trec.nist.gov

EE-608
Deep Learning for Natural Language Processing

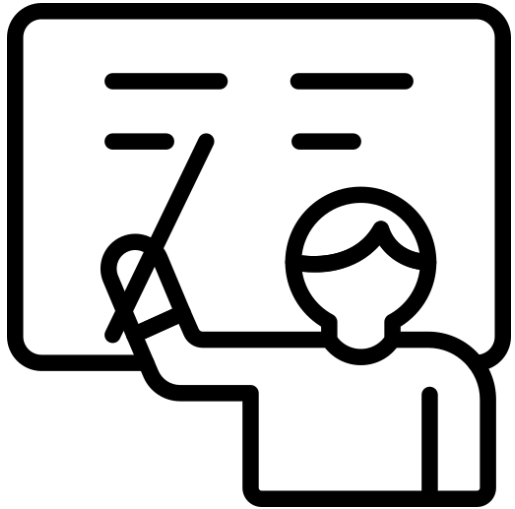
Final Project Presentation

Multi-modal retrieval with smooth weighting of negatives

Aleksandr Timofeev
Julia Majkowska
Paweł Młyniec
Sofia Blinova

Presentation plan

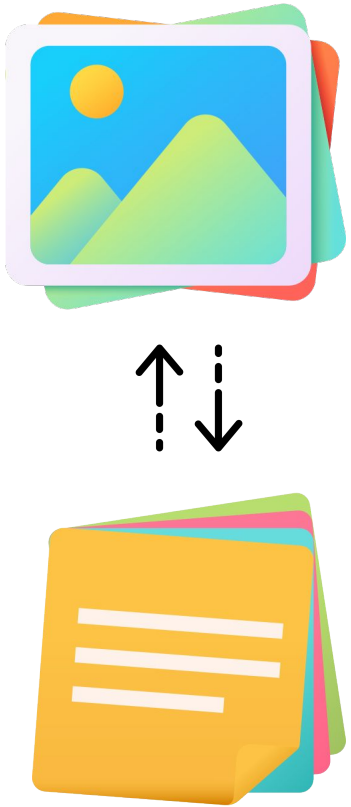
- Problem Statement and Motivation
- Contribution
- Our Approach
- Implementation
- Testing
- Experiments
- Results
- Possible Extensions
- Conclusions



Problem Statement and Motivation

- Multi-modal retrieval task aims to find the most similar images to text queries and vice versa.

- Multi-modal retrieval models are used in search engines for databases containing multiple modalities, recommender systems, matching tasks, etc.



Contribution

- The standard loss for training multimodal retrieval model is the maximal triplet loss [1] [2] [3]
- There is little research exploring the influence of different loss functions on the model performance
- We explore SimCLR [4] and BarlowTwins [5] loss functions
- Both those approaches exhibited stellar performance in computer vision
- We adapt these losses for the multi-modal setting
- There are different options for model architectures [6] [7]. We separate image and text branches which are represented by DistillBERT [8] and ResNet32 [9].

Problem of the maximal triplet loss

$$\ell_{MH}(i,c) = \max_{c'} \left[\alpha + s(i,c') - s(i,c)\right]_+ + \max_{i'} \left[\alpha + s(i',c) - s(i,c)\right]_+$$

i,c *is a positive image-caption pair;* ***i', c'*** *denote negative samples;*

s(i, c) *is a similarity function;* α *is a margin*

- This is an adaptation of a hinge loss which is used in SVM
- Better performance than summation over negative samples
- It poses several problems as an objective in DL:
  - Harder to optimize
  - MAX squeezes information about other samples
  - Not robust to outliers
  - Batch maximum is a biased estimator of the original loss.

Our approach

- We introduce a smoothed version of this loss based on SimCLR work:

$$\ell_{SimCLR}(i,c) = -\log \frac{\exp(s(i,c)/\tau)}{\sum_{c' \in C} \mathbb{1}[c' \neq c] \exp(s(i,c')/\tau)} - \log \frac{\exp(s(i,c)/\tau)}{\sum_{i' \in I} \mathbb{1}[i' \neq i] \exp(s(i',c)/\tau)}$$

- It is symmetrized by adding another component
- It does not have any problems mentioned before

Our approach

- BarlowTwins demonstrate better results than SimCLR in computer vision
- It computes a cross-correlation matrix of batch-normalized embeddings and apply the following loss:

$$\mathcal{L}_{\mathcal{BT}} \triangleq \sum_{i}(1-\mathcal{C}_{ii})^2 + \lambda \sum_{i}\sum_{j\neq i}\mathcal{C}_{ij}{}^2$$

Model architecture

Positive sample:
a black and white zebra eating some green grass

Negative sample:
a man staring intently at an oversized carrot.

Preprocess

Image network: ResNet32

Projection layer

Tokenizer

Text network: DistilBert

Projection layer

Common embedding space: Computing similarity and loss function

Data preprocessing

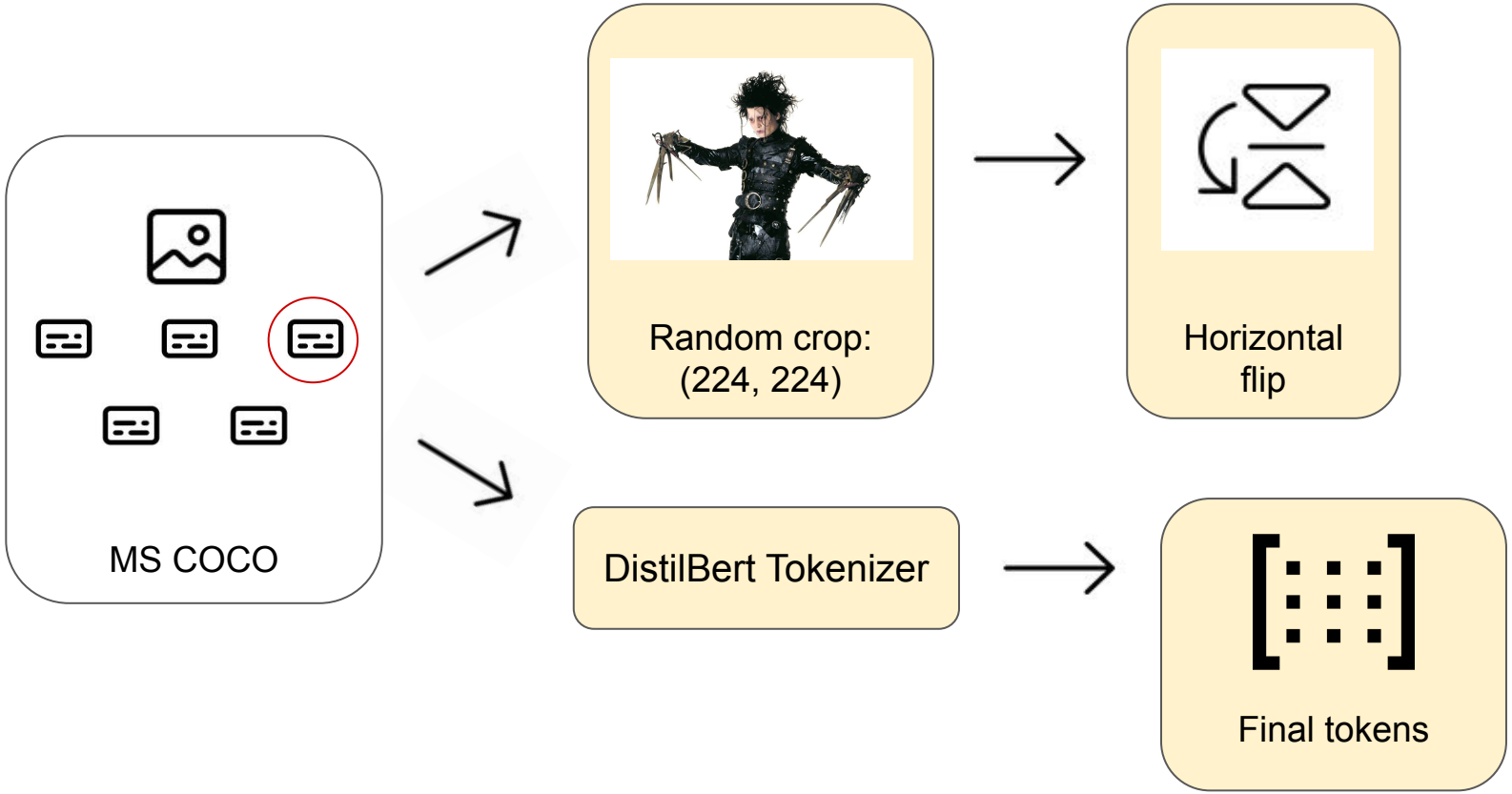
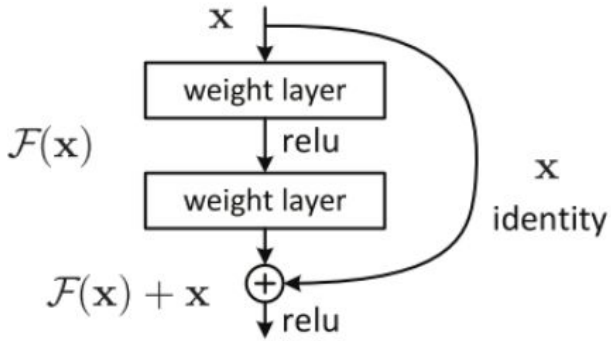
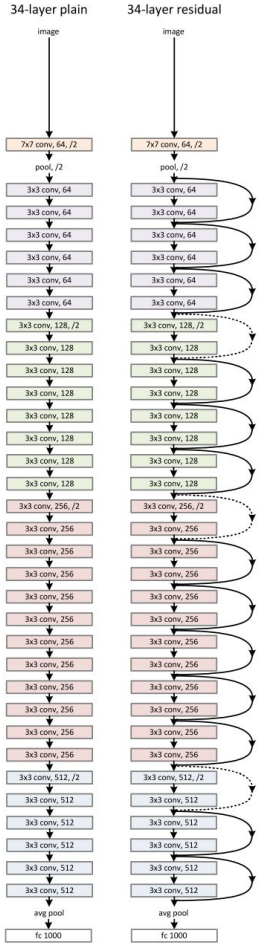


Image network

The core idea of ResNet is introducing a so-called “identity shortcut connection” that skips one or more layers, as shown in the following figure:



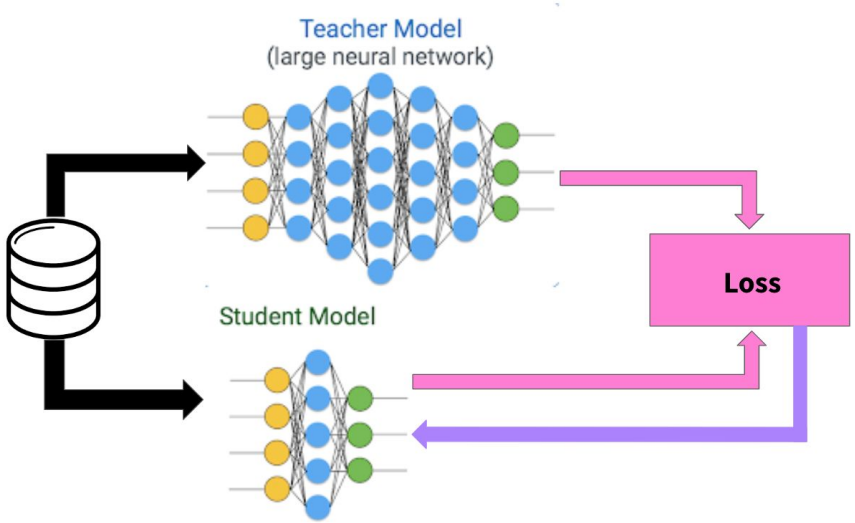
Projection layer: remove classification head and add fully-connected layer (512, 128)




Google

Distillation process

- *Knowledge distillation* is a compression technique in which a small model is trained to reproduce the behavior of a larger model.
- In the teacher-student training, it's trained a student network to mimic the full output distribution of the teacher network (its knowledge).
- Kullback-Leibler loss: $KL(p||q) = \mathbb{E}_p(log(\frac{p}{q})) = \sum_i p_i * log(p_i) - \sum_i p_i * log(q_i)$



DistilBERT

DistilBERT is a small version of BERT in which were *removed the token-type embeddings and the pooler* (used for the next sentence classification task) and were kept the rest of the architecture identical while reducing the numbers of layers by a factor of two.

Table 1: **DistilBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Projection layer: fully-connected layer (768, 128)

Training process

Positive sample:
a black and white zebra eating some green grass

Negative sample:
a man staring intently at an oversized carrot.

Preprocess

Image network: ResNet32

Projection layer

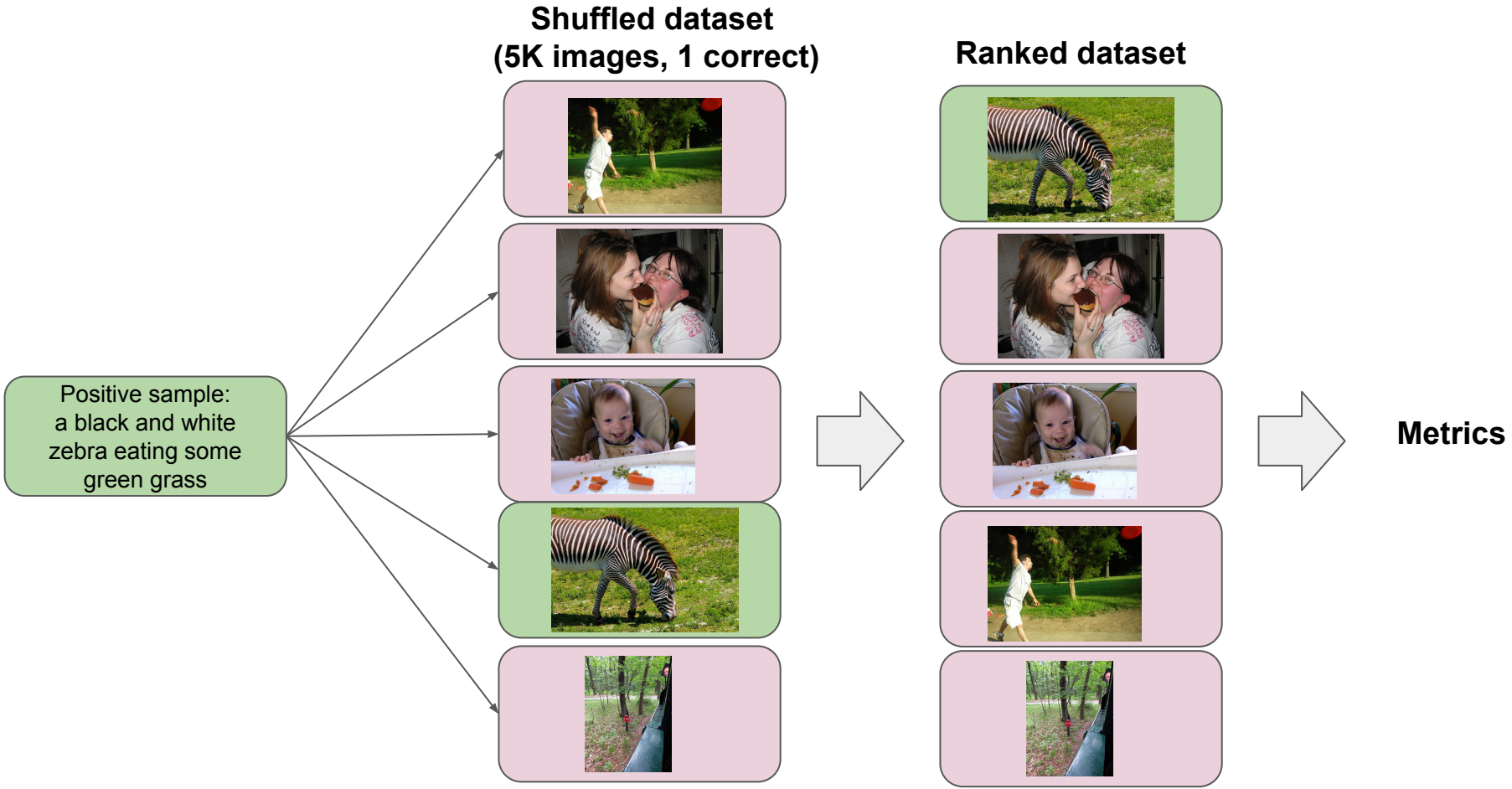
Tokenizer

Text network: DistilBert

Projection layer

Common embedding space: Computing similarity and loss function

Text to image retrieval testing



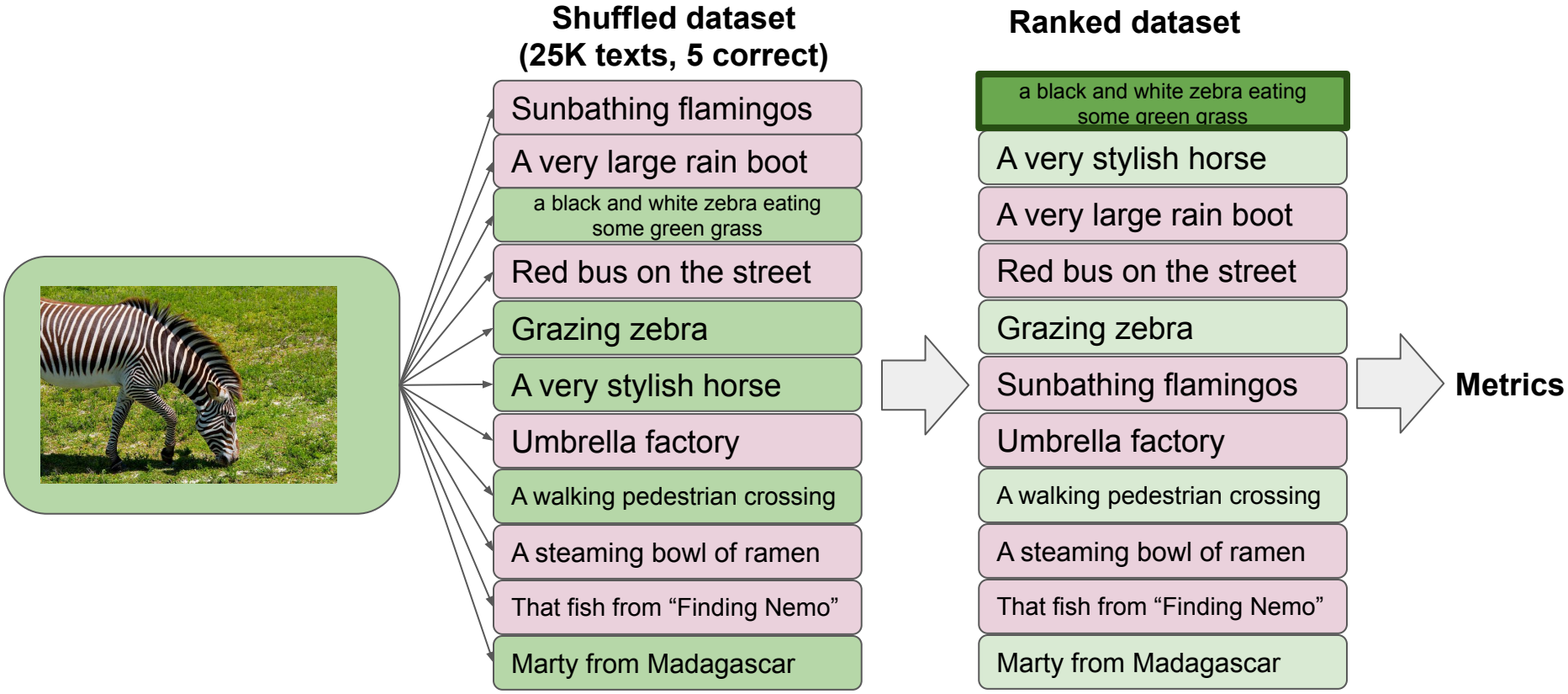
Metrics

$$t2i\_recall@K = \frac{|\{i,j{:}rank(text_{i,j},image_i)<K\}|}{N}$$

$$t2i\_mean\_rank = \frac{\sum_{i,j} rank(text_{i,j},image_i)}{N}$$

$$i2t\_med\_rank = median_{i,j}(rank(image_i, text_{i,j}))$$

Image to text retrieval testing



Metrics

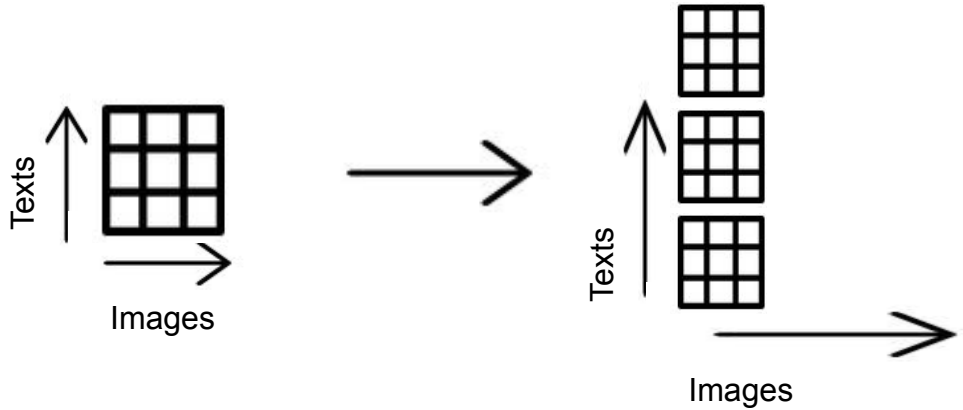
$$i2t\_recall@K = \frac{|\{i : min_j(rank(image_i, text_{i,j})) < K\}|}{N}$$

$$i2t\_mean\_rank = \frac{\sum_i min_j(rank(image_i, text_{i,j}))}{N}$$

$$i2t\_med\_rank = median_i(min_j(rank(image_i, text_{i,j})))$$

The main problems we faced

- Long training time
  - Figured out that the bottleneck of the training process was in the data loading and fixed it



The main problems we faced

- Long training time
  - Figured out that the bottleneck of the training process was in the data loading and fixed it
- Overfitting and low metrics
  - MS COCO contains 5 captions per image => randomly choose one. Input batch: 256 images and captions.
  - Optimizing hyperparameters
  - MultiStepLR scheduler with reduction by 0.5 on 1,4,10 epochs performs better than CosineAnnealingLR and GradualWarmup
  - Training the whole model is better than training only projections or only one branch
  - Losses parameters: margin = 0.2 for maximum triplet loss and temperature = 0.07 for SimCLR

Results

Loss	Image-to-text					Text-to-image				
	r@1↑	r@5↑	r@10↑	Mr↓	Med r↓	r@1↑	r@5↑	r@10↑	Mr↓	Med r↓
Sum Triplet	16.5	39.0	52.5	**41.7**	9.0	13.2	36.0	50.3	**37.9**	10.0
Max Triplet	15.9	39.4	52.4	42.8	9.0	12.8	34.5	49.0	39.0	11.0
SimCLR	**19.7**	**43.7**	**56.4**	55.8	**7.0**	**15.7**	**39.8**	**54.2**	58.5	**9.0**
Barlow Twins	2.0	7.6	12.0	397.7	111.0	1.9	7.9	13.6	154.9	62.0

Examples

"Baby elephant plays in the water"

Examples

"Big hamster plays on the grass"

Predicted

- **The elephants are standing beside each other near the water.**
- A very big pretty elephant laying down in the water.
- A young elephant walks with adult elephants along a dirt path
- A small baby elephant walking with other larger elephants
- **A group of elephants walking in muddy water**

Original

- A herd of elephants walking through a lake filled with water.
- A family of elephants washing up at a watering hole
- **A group of elephants walking in muddy water**
- Group of elephants walking in muddy water today.
- **The elephants are standing beside each other near the water**

Examples

Closeup of a brown bear sitting in a grassy area

A large bear that is sitting on grass.

A big burly grizzly bear is show with grass in the background

A black bear stands in the wild amongst dead grass

A zeebra running on a grass feild in a park

Possible extensions

- Apply our approach to finetune a SOTA multi-modal retrieval baseline model.
- Experiment with optimizable loss parameters for loss functions
- Test our model on different datasets like Flickr 30K
- Test our approach with a multilingual model [10][11].
- Add image-to-image and caption-to-caption similarity on augmented data [12][13]

Conclusion

- The new (in the domain) SimCLR loss showed much better performance
- Smooth loss functions are easier for optimization
- Our method is able to design an embedding space where texts and images with similar sense are close to each other
- Transformers may not show the best results in a combination with inappropriate loss functions
- Choice of right hyper-parameters significantly affect the final performance

Q&A