Sentence Representation Learning

Exercise 8

Fabian David Schmidt & Benedikt Ebing

Supervised Representation Learning (1/2)

Q2.1: Explain the training objective of the original Sentence-BERT transformer. Why does the objective enable cosine similarity search at inference time?

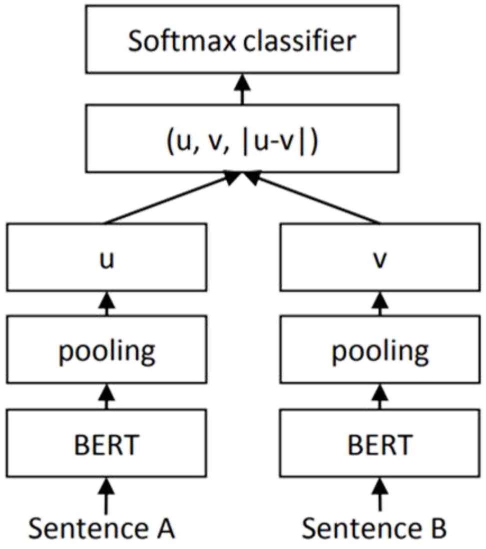


Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

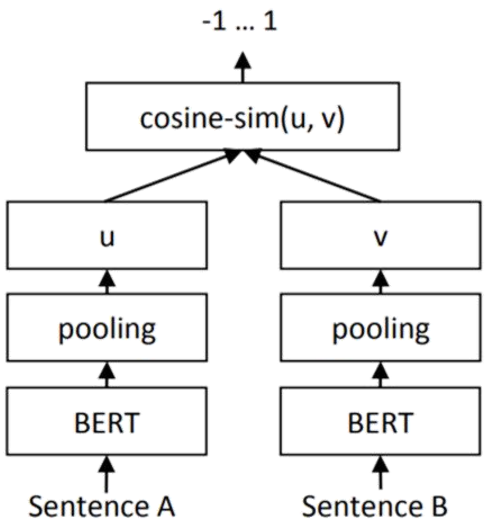


Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

- u, v are sentence representations of sentence pair
- Softmax objective trained on NLI linearly separates (u, v, |u-v|) into entailment, contradiction, neutral
- Linear separation into classes closely related to angle of canonical class representation (i.e, each class vector in classifier)
- Classes align well with idea of sentence-level semantics
- Good downstream (e.g., semantic search) representations

Supervised Representation Learning (2/2)

Q2.2: Can you think of intuitions as to why SRoBerta does not outperform SBERT, in contrast to other types of downstream tasks?

- BERT pre-trained with Masked Language Modelling and Next Sentence Prediction objectives
- RoBERTa only trained with Masked Language Modelling
- Neither of the two pretrains on sentence-level semantics very well, esp. on mean-pooled representations of token as a sentence embedding

Self-Supervised Representation Learning (1/3)

Q3.1: Briefly explain the core idea of contrastive learning and how the training objective is typically constructed.

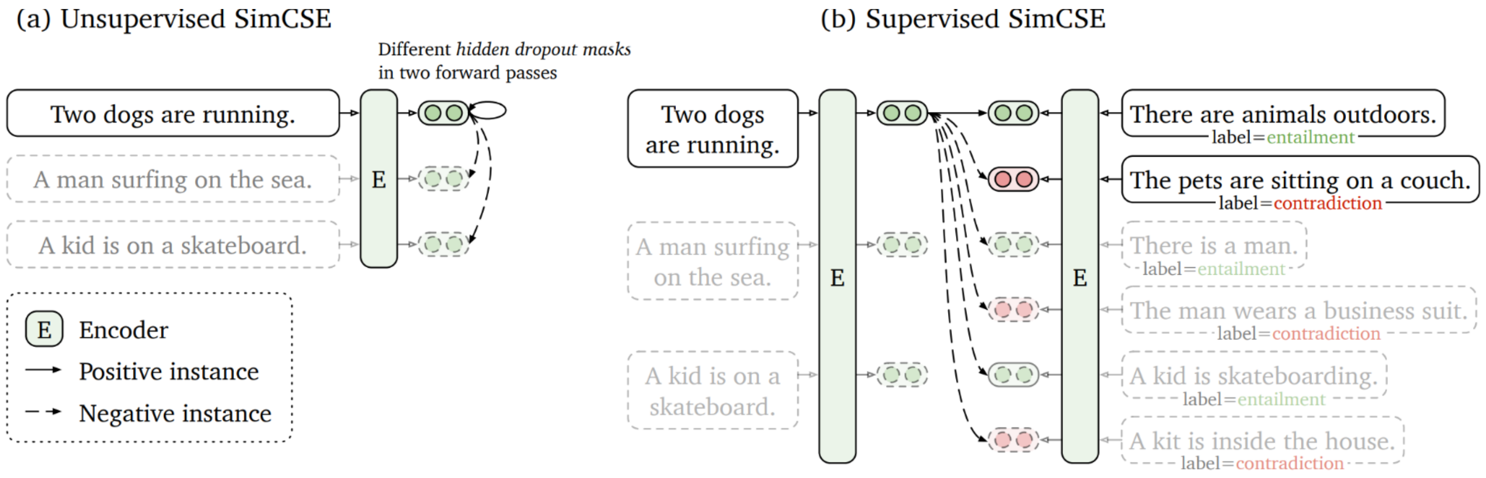


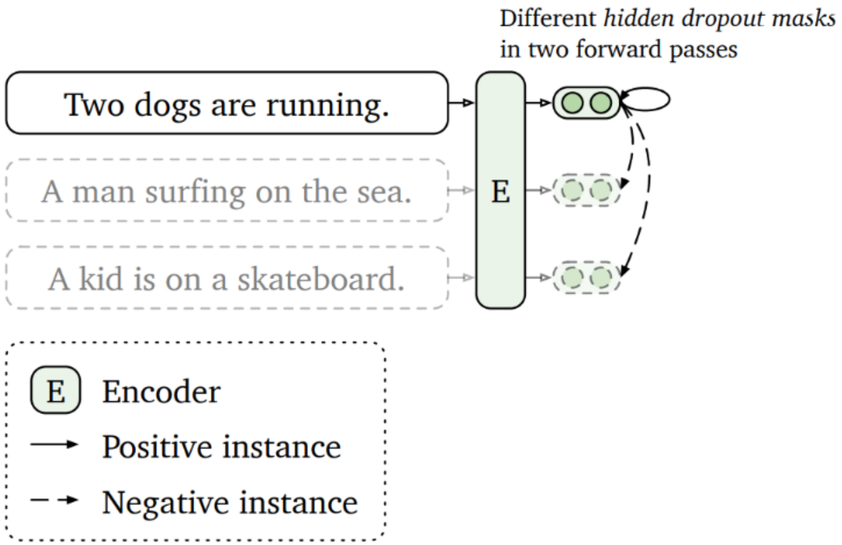
Figure 1: (a) Unsupervised SimCSE predicts the input sentence itself from in-batch negatives, with different hidden dropout masks applied. (b) Supervised SimCSE leverages the NLI datasets and takes the entailment (premise-hypothesis) pairs as positives, and contradiction pairs as well as other in-batch instances as negatives.

- **Core idea:** attract positive instances closer in representation space, repel negative instances
- **Loss:** softmax over cosine similarity typically in batch expressed as “multi-class classification” with 1 to k positive examples, all other in-batch instances are negatives
- **Considerations:** how to treat more than 1 positive, batch size (the larger the better!), multi-GPU training (where to put examples, other objectives, etc.)

Self-Supervised Representation Learning (2/3)

Q3.2: How does unsupervised SimCSE learn sentence-level representations in a selfsupervised fashion? How does it thereby improve over other potentially selfsupervised objectives?

(a) Unsupervised SimCSE



Data augmentation			STS-B
None (unsup. SimCSE)			**82.5**
Crop	*10%*	*20%*	*30%*
	77.8	71.4	63.6
Word deletion	*10%*	*20%*	*30%*
	75.9	72.2	68.2
Delete one word			75.9
w/o dropout			74.2
Synonym replacement			77.4
MLM 15%			62.2

Table 1: Comparison of data augmentations on STS-B development set (Spearman's correlation). *Crop k%*: keep 100-k% of the length; *word deletion k%*: delete k% words; *Synonym replacement*: use `nlpaug` (Ma, 2019) to randomly replace one word with its synonym; *MLM k%*: use $\text{BERT}_{\text{base}}$ to replace k% of words.

- **Unsupervised SimCSE:** positive pair are repeated forward passes of the **same** instance, negatives are all other sentence within a batch
- **Repeated forward pass** results in very different sentence embeddings since initial output is highly misaligned and dropout masks meaningfully distort output
- **Other strategies** (cropping, word deletion, MLMing) are destructive in semantics to potentially align output incorrectly

Self-Supervised Representation Learning (3/3)

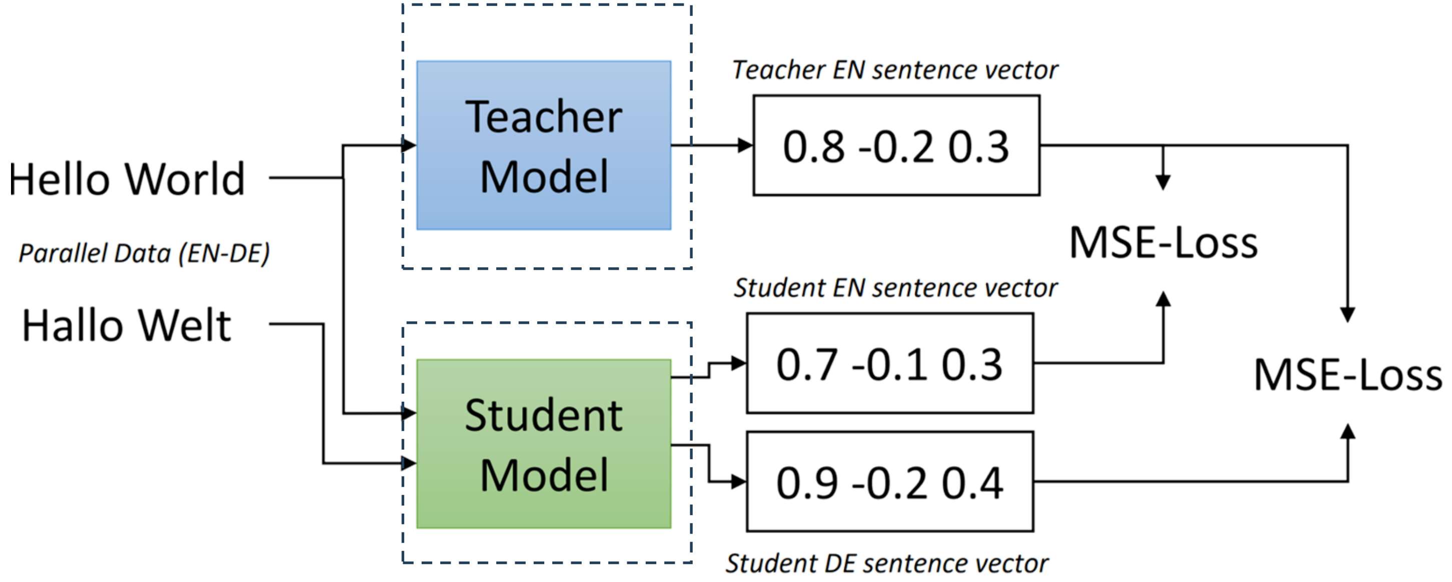
Q3.3: . Imagine you want to train your own multilingual sentence transformer. List and briefly explain some key considerations in scaling up the training procedure.

- **Training objective:** typically some variant of contrastive loss, but maybe should also include language modelling (MLM, TLM) objectives
- **Training data:** large scale monolingual and parallel (bi- or n-way multilingual data)
- **Architecture:** sentence embedding models are typically not exorbitantly large; 12 to 24 layers should suffice
- **Tokenizer:** large-scale multilingual models should probably allocate a large capacity into the number of tokens (250-750K); trend goes towards larger vocabularies (varying scripts in multilinguality, programming languages, etc.)

Knowledge Distillation

Q4.1: What is knowledge distillation and how does it work (on the case of multilingual sentence transformers)?

Teacher Model Typically is a English sentence transformer trained on large-scale data



Multilingual model, e.g. XLM-Roberta

- **Core idea:** we re-lever sentence alignment of a pre-trained sentence embedder (teacher model) to align or multilingual model on parallel data
- **Parallel data:** sentence translations that are guaranteed to be semantically aligned
- **Objective:** MSE loss to minimize distance between teacher and student embeddings; other variants, e.g, on cosine similarity also conceivable
- **Q3.2:** quality of teacher and amount of data most critical – we can „only“ replicate teacher and do so in best possible fashion

Q5.1: You are given the following two embedding pairs from a bi-encoder. Compute the InfoNCE loss with cosine similarity and temperature T=0.5 as shown in the lecture slides.

$$\text{Positive Pair} \rightarrow [0.8109, -0.9391, 0.2519], [-1.2887, 1.5057, 0.4449]$$
$$\text{Negative Pair} \rightarrow [0.8109, -0.9391, 0.2519], [2.1968, 0.4785, 1.5207]$$

Q5.1: You are given the following two embedding pairs from a bi-encoder. Compute the InfoNCE loss with cosine similarity and temperature T=0.5 as shown in the lecture slides.

L2-Normalized Embeddings:

$$\text{Normalized Positive Pair}$$
$$\rightarrow a: [0.6405, -0.7417, 0.1990], \text{b:}[-0.6344, 0.7413, 0.2190]$$
$$\text{Normalized Negative Pair}$$
$$\rightarrow a: [0.6405, -0.7417, 0.1990], c: [0.8093, 0.1763, 0.5603]$$

Cosine Similarity

$$ab^T/0.5 = -1.825$$
$$ac^T/0.5 = 0.9982$$

Q5.1: You are given the following two embedding pairs from a bi-encoder. Compute the InfoNCE loss with cosine similarity and temperature T=0.5 as shown in the lecture slides.

Loss:

$$loss = -\ln \frac{e^{\frac{s_{i,j}}{T}}}{e^{\frac{s_{i,j}}{T}} + \sum_{k=1}^{N} e^{\frac{s_{i,k}}{T}}}$$

$$loss = -\ln \frac{e^{-1.825}}{e^{-1.825} + e^{0.9982}} = 2.8811$$