CentraleSupélec | université PARIS-SACLAY

Sentence-BERT

Idriss Mortadi, Elisa Faure, Marianne Brugidou, Saad Taharraoui, Abdellah Oumida

Sentence Embeddings using Siamese BERT-Networks

06/01/2025

OUTLINES

CentraleSupélec | université PARIS-SACLAY

INTRODUCTION

How to search for documents?

CentraleSupélec | université PARIS-SACLAY

INTRODUCTION

How to search for documents?

- Keywords ?
- TF-IDF embeddings ?
- Word2Vec ? Doc2Vec?
- …

RETRIEVAL PROBLEM !

CentraleSupélec | université PARIS-SACLAY

INTRODUCTION

The Problem: Traditional Retrieval Challenges

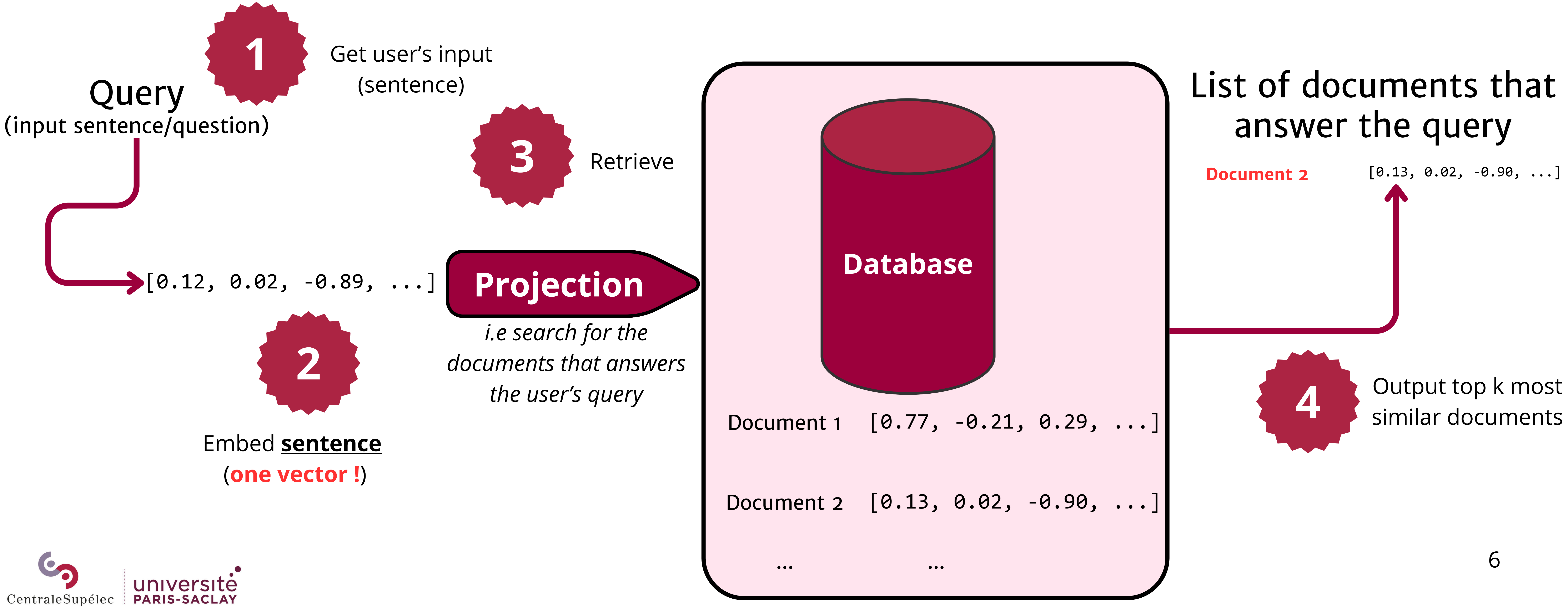
- **Inefficient Search**: Traditional methods struggle with understanding the semantic meaning of queries and documents.
- **Limited Context**: Keyword-based searches often miss the nuanced context and meaning.
- **High Computational Cost**: Processing each query-document pair is resource-intensive.

What is Retrieval?

- Process of finding and ranking relevant information/documents based on a user's query.
- Goal: Return the **most relevant results** to satisfy the user's information need.

INTRODUCTION

How it should work



INTRODUCTION

Limitation of BERT

Example of limitation

- Finding the most similar pair in 10,000 sentences:
- BERT: ~50 million computations (65 hours)

Why This Matters:

- Many real-world applications need fast sentence similarity:
  - Semantic search
  - Clustering
  - Information retrieval
  - Smart duplicate detection

INTRODUCTION

Problem Statement

- **<u>Objective</u>**: Effectively retrieve semantically similar sentences from a large corpus.
- **<u>Challenge</u>**: Traditional BERT models are not optimized for sentence-level retrieval tasks. BERT produces **contextualized word embeddings**, which are not directly comparable for **sentence-level semantic similarity**.
- **<u>Issue</u>**: Comparing sentence semantics using BERT embeddings is inefficient and ineffective for tasks like clustering, semantic search, and paraphrase identification.

Goal

- Modify BERT to produce semantically meaningful sentence embeddings.
- Be able to fine-tune the model on sentence-level tasks.
- Enable efficient comparison of sentence embeddings using cosine similarity.

INTRODUCTION

How S-BERT Proposes to Enhance Retrieval:

Sentence Embeddings

- Captures the semantic meaning of entire sentences, improving search accuracy.

Semantic Similarity

- Measures how similar sentences are, aiding in tasks like Textual Semantic Similarity (STS).

Information Extraction

- Facilitates semantic search, making it easier to find relevant information.

Use of optimized index structures*

- 10,000 sentence embeddings takes ~5 seconds, with S-BERT and computing cosine-similarity ~0.01 seconds.
  - Finding the most similar question can be reduced from 50 hours to a few milliseconds

**Johnson et al., 2017, Billion-scale similarity search with GPUs*

ARCHITECTURE & TRAINING

S-BERT

ARCHITECTURE

Solution:

- SBERT: Sentence-BERT with siamese network structure.
- Fine-tuned on tasks like Natural Language Inference (NLI) and paraphrase identification.
- Produces sentence embeddings that capture semantic meaning and are easily comparable.

How it Works

- Let **S1** and **S2** be two sentences.
- SBERT produces sentence embeddings **u** and **v** for **S1** and **S2**, respectively.
- The similarity between **S1** and **S2** is measured for example using cosine similarity:

$$\boxed{\text{similarity}(S_1, S_2) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}}$$

- Where:
  - **u** · **v** is the dot product of the embeddings.
  - **||u||** and **||v||** are the magnitudes (norms) of the embeddings.

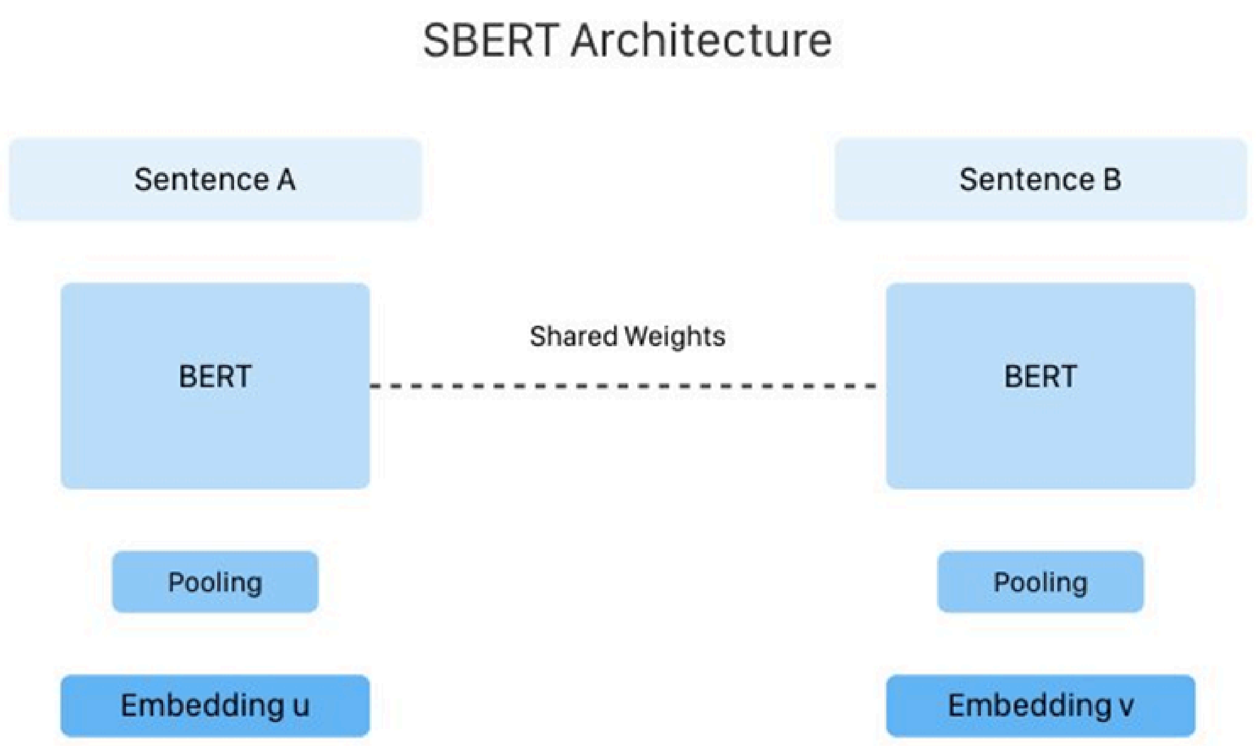
ARCHITECTURE

Two identical BERT networks with shared weights

Each network processes one sentence independently

Pooling strategies (of individual token embeddings) **to get fixed-size embeddings per sentence**:

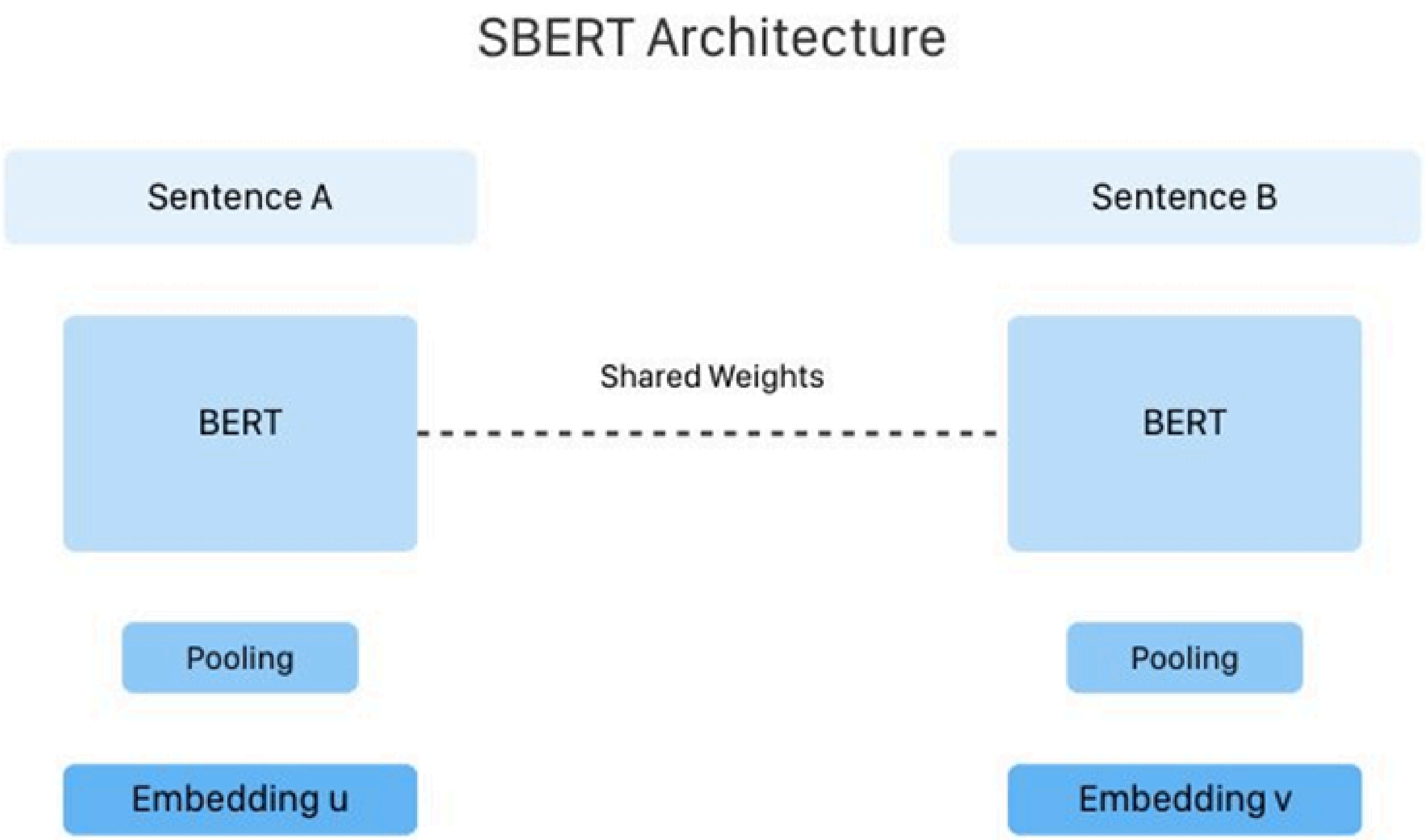
- MEAN pooling (default)
- MAX pooling
- CLS token



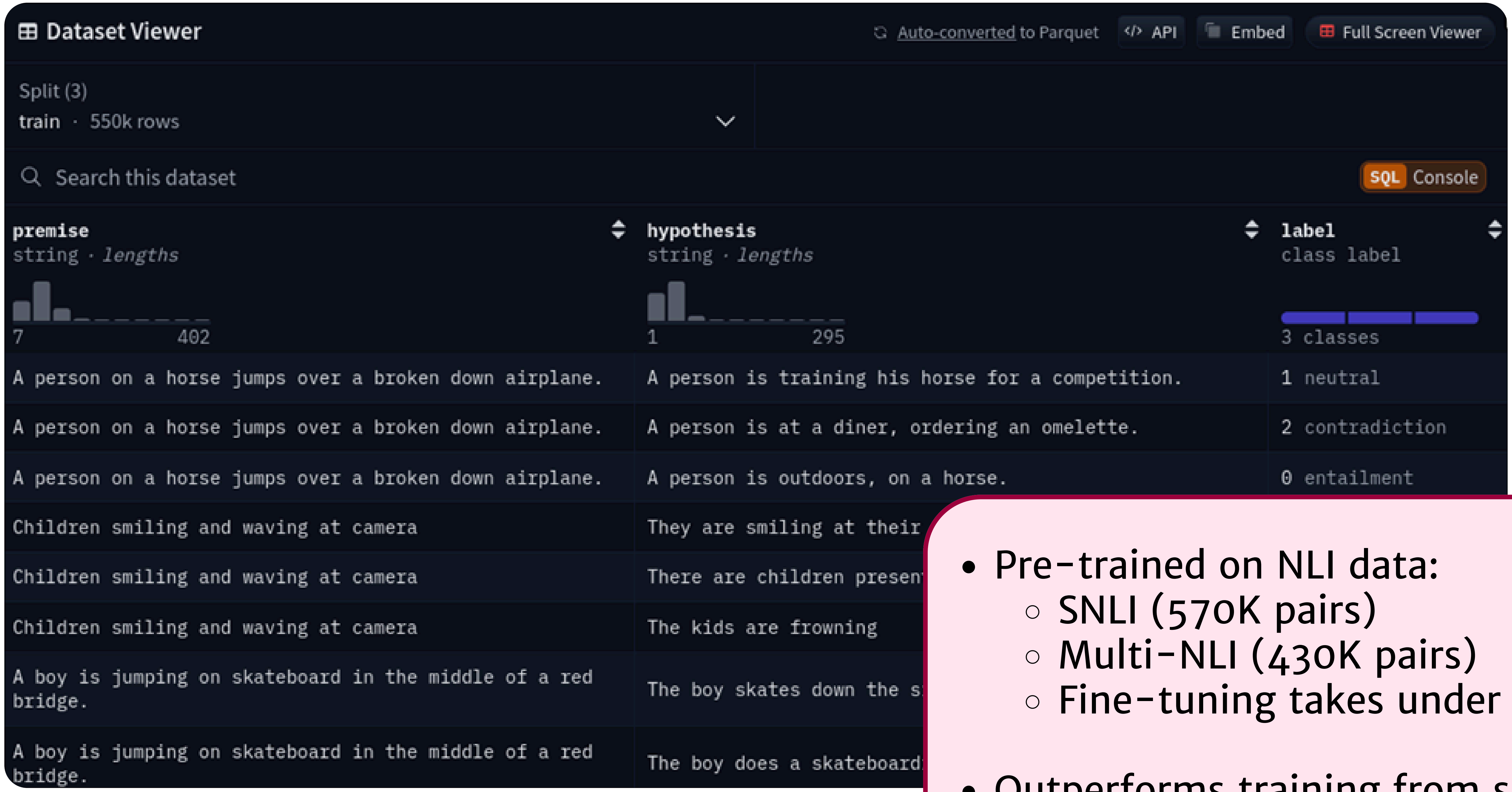
ARCHITECTURE

Siamese ?

- Siamese Network Structure:
    - A siamese network consists of two identical sub-networks that share weights.
- In SBERT, two identical BERT networks process two sentences simultaneously.
    - The shared weights ensure that the embeddings are in **the same semantic space**, making them directly comparable.



TRAINING DETAILS



premise string · *lengths* 7 402	hypothesis string · *lengths* 1 295	label class label 3 classes
A person on a horse jumps over a broken down airplane.	A person is training his horse for a competition.	1 neutral
A person on a horse jumps over a broken down airplane.	A person is at a diner, ordering an omelette.	2 contradiction
A person on a horse jumps over a broken down airplane.	A person is outdoors, on a horse.	0 entailment
Children smiling and waving at camera	They are smiling at their	
Children smiling and waving at camera	There are children present	
Children smiling and waving at camera	The kids are frowning	
A boy is jumping on skateboard in the middle of a red bridge.	The boy skates down the s	
A boy is jumping on skateboard in the middle of a red bridge.	The boy does a skateboard	

- Pre-trained on NLI data:
  - SNLI (570K pairs)
  - Multi-NLI (430K pairs)
  - Fine-tuning takes under 20 minutes
- Outperforms training from scratch

CentraleSupélec | université PARIS-SACLAY

THREE TRAINING OBJECTIVES

Classification (NLI datasets)

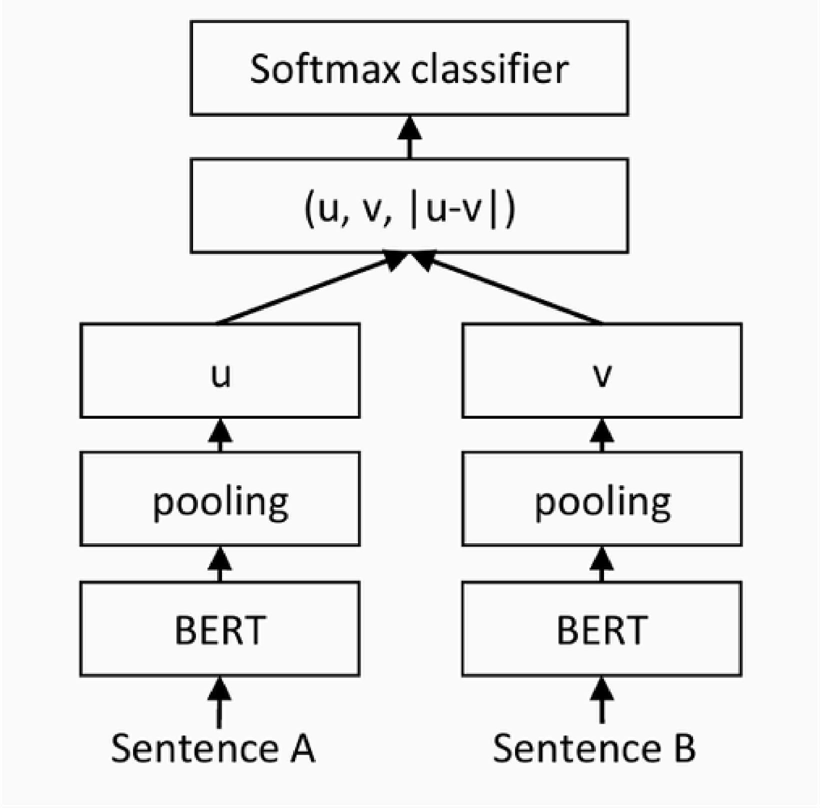


Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

Regression (STS data)

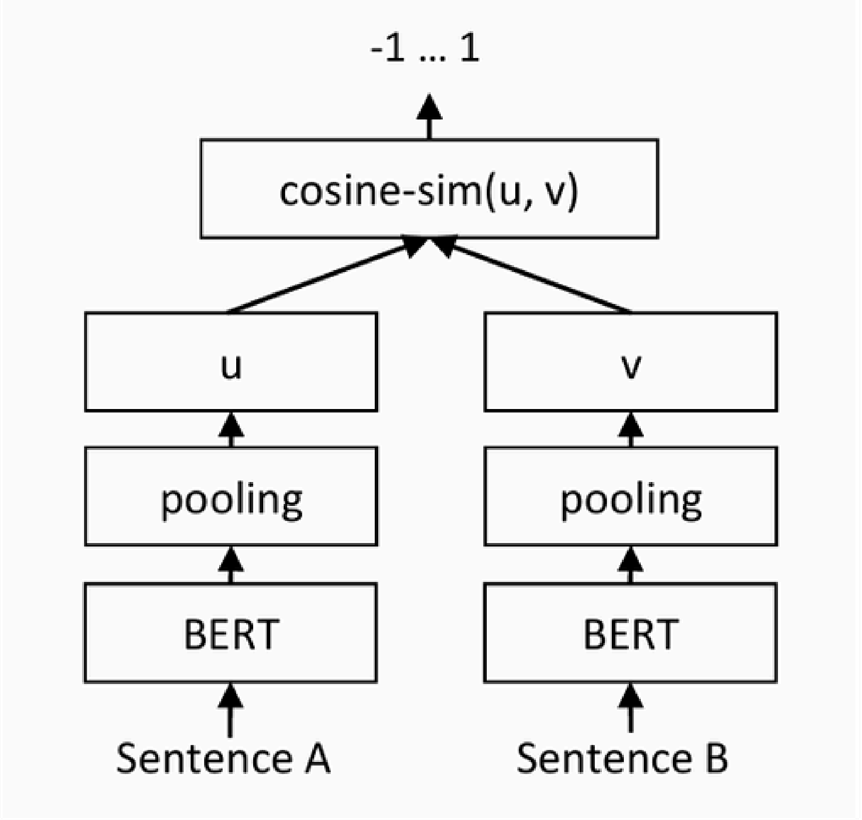


Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

THREE TRAINING OBJECTIVES

Classification (NLI datasets)
- Input: Sentence pairs (A,B)
- Output: entailment/contradiction/neutral
- Combines: embeddings u,v and |u-v|

$$o = \mathrm{softmax}(W_t(u, v, |u - v|))$$

- **u** and **v** being the embeddings of sentences **A** and **B** respectively.
- **o** classification output vector

Regression (STS data)
- Input: Sentence pairs
- Output: Similarity score
- Uses cosine similarity

$$\mathrm{similarity}(S_1, S_2) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$

Triplet (Online mining)
- Input: (anchor, positive, negative)
- Brings similar sentences closer
- Pushes different sentences apart

$$\mathcal{L}_{\text{triplet}} = \max(0, \|\mathbf{u}_A - \mathbf{u}_P\|^2 - \|\mathbf{u}_A - \mathbf{u}_N\|^2 + m)$$

$\mathbf{u}_A$ anchor sentence
$\mathbf{u}_P$ positive sentence
$\mathbf{u}_N$ negative sentence
m margin hyperparameter

EVALUATION

EVALUATION

Semantic Textual Similarity (STS)

- Unsupervised STS **(STS tasks 2012-2016, STSb, SICK)** => Regression Objective
- Supervised STS **(STSb)** => Cosine Similarity
- Argument Facet Similarity => Regression Objective
- Wikipedia Section Distinction => Triplet Objective

→ Quality of embeddings for similarity (clustering, search...)

SentEval

- Use output embeddings to train a logistic regression model on different classification tasks (sentiment prediction, subjectivity prediction, ...)

→ "Usefulness" of embeddings

EXPERIMENTAL RESULTS

EXPERIMENTAL RESULTS: 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.

EXPERIMENTAL RESULTS: 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.

KEY INNOVATIONS

KEY INNOVATIONS & BENEFITS

Smart batching strategy

Grouping sentences of similar lengths together in the same batch to minimize padding effects and reduce computational overhead

Embedding concatenation

Usefulness of concatenating [u, v, |u - v|] (ablation study)

Different training objectives

Adapts to different kinds of tasks

Powerful fine-tuning performance

S-BERT can be tuned in less than 20 minutes, while yielding better results than comparable sentence embedding methods.

CONCLUSION

Conclusion: Sentence-BERT and Beyond

A Milestone in NLP

- Revolutionized how we encode sentences.
- Enabled scalable real-world applications.

Limitations

- Restricted to 512-token sequences.
- Poor performance for large-scale tasks (e.g., code, legal text).
- Requires fine-tuning for domain-specific applications.

Conclusion: Sentence-BERT and Beyond

Looking Ahead with ModernBERT

- Addresses key limitations of traditional models with advanced features:
  - 8192-token support for longer contexts.
  - Optimized speed and memory with innovations like Flash Attention, Rotary Positional Embeddings and GeGLU layers.
  - Modern training data scales (2 trillion tokens) and mixtures (including code ande math data)

THANKS FOR YOUR ATTENTION