Sentence Embeddings

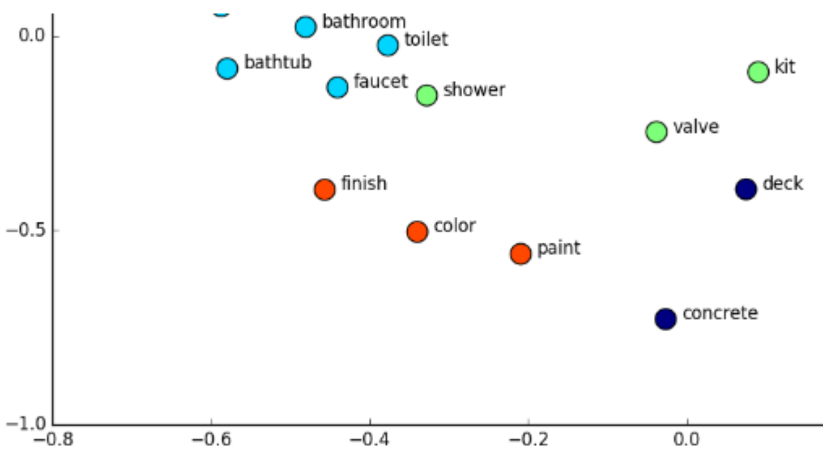
Understanding the Foundation of SBERT

What are Sentence Embeddings?

Sentence embeddings are numerical representations of sentences, capturing their meaning in a high-dimensional space.

Sentence	Numbers			
Hello, how are you?	0.419	1.28	...	-0.06
I'm going to scool today	-0.74	1.02	...	1.35
...			...	
Once upon a time	-0.82	-1.32	...	0.23

Word	Numbers			
A	-0.82	-0.32	...	-0.23
Aardvark	0.419	1.28	...	-0.06
...			...	
Zygote	-0.74	-1.02	...	1.35



https://txt.cohere.com/sentence-word-embedding

https://neptune.ai/blog/word-embeddings-guide

Role of Sentence Embeddings in NLP

Key Role in NLP:

- Sentence embeddings are crucial for understanding the meaning and context of text in NLP.
- They provide a way to quantify and compare whole sentences/documents beyond individual words.

NLP Tasks Enhanced by Sentence Embeddings:

- **Semantic textual similarity (STS)** — comparison of sentence pairs.
  - Finding Questions similar to the New Question
- **Semantic search** — information retrieval (IR) using semantic meaning. Given a set of sentences, we can search using a ‘query’ sentence and identify the most similar records. Enables search to be performed on concepts (rather than specific words).
- **Clustering** — we can cluster our sentences, which is useful for topic modeling.

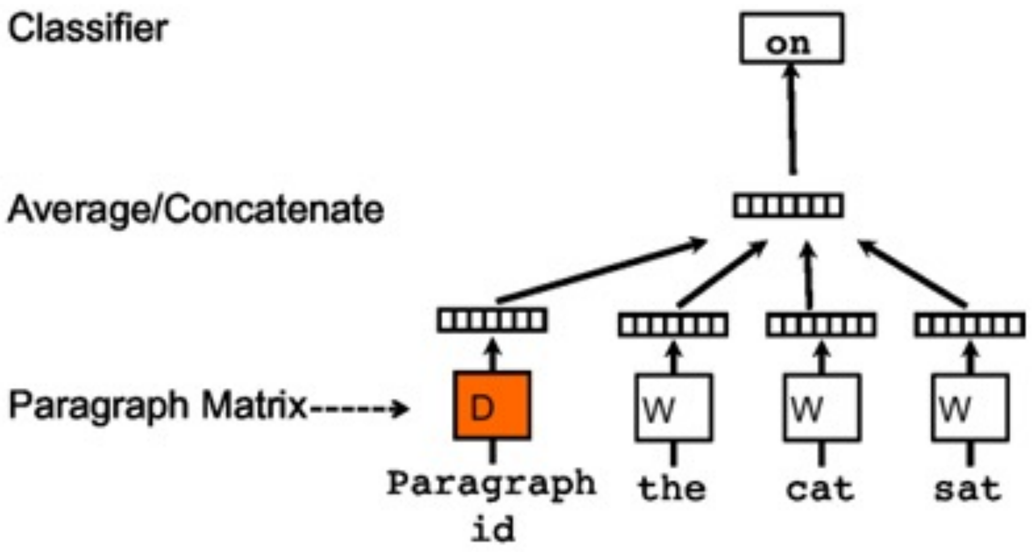
W2VEC to Sentence Embeddings

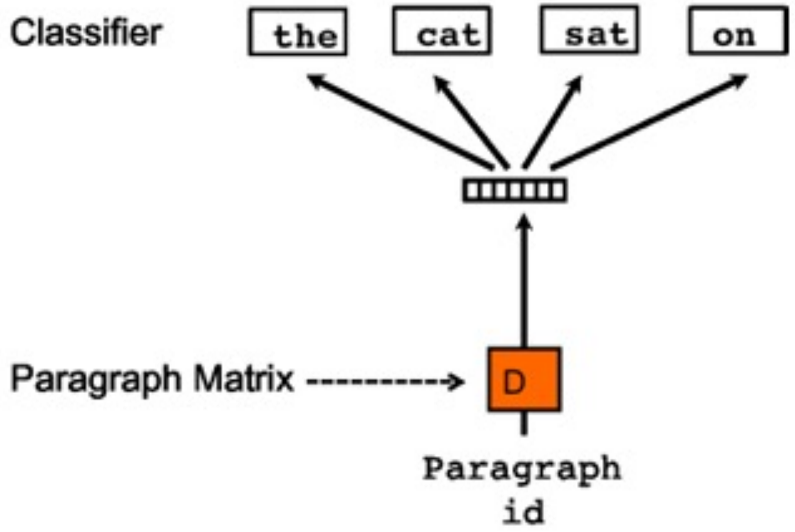
- No: [1,0,0,0]
- I: [0,2,0,0]
- Am: [-1,0,1,0]
- Good: [0,0,1,3]

- “No, I am good!” – [0,2,2,3]
- “I am No good!” – [0,2,2,3]

https://txt.cohere.com/sentence-word-embedding

Doc2Vec

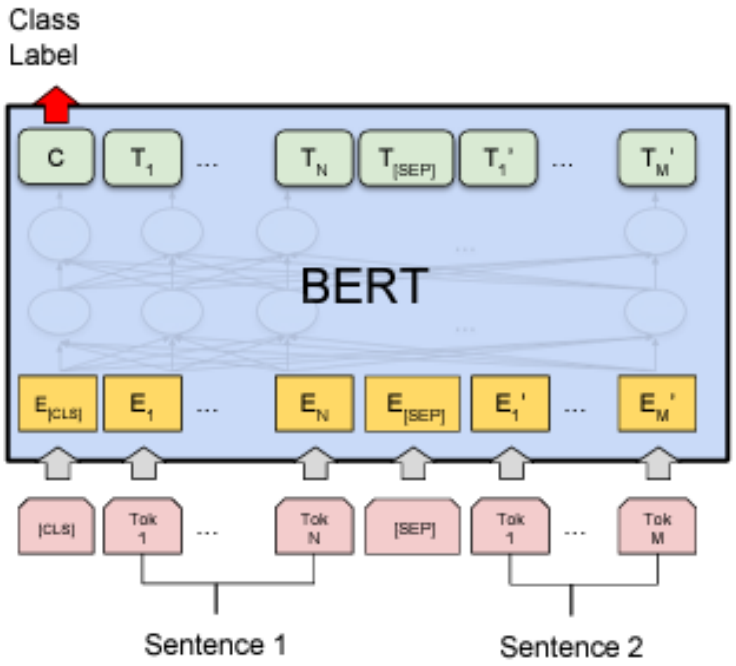




https://arxiv.org/abs/1405.4053

Cross-Encoder

BERT for Sentence Similarity



Cross-Encoder architecture

Unsuitable for pair regression tasks

- Finding existent questions on Quora most similar to new Question
- Finding in a collection of n=10000 sentences the pair with the highest similarity requires with BERT n·(n−1)/2=49995000 inference computations

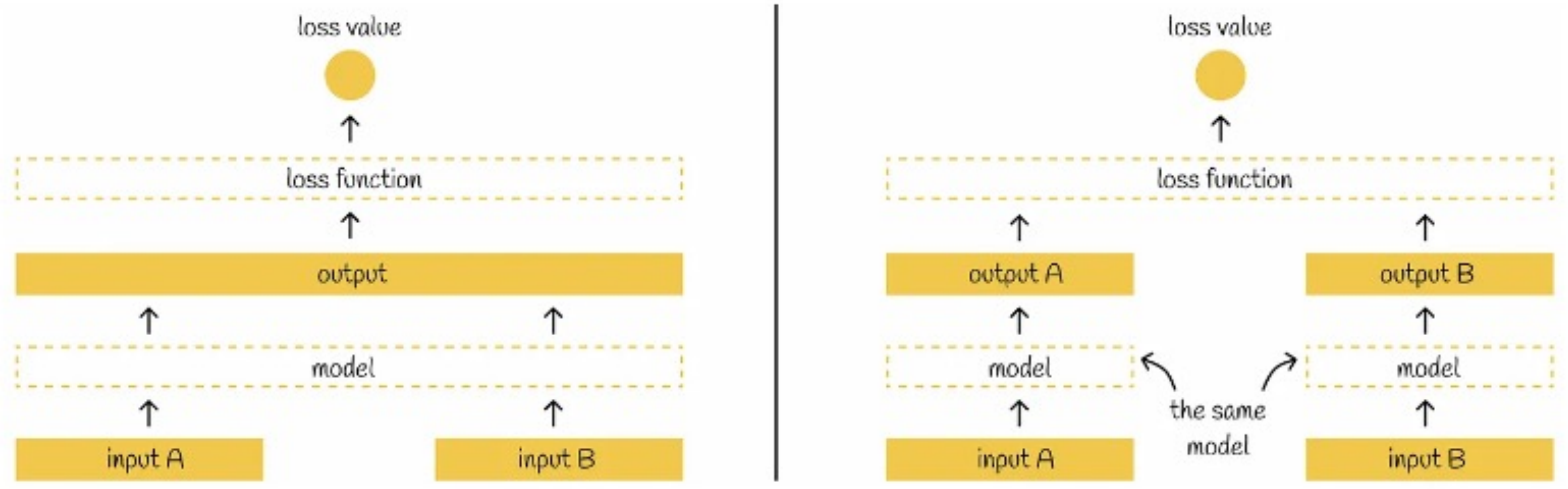
Possible Solution – Use BERT to get sentence Embeddings

- Use the CLS token as sentence representation
- Average The BERT output layer

- Problem
  - Bad sentence embeddings, often worse than averaging GloVe embeddings

Siamese Network (Bi-Encoder)

Siamese Network



Non-Siamese Network – Cross Encoder

Siamese Network – Bi-Encoder

Figure from : https://towardsdatascience.com/sbert-deb3d4aef8a4

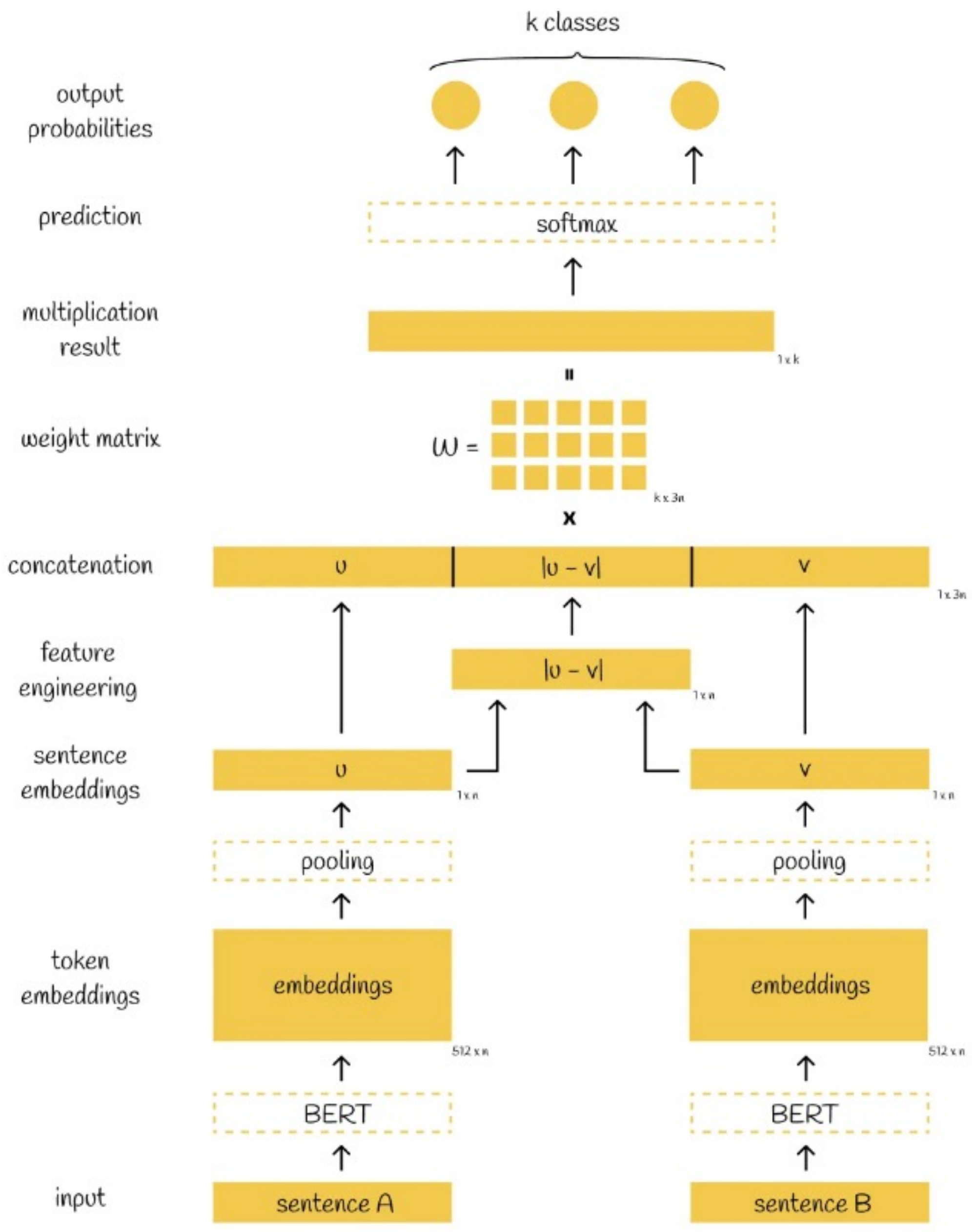
Siamese Network- Cross Entropy

	NLI
Pooling Strategy	
MEAN	**80.78**
MAX	79.07
CLS	79.80
Concatenation	
(u, v)	66.04
$(\|u - v\|)$	69.78
$(u * v)$	70.54
$(\|u - v\|, u * v)$	78.37
$(u, v, u * v)$	77.44
$(u, v, \|u - v\|)$	**80.78**
$(u, v, \|u - v\|, u * v)$	80.44

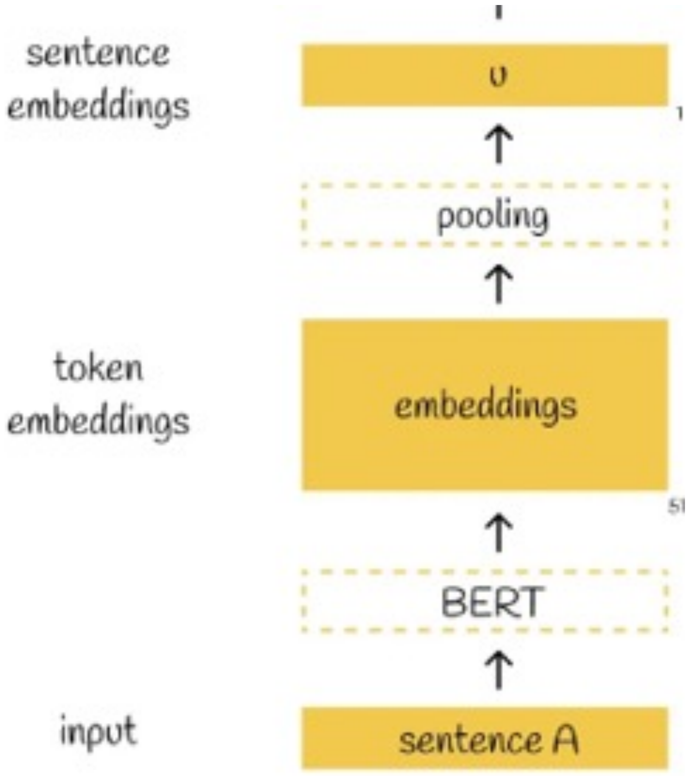
K = 3 for NLI dataset

Loss function = Cross Entropy Loss function

Sentence A (Premise)	Sentence B (Hypothesis)	Label
A soccer game with multiple males playing.	Some men are playing a sport.	Entailment
A black race car starts up in front of a crowd.	A man is driving down a lonely road.	Contradiction
A smiling costumed woman is holding an umbrella.	A happy woman in a fairy costume holds an umbrella.	Neutral



Siamese Network - Inference



- Finding in a collection of n=10000 sentences the pair with the highest similarity requires with BERT (**cross-encoder**) $n\cdot(n-1)/2=49995000$ inference computations.

- On V100 GPU, the most similar sentence pair in a collection of 10,000 sentences is reduced from 65 hours with BERT to the computation of 10,000 sentence embeddings (~5 seconds with SBERT).

- If your data is structured as sentence pairs and the objective is to find the similarity between only the sentence pairs, BERT will give better results than SBERT.

BERT vs SBERT Applications

BERT (cross-encoder) Application Examples

- **Customer Support Automation:** BERT can be used to understand customer queries by comparing them to a fixed small set of known questions and providing precise answers, where the volume of queries is not excessively high but accuracy is crucial.

- **Legal Document Analysis:** For comparing specific pairs of clauses or sentences in contracts to ensure compliance or detect anomalies, where the precision of the context understanding is paramount.

- **Quality Control in Manufacturing**: BERT can analyze reports or descriptions of defects, matching them with known issues to provide accurate assessments where the dataset of known issues is finite and manageable.

SBERT (Bi-encoder) Application Examples

- **Semantic Search Engines**: SBERT can power search engines that quickly sift through large databases of documents to find the most relevant content based on semantic similarity to the search query.

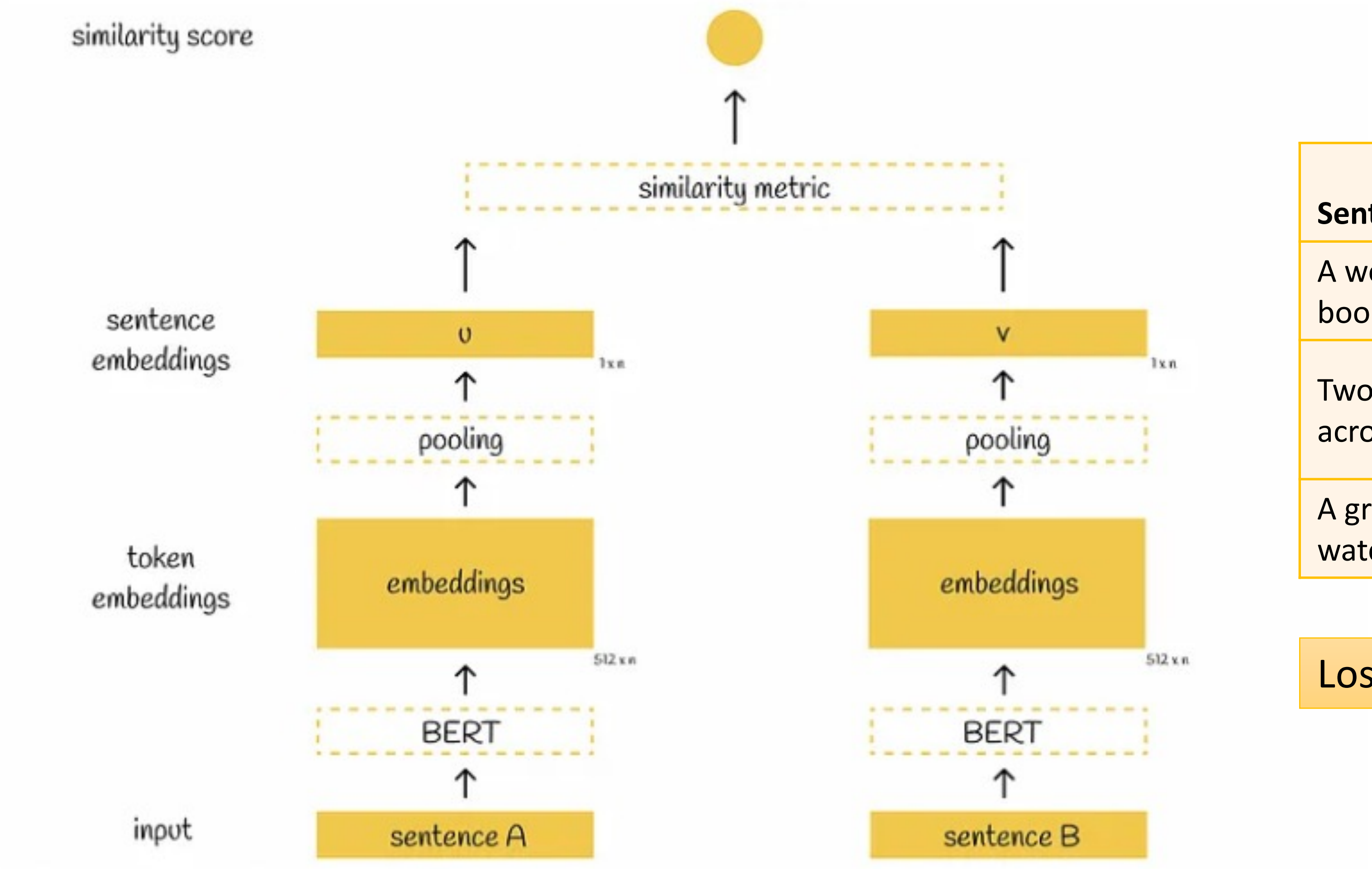
- **Content Recommendation Systems**: SBERT can be used to match users with relevant articles, products, or services by quickly comparing user preferences or profiles with a large inventory of items.

- **Data Deduplication**: In large databases, SBERT can efficiently identify and group similar entries, which is useful for cleaning up and organizing data in CRM systems or product catalogs.

- **Real-time Social Media Monitoring**: For businesses that need to monitor brand mentions across social media platforms, SBERT can quickly process large volumes of data to find and categorize relevant posts.

Other Loss Functions for Siamese Network

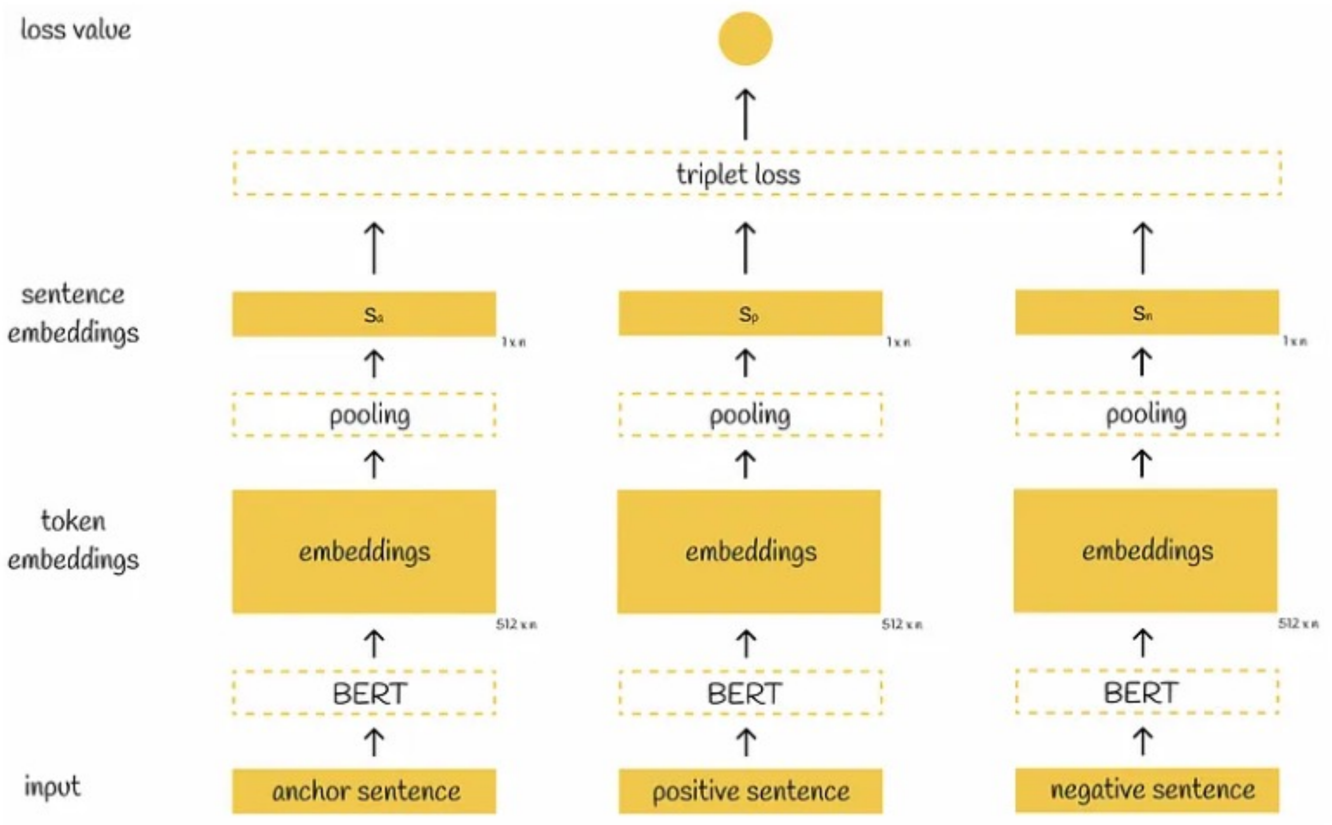
Siamese Network- Regression



Sentence A	Sentence B	Similarity Score
A woman is reading her book in the park.	A lady is enjoying a novel outdoors.	4.2
Two dogs are running across the field.	A pair of canines are sprinting through a meadow.	3.8
A group of people is watching a movie.	Some individuals are viewing a film together.	3.5

Loss function = MSE

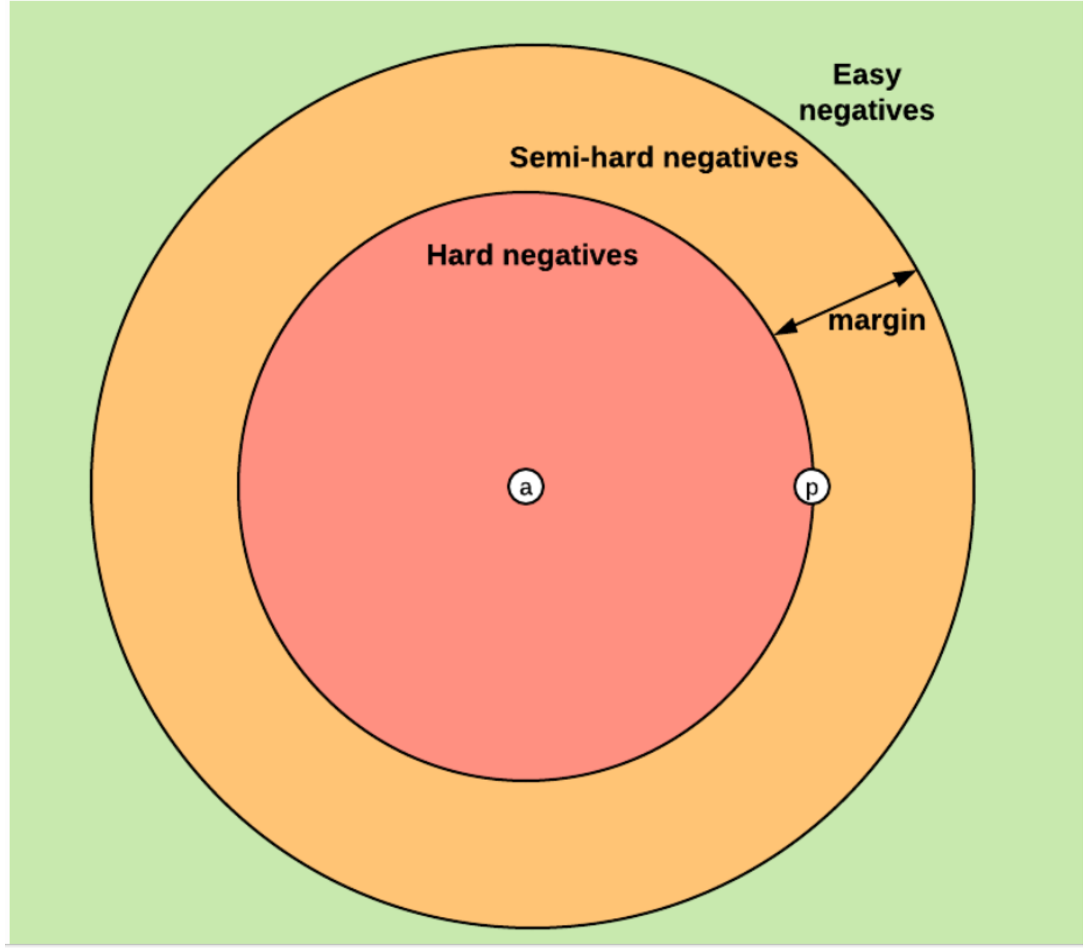
Triplet Network



$$max(||s_a - s_p|| - ||s_a - s_n|| + \epsilon, 0)$$

with s_x the sentence embedding for $a/n/p$, $||\cdot||$ a distance metric and margin ϵ. Margin ϵ ensures that s_p is at least ϵ closer to s_a than s_n. As metric we use Euclidean distance and we set $\epsilon = 1$ in our experiments.

Triplet Mining



- **easy triplets**: triplets which have a loss of 0, because $d(a,p) + margin < d(a,n)$
- **hard triplets**: triplets where the negative is closer to the anchor than the positive, i.e. $d(a,n) < d(a,p)$
- **semi-hard triplets**: triplets where the negative is not closer to the anchor than the positive, but which still have positive loss: $d(a,p) < d(a,n) < d(a,p) + margin$

https://omoindrot.github.io/triplet-loss

What if we do not have negatives?

Duplicate Question Answer Dataset

What is your name?	What's your name?
Can you help me?	Could you assist me?
Where is the library?	Where can I find the library?
When does the class start?	What time does the class begin?

Duplicates

Non-Duplicates

Nion-Duplicates

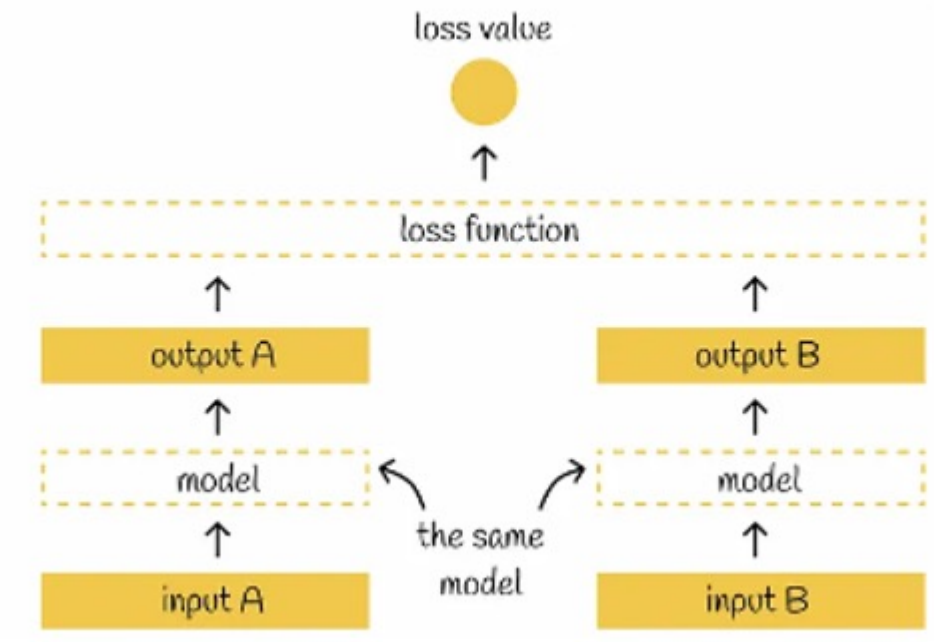
Sentence A

Sentence B

Multiple Negatives Ranking Loss

What is your name?	What's your name?
Can you help me?	Could you assist me?
Where is the library?	Where can I find the library?
When does the class start?	What time does the class begin?

Sentence A　　　　　Sentence B



Similarity Score

0.6	0.3	0.2	0.1
-0.8	0.5	0.1	0.3
- 0.3	-0.5	-0.1	-0.7
0.6	-0.2	0.1	1.0

Diagonals are - (Anchor, Positive)
Non-diagonals are (Anchor Negative)

- Treat this as multiclass classification problem with labels as [0, 1 ,2 ,3]