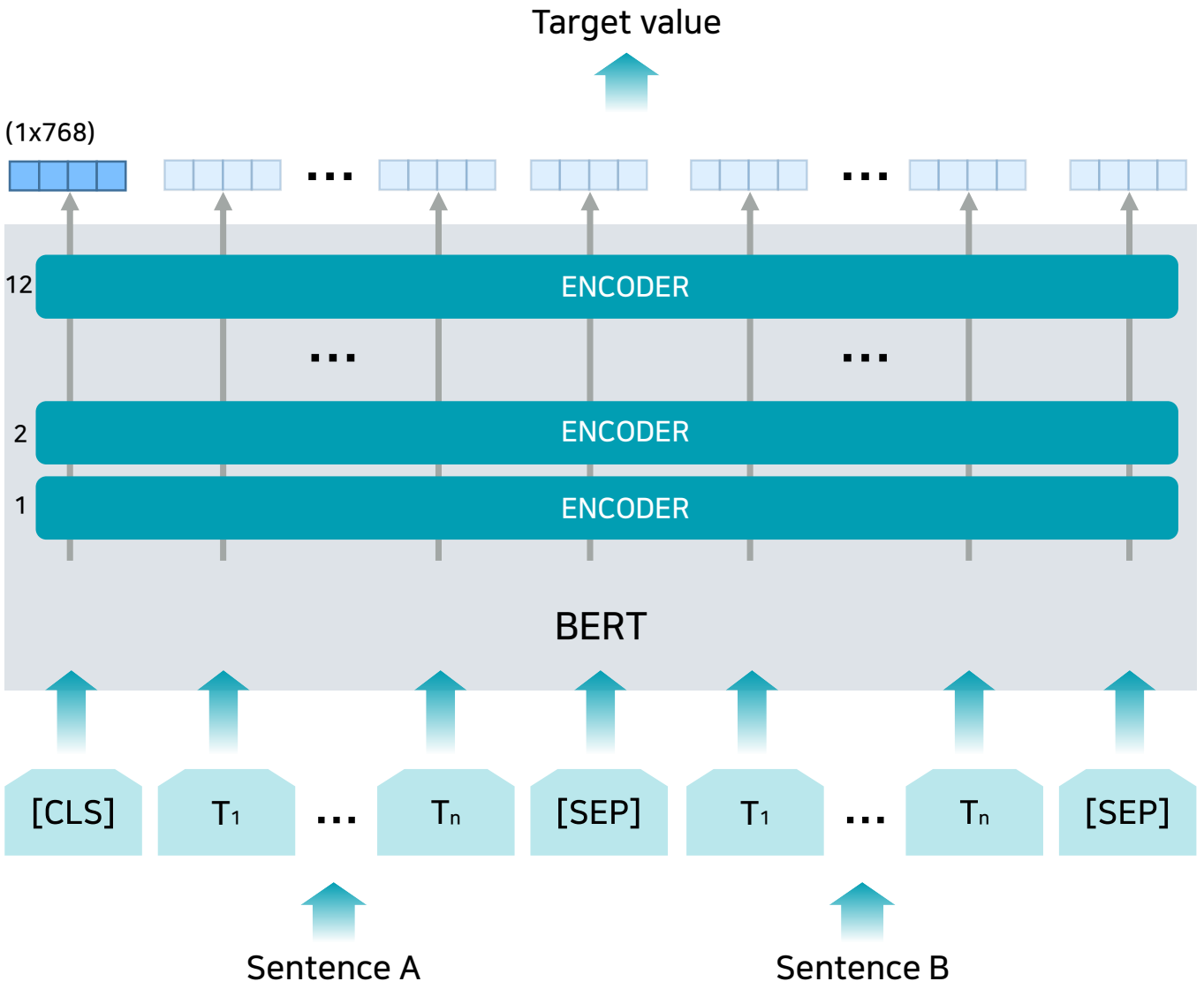
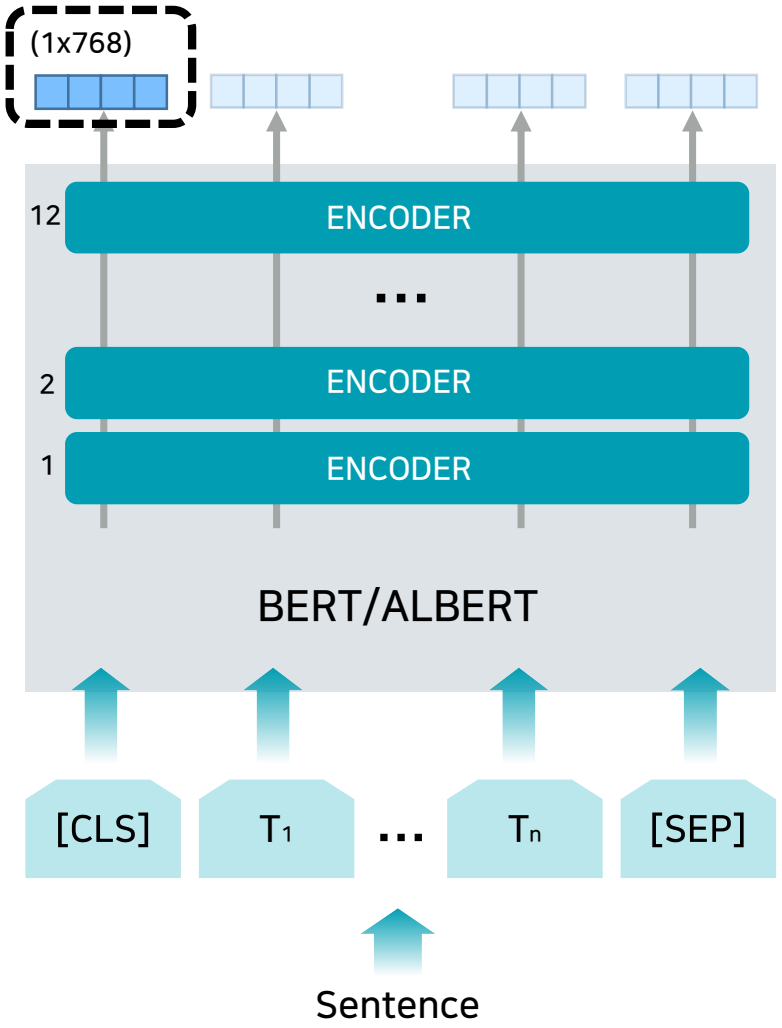
Evaluation of BERT and ALBERT Sentence Embedding Performance on Downstream NLP Tasks

Hyunjin Choi, Judong Kim, Seongho Joe, Youngjune Gwon
Samsung SDS

BERT on sentence-pair regression tasks



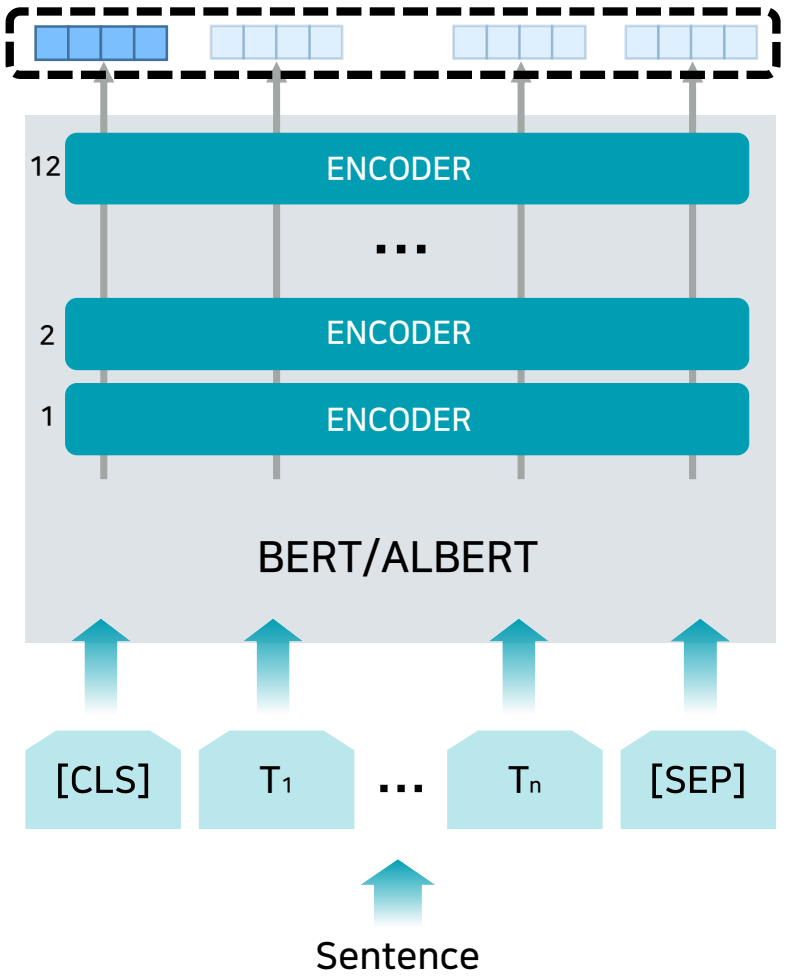
Sentence Embedding Models



❶ [CLS] token embedding

- [CLS] token summarizes the information from other tokens via a self-attention mechanism
- The most straightforward sentence embedding
- Can be further optimized while fine-tuning the downstream task

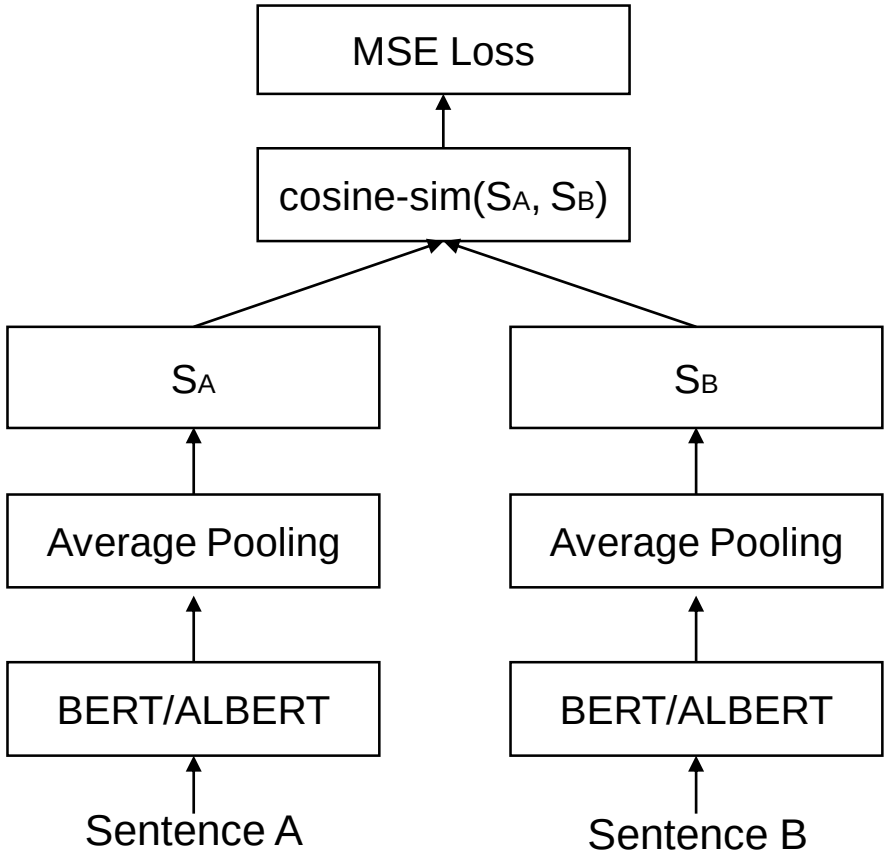
Sentence Embedding Models



❷ Pooled token embeddings

- Make fixed-length sentence vector by
  (1) averaging the token embedding output
  (2) max pooling
- Works like a pooling layer in a convolutional neural net

Sentence Embedding Models



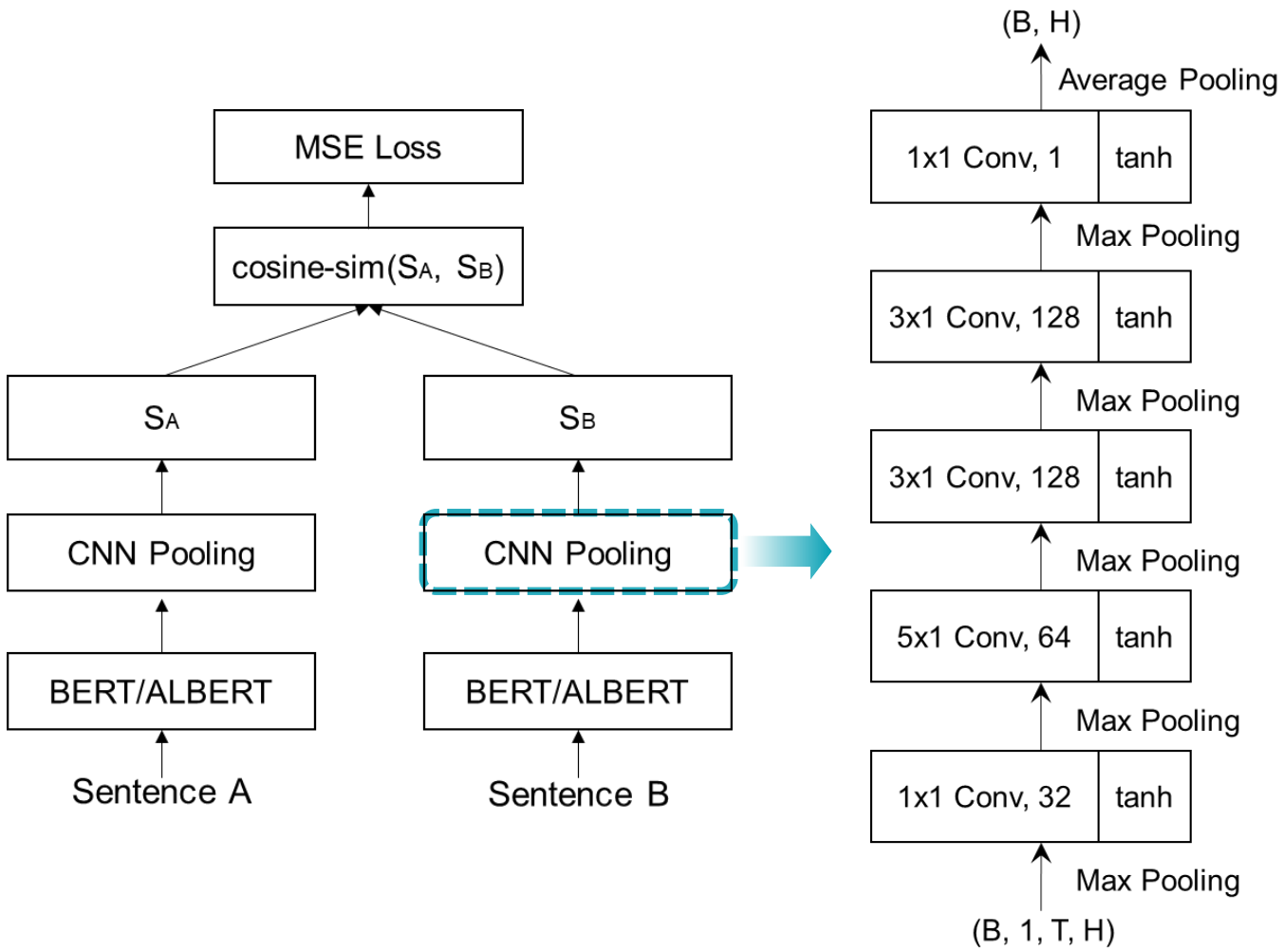
❸ Sentence-BERT (SBERT)

- Reimers & Gurevych
- Siamese network structures
- Average-pools a pair of the BERT embeddings to fixed-size sentence embeddings
- Using cosine similarity to derive semantically meaningful sentence embeddings

❹ Sentence-ALBERT (SALBERT)

- Based on ALBERT
- Same Siamese networks as SBERT

Sentence Embedding Models



❺ CNN-SBERT **❻ CNN-SALBERT**

- Employ a CNN architecture instead of average pooling to make fixed-size sentence vectors
- Convolutional layers with the hyperbolic tangent activation function interlaced with pooling layers

Datasets and tasks

❶ Semantic Textual Similarity

- Evaluate the similarity between two sentences (regression task)
- Semantic Textual Similarity benchmark (STSb)

❷ Natural Language Inference

- Determine whether a “hypothesis” is true (entailment), false (contradiction), or undetermined (neutral) given a “premise”
- Stanford Natural Language Inference (SNLI) corpus
- Multi-Genre Natural Language Inference (MultiNLI) corpus

Results

TABLE I
EVALUATION ON THE STSB BY FINE-TUNING SENTENCE EMBEDDINGS ON STS, NLI, AND BOTH

Model	Spearman (Pearson)
Not fine-tuned	
BERT [CLS]-token embedding	6.43 (1.70)
BERT Avg. pooled token embedding	47.29 (47.91)
ALBERT [CLS]-token embedding	0.86 (4.57)
ALBERT Avg. pooled token embedding	47.84 (46.57)
Fine-tuned on STSb	
BERT [CLS]-token embedding	12.96 (7.49)
BERT Avg. pooled token embedding	55.76 (54.90)
SBERT	84.66 (84.86)
CNN-SBERT	**85.72 (86.15)**
ALBERT [CLS]-token embedding	37.98 (27.89)
ALBERT Avg. pooled token embedding	61.06 (60.41)
SALBERT	74.33 (75.26)
CNN-SALBERT	**82.30 (83.08)**
Fine-tuned on NLI (MultiNLI + SNLI)	
BERT [CLS]-token embedding	32.72 (26.88)
BERT Avg. pooled token embedding	69.57 (68.49)
SBERT	**77.22 (74.53)**
CNN-SBERT	76.77 (75.31)
ALBERT [CLS]-token embedding	24.87 (4.11)
ALBERT Avg. pooled token embedding	54.21 (53.58)
SALBERT	**74.05 (70.78)**
CNN-SALBERT	73.70 (72.24)
Fine-tuned on NLI (MultiNLI + SNLI) and STSb	
BERT [CLS]-token embedding	44.77 (38.74)
BERT Avg. pooled token embedding	67.61 (65.30)
SBERT	85.32 (84.51)
CNN-SBERT	**85.91 (85.63)**
ALBERT [CLS]-token embedding	40.35 (33.46)
ALBERT Avg. pooled token embedding	60.24 (59.98)
SALBERT	77.59 (77.82)
CNN-SALBERT	**83.49 (83.87)**

Conclusion

- This paper presents an evaluation of BERT and ALBERT sentence embedding models on Semantic Textual Similarity (STS)
- Knowing limitations of the [CLS] token vector, we adopt Siamese network architecture by Reimers & Gurevych for BERT and ALBERT
- Developed a CNN architecture that takes in the token embeddings to compute a fixed-size sentence vector
- CNN architecture improves ALBERT model up to 8% in Spearman's rank correlation
- Despite significantly fewer model parameters, ALBERT sentence embedding is highly competitive to BERT in downstream NLP evaluations