Sentence-T5 (ST5): Scalable Sentence Encoders from Pre-trained Text-to-Text Models

By: Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma,

Keith B. Hall, Daniel Cer, Yinfei Yang

Affiliation: Google Research

Meta Data

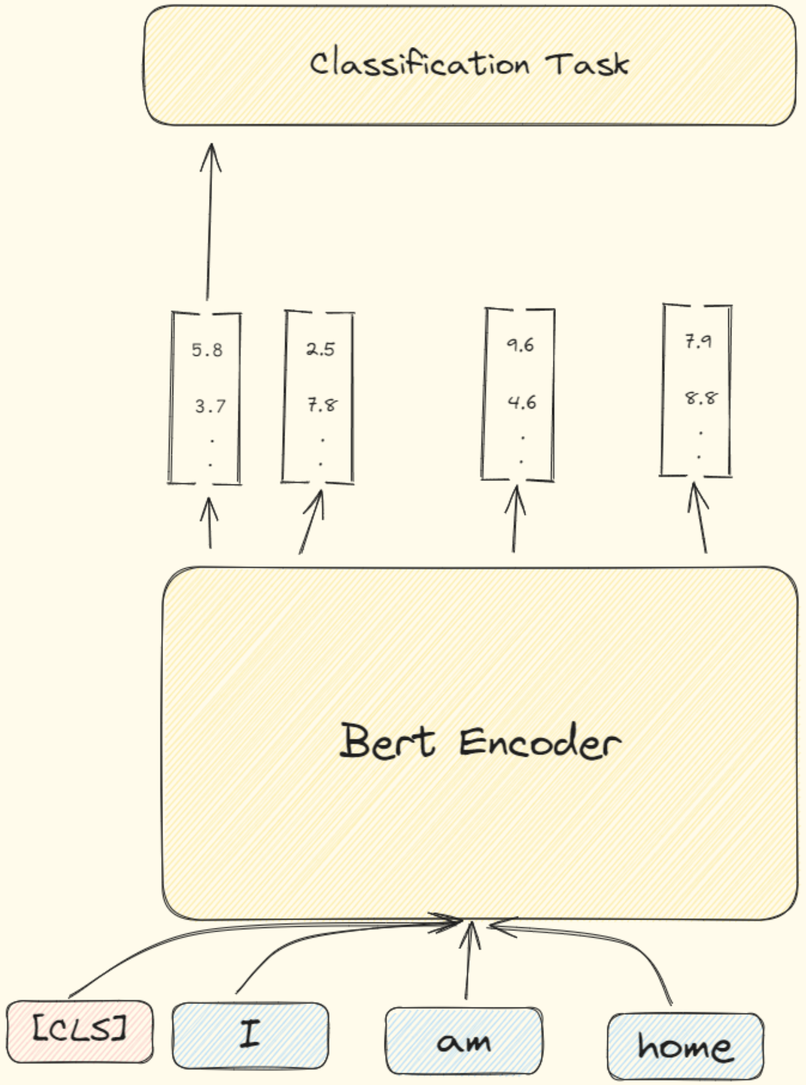
- Venue: ACL 2022
  - CORERank: A*
  - h5-index: 215
- Year 2022
- Citations: 380

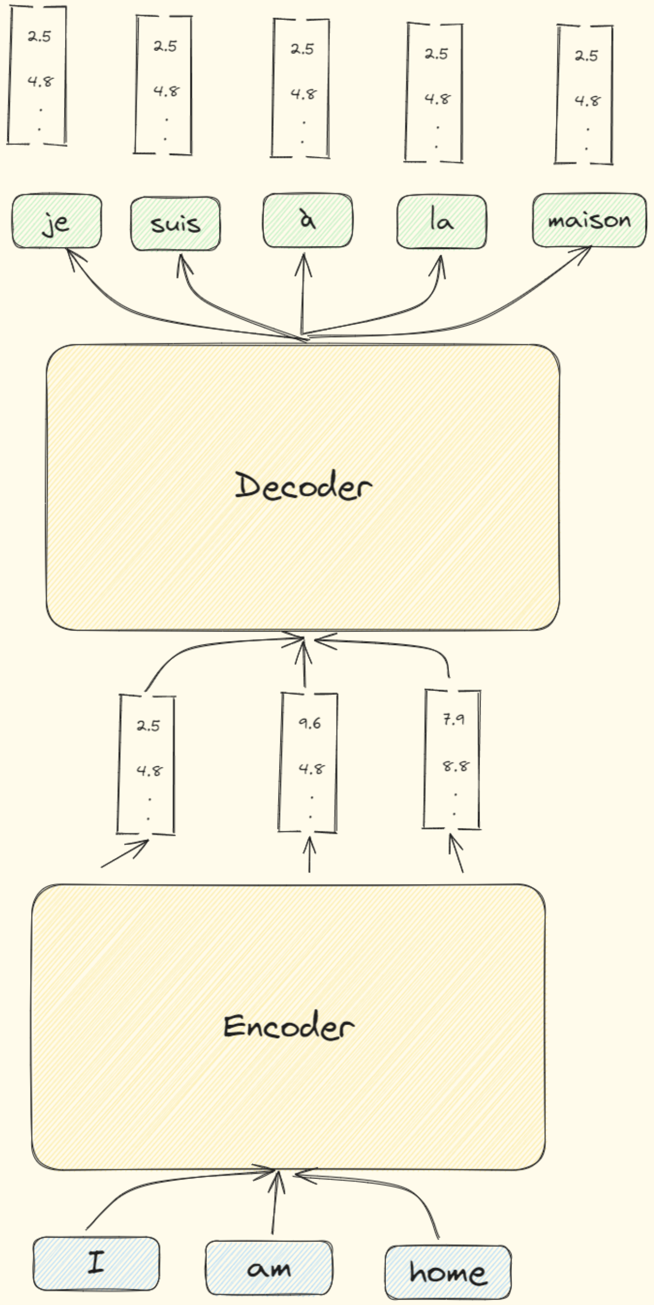
Core Contributions

- First to study using large-scale pretrained seq2seq models for sentence representation learning and scale sentence embedding upto 11B parameters.
- Sentence-T5 (ST5) encoder only models perform well on sentence transfer tasks by out performing state-of-the-art (SOTA) models such as SimCSE BERT and SimCSE RoBERTa models, even without fine-tuning.
- They show that their encoder-decoder ST5 models perform better in semantic textual similarity tasks (STS), achieving SOTA performance on STS.
- They show that contrastive learning is effective for fine-tuning sentence encoders from T5-style pretrained models (essentially meaning generative span corruption pre-training task).
- They show that training ST5 longer and with more data using a contrastive loss leads to consistent improvement in sentence transfer and STS tasks.
- They also created a new sentence representation transfer benchmark dataset called SentGLUE.

Significance of this work

- Seq2Seq perform well on generative tasks (summarization, translation), but fail when it comes to tasks such as classification, clustering. By obtaining sentence level representation, seq2seq models also can be used for such tasks.
- Since seq2seq models generate string of tokens, it is difficult to perform contrastive learning approaches as there is not a single representation to the input for which we can perform contrastive learning.


Classification Task
5.8
3.7
2.5
7.8
9.6
4.6
7.9
8.8
Bert Encoder
[CLS]
I
am
home

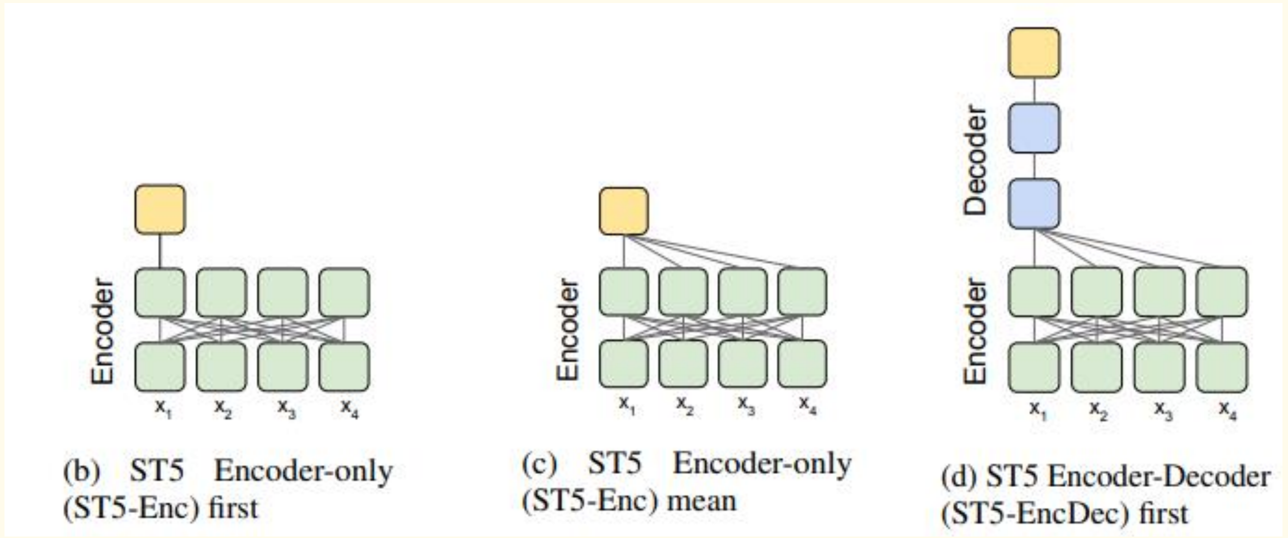

2.5
4.8
je
suis
à
la
maison
Decoder
2.5
4.8
9.6
4.8
7.9
8.8
Encoder
I
am
home

Training Specifications

- Model used: T5 family
- SOTA Models used: BERT, RoBERTa
- Evaluation: SentEval which contains 7 transfer task and 7 STS task. SentGLUE (their own benchmark dataset)
- JAX framework was used for training and model implementation.
- Training
  - Adafactor Optimizer
  - Learning rate of 0.001
  - Linear decay after 10% of the total number of training step.
  - Softmax temperature of 0.01

Approach

What token embedding should we take?



(b) ST5 Encoder-only (ST5-Enc) first

(c) ST5 Encoder-only (ST5-Enc) mean

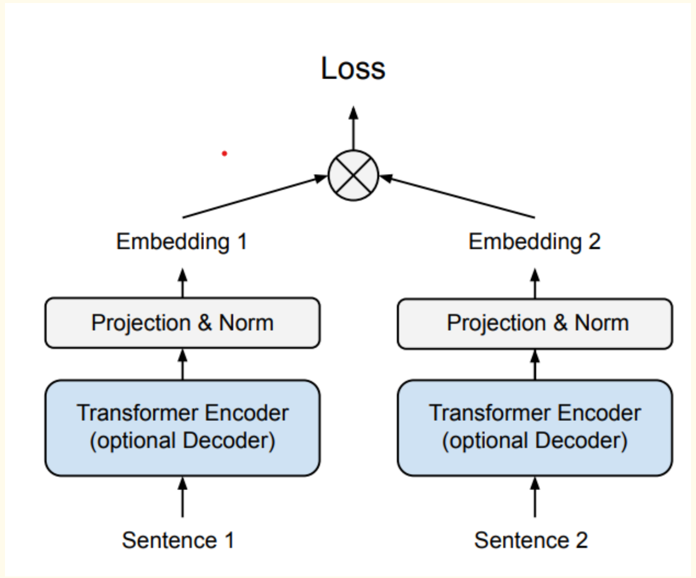
(d) ST5 Encoder-Decoder (ST5-EncDec) first

They approach this in three ways

- Encoder-only first: the encoder output of the first token. (diagram b)
- Encoder-only mean: the average encoder output across all input tokens. (diagram c)
- Encoder-Decoder first: the first decoder output when the input text is fed into the encoder and the standard "start" token is fed into the decoder. (diagram d)

Assumption : *"we assume the decoder is aware of the semantics of the entire input sentence when generating its first token prediction; and if so, the first decoder output embeddings (i.e. input to the softmax layer) might naturally capture the sentence semantics."*

Sentence Encoder Training



- They adopt a dual encoder architecture.
- The architecture consist of two shared weight transformer.
- Once the sentences are encoded by the transformer, the embedding is passed through a projection layer and L2-normalization unit.
- Finally, a contrastive loss is applied on these two sentence embeddings.

Contrastive Loss

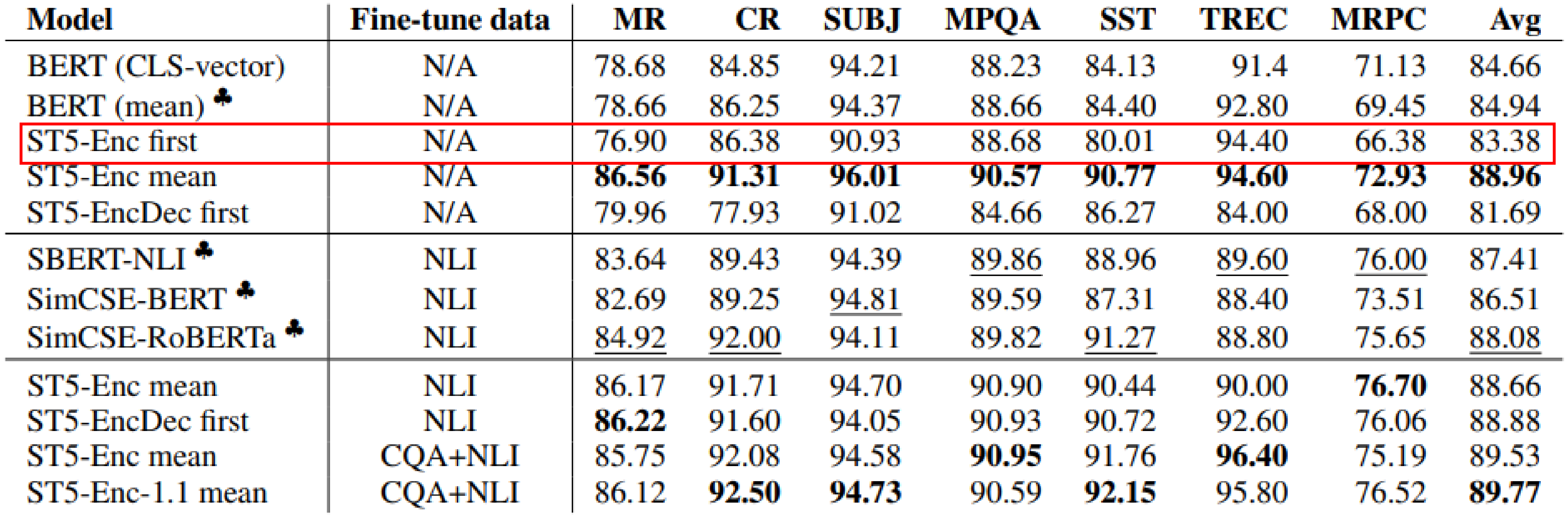
Contrastive loss improves uniformity in sentence embedding, which leads to better performance on downstream task.

Authors employ contrastive learning in the fine-tuning stage.

$$\mathcal{L} = \frac{e^{\mathrm{sim}(v_i, v_i^+)/\tau}}{\sum_{j \in \mathcal{B}} e^{\mathrm{sim}(v_i, v_j^+)/\tau}},$$

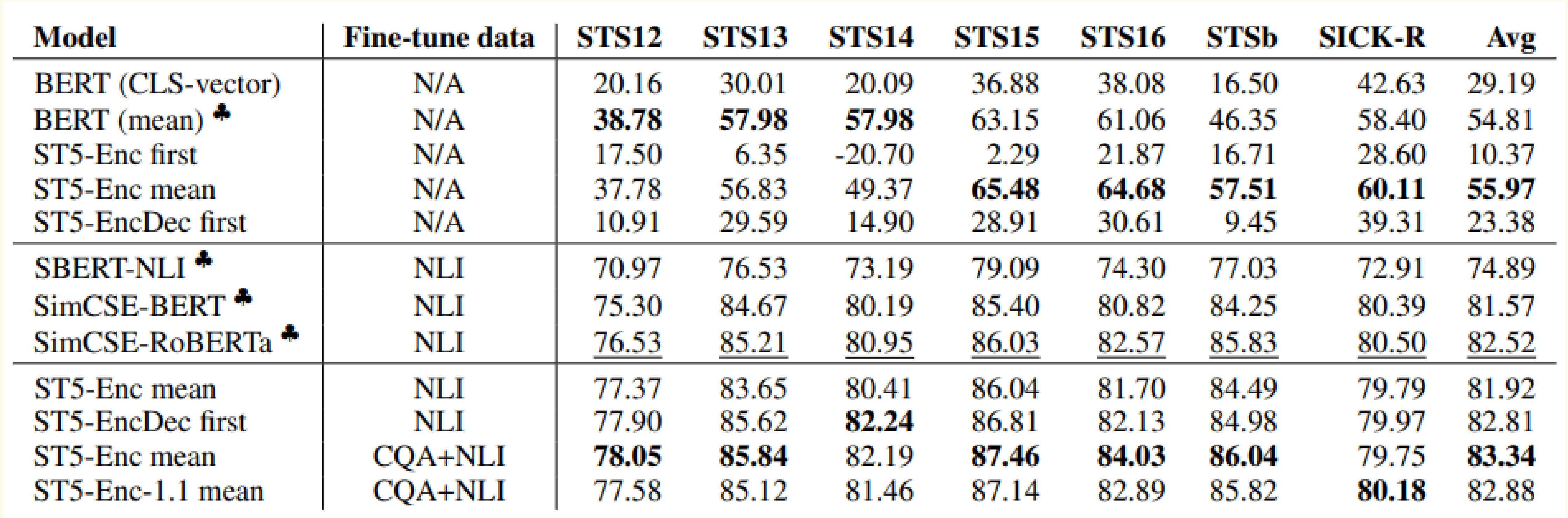
$$\mathcal{L} = \frac{e^{\mathrm{sim}(v_i, v_i^+)/\tau}}{\sum_{j \in \mathcal{B}} e^{\mathrm{sim}(v_i, v_j^+)/\tau} + e^{\mathrm{sim}(v_i, v_j^-)/\tau}}.$$

Results



Model	Fine-tune data	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg
BERT (CLS-vector)	N/A	78.68	84.85	94.21	88.23	84.13	91.4	71.13	84.66
BERT (mean) ♣	N/A	78.66	86.25	94.37	88.66	84.40	92.80	69.45	84.94
ST5-Enc first	N/A	76.90	86.38	90.93	88.68	80.01	94.40	66.38	83.38
ST5-Enc mean	N/A	**86.56**	**91.31**	**96.01**	**90.57**	**90.77**	**94.60**	**72.93**	**88.96**
ST5-EncDec first	N/A	79.96	77.93	91.02	84.66	86.27	84.00	68.00	81.69
SBERT-NLI ♣	NLI	83.64	89.43	94.39	89.86	88.96	89.60	76.00	87.41
SimCSE-BERT ♣	NLI	82.69	89.25	94.81	89.59	87.31	88.40	73.51	86.51
SimCSE-RoBERTa ♣	NLI	84.92	92.00	94.11	89.82	91.27	88.80	75.65	88.08
ST5-Enc mean	NLI	86.17	91.71	94.70	90.90	90.44	90.00	**76.70**	88.66
ST5-EncDec first	NLI	**86.22**	91.60	94.05	90.93	90.72	92.60	76.06	88.88
ST5-Enc mean	CQA+NLI	85.75	92.08	94.58	**90.95**	91.76	**96.40**	75.19	89.53
ST5-Enc-1.1 mean	CQA+NLI	86.12	**92.50**	**94.73**	90.59	**92.15**	95.80	76.52	**89.77**

Table 2: Performance on transfer tasks on the SentEval benchmark. All models are using the **Base** architecture. ♣ results are from (Gao et al., 2021). For all tasks, a logistic regression classifier is trained using the sentence embeddings as features and the classification accuracy on test sets are reported.



Model	Fine-tune data	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg
BERT (CLS-vector)	N/A	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
BERT (mean) ♣	N/A	**38.78**	**57.98**	**57.98**	63.15	61.06	46.35	58.40	54.81
ST5-Enc first	N/A	17.50	6.35	-20.70	2.29	21.87	16.71	28.60	10.37
ST5-Enc mean	N/A	37.78	56.83	49.37	**65.48**	**64.68**	**57.51**	**60.11**	**55.97**
ST5-EncDec first	N/A	10.91	29.59	14.90	28.91	30.61	9.45	39.31	23.38
SBERT-NLI ♣	NLI	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SimCSE-BERT ♣	NLI	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
SimCSE-RoBERTa ♣	NLI	76.53	85.21	80.95	86.03	82.57	85.83	80.50	82.52
ST5-Enc mean	NLI	77.37	83.65	80.41	86.04	81.70	84.49	79.79	81.92
ST5-EncDec first	NLI	77.90	85.62	**82.24**	86.81	82.13	84.98	79.97	82.81
ST5-Enc mean	CQA+NLI	**78.05**	**85.84**	82.19	**87.46**	**84.03**	**86.04**	79.75	**83.34**
ST5-Enc-1.1 mean	CQA+NLI	77.58	85.12	81.46	87.14	82.89	85.82	**80.18**	82.88

Table 3: Spearman's correlation coefficient ($\times 100$) on STS tasks on the SentEval benchmark. All models are using the **Base** architecture. ♣ results are from (Gao et al., 2021).

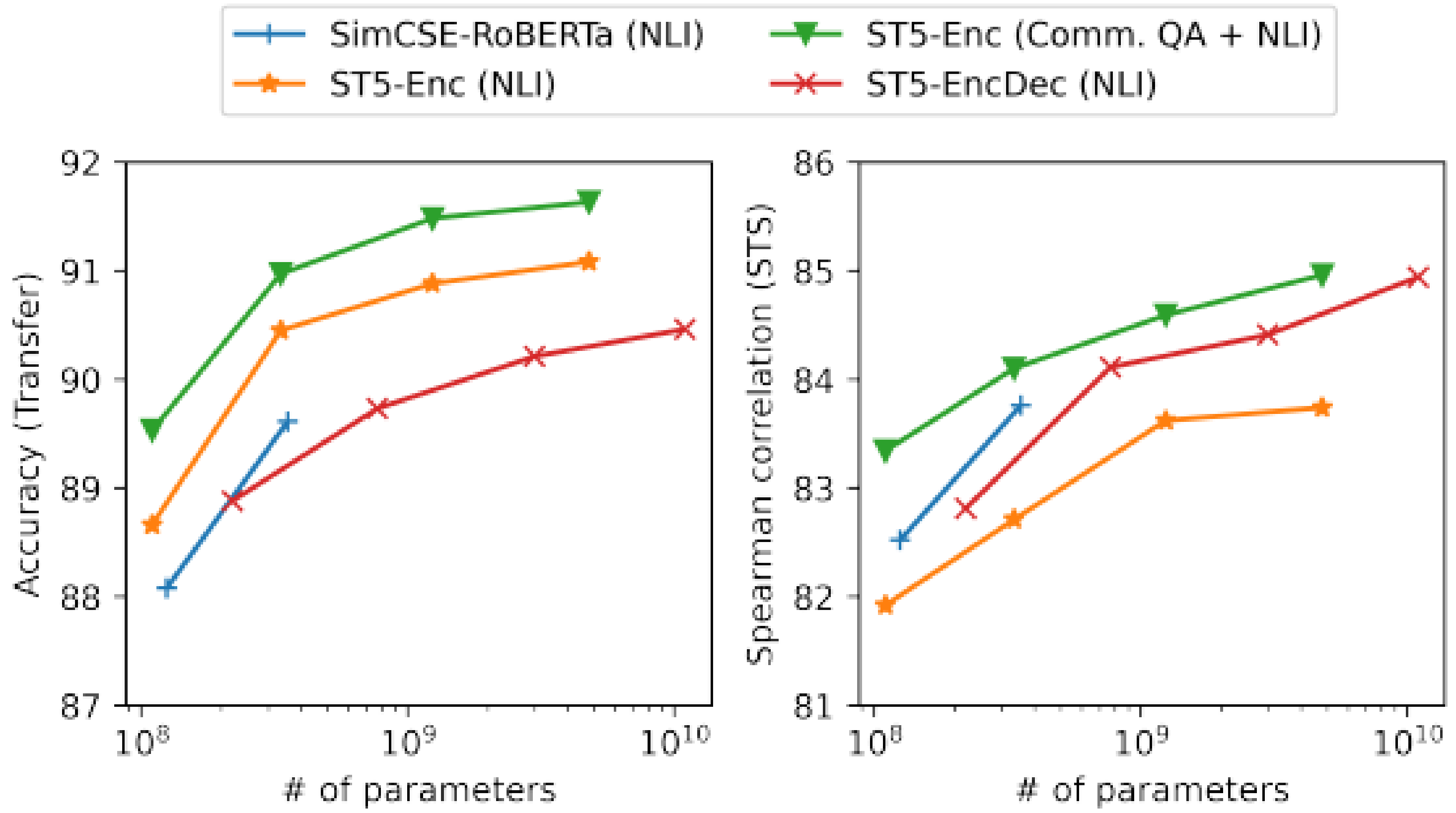


Figure 1: Scaling up our ST5 model size improves performance on SentEval (left) and STS (right).