Fine-Tuning BERT to Understand Semantic Textual Relatedness

Justin Liu

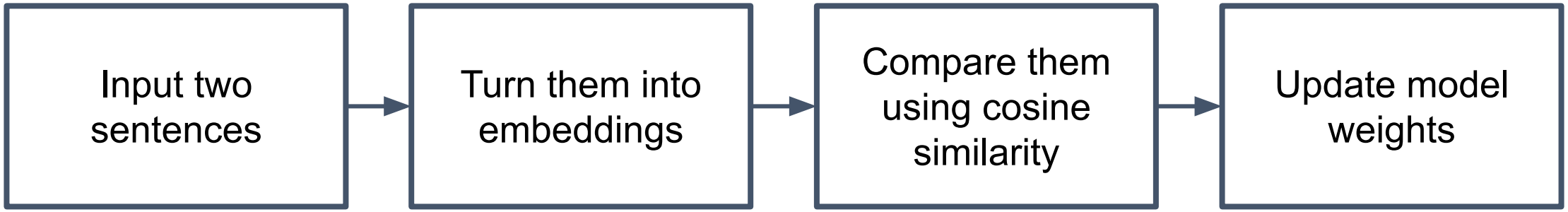
Data

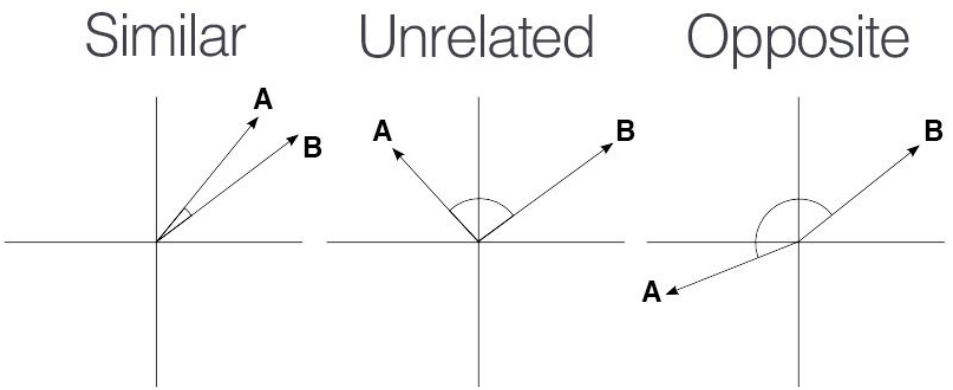
- Abdalla et al. (2023): What Makes Sentences Semantically Related? A Textual Relatedness Dataset and Empirical Study
  - 5500 pairs of English sentences, each with a score ranging from 0 (completely unrelated) to 1 (maximally related) indicating how **similar** the sentences are in meaning (i.e., degree of semantic relatedness)
  - The scores were obtained using **comparative annotation** (sentence pairs are presented together, and each annotator ranks them from most to least similar)

Score	Sentence 1	Sentence 2
1.0	No, I really didn't want New York to win.	No I didn't want New York to win
0.88	Excuse me, but perhaps you misunderstand me.	I'm sorry, but I don't think you understood me
0.34	Two men standing in grass staring at a car	Two women sitting in chairs in a shop
0.06	This isn't an English sentence	Life isn't always like that

Using BERT on its own

- BERT is a popular language model that was released by Google in 2018
- BERT is not specifically made for sentence similarity tasks, so we need to **fine-tune it**





How the model learns!

- However, BERT outputs **embeddings**, not similarity scores!
  - Without these scores, we can’t update the model weights

Solution: Siamese BERT model architecture

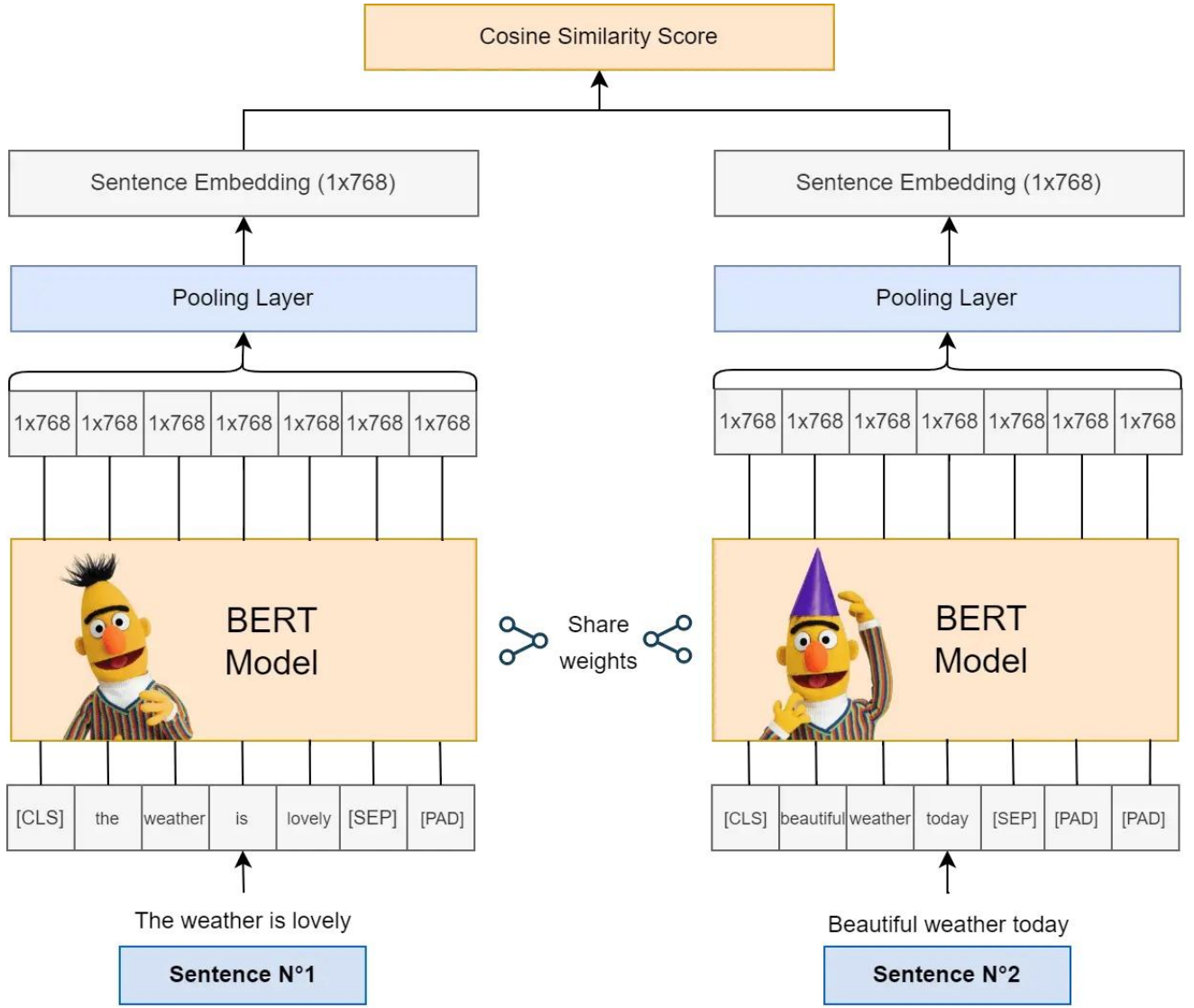
5) Get a cosine similarity score between -1 (very different) and 1 (very similar)

4) The token embeddings are pooled together (usually by taking the mean) to get an overall sentence embedding

3) Each token is turned into a 768-dimensional embedding

2) The sentences are tokenized

1) 2 sentences are taken as input



Source: [The Python Code]

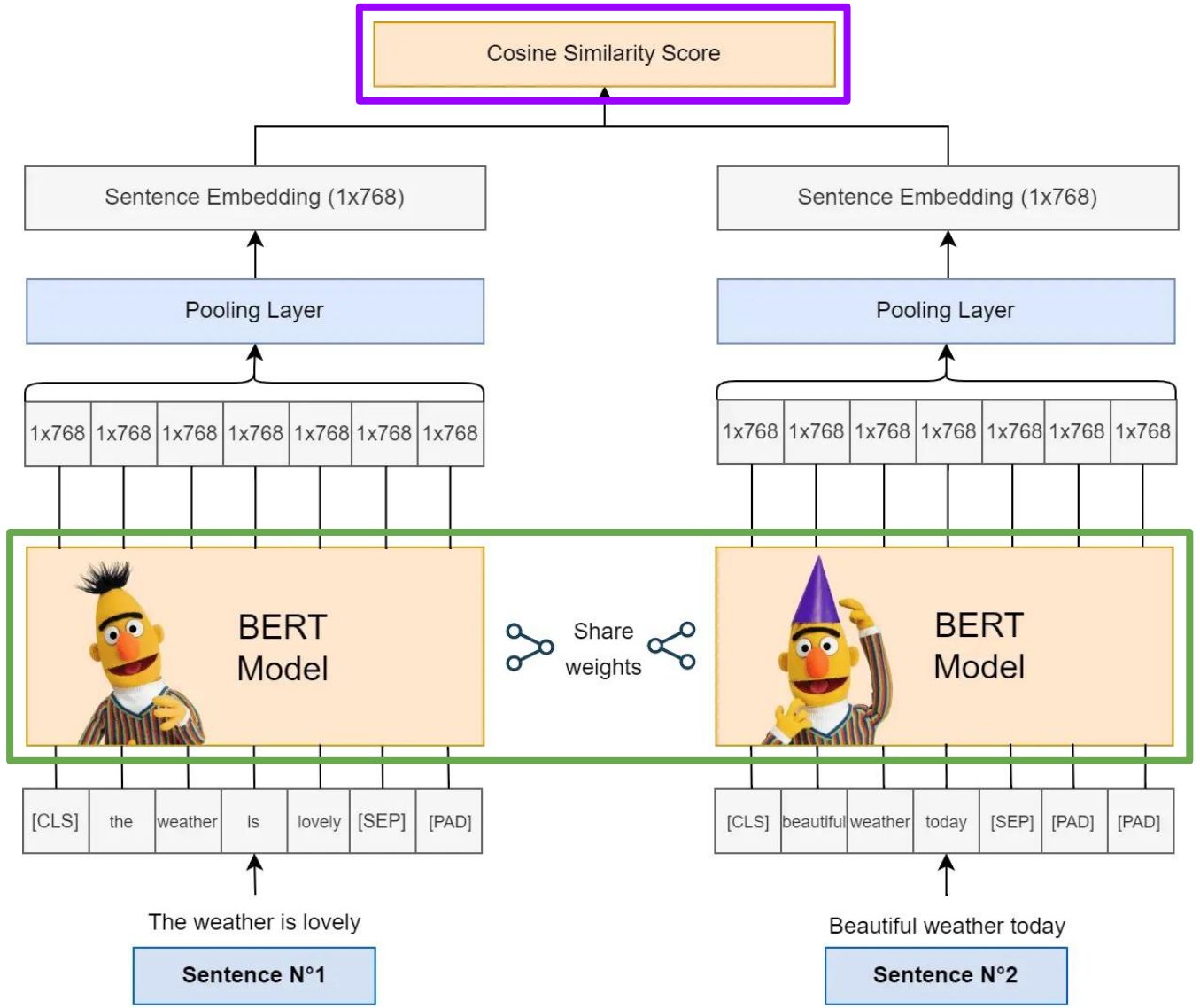
Loss function and models

Loss function

- **Cosine similarity loss:** tries to minimize the error between the predicted and actual similarity scores

Models

- **BERT**
- **DistilBERT:** smaller version of BERT
- **RoBERTa:** trained on more data and for more time than BERT
- **MiniLM:** BERT model pretrained on sentence similarity task (for comparison with the other BERT models)

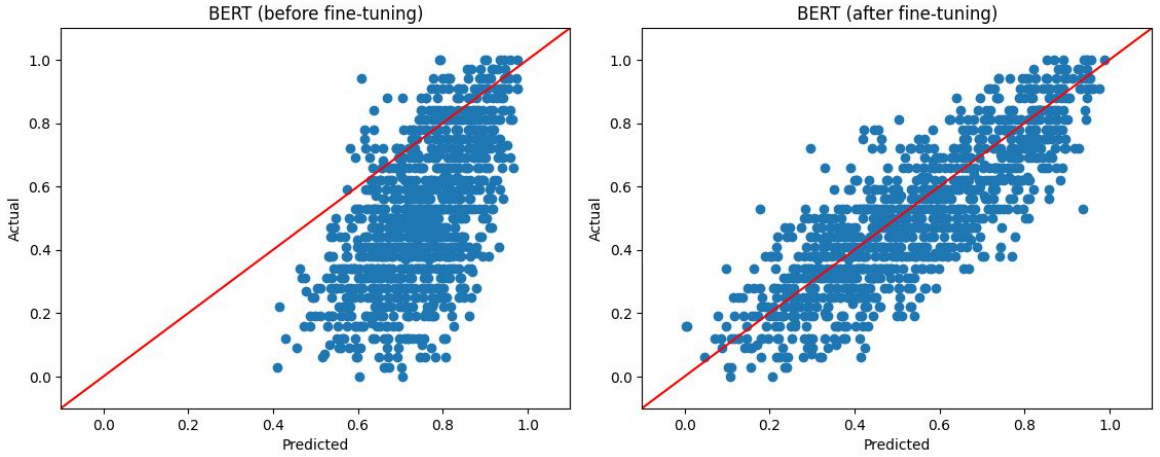


Source: [The Python Code]

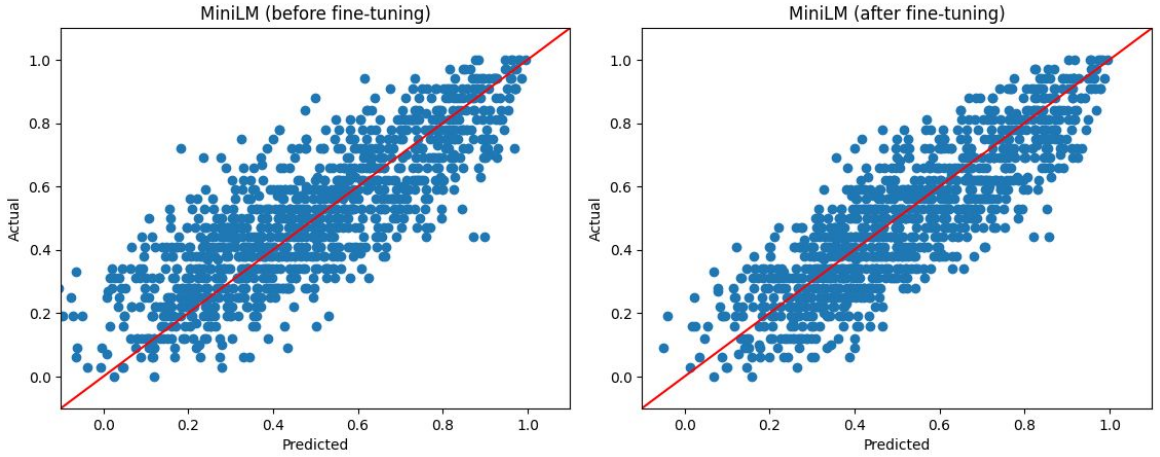
Fine-tuning the models

- Each model was trained for 10 epochs
  - MiniLM model (pretrained): around 40 minutes
  - BERT models (not pretrained): ranged from 2.5 ~ 5 hours
- Spearman's correlation between predicted and actual similarity scores **increased** for all models
  - Much bigger difference for BERT models compared to MiniLM

BERT



MiniLM



The red line represents a perfect prediction of the data

Results of fine-tuning

- MiniLM performed the best across all metrics
  - This makes sense, given that it was trained for sentence similarity tasks
- However, the fine-tuned BERT models have **similar performance**
  - This shows that BERT models can also learn this task, given the right model architecture

		BERT	DistilBERT	RoBERTa	MiniLM
Spearman's ρ	**Before training**	0.58619	0.62125	0.56774	0.82041
	After training	0.83162	0.82300	0.83659	0.85417
MSE	**Before training**	0.09184	0.13874	0.27248	0.02043
	After training	0.01671	0.01710	0.01650	0.01461
MAE	**Before training**	0.25653	0.32793	0.47468	0.11252
	After training	0.10302	0.10379	0.10154	0.09795