Exploring the Impact of Negative Samples of Contrastive Learning: A Case Study of Sentence Embedding

Rui Cao[1,*] , Yihao Wang[1,*] , Yuxin Liang[1]

Ling Gao[1,†] , Jie Zheng[1] , Jie Ren[2] , Zheng Wang[3]

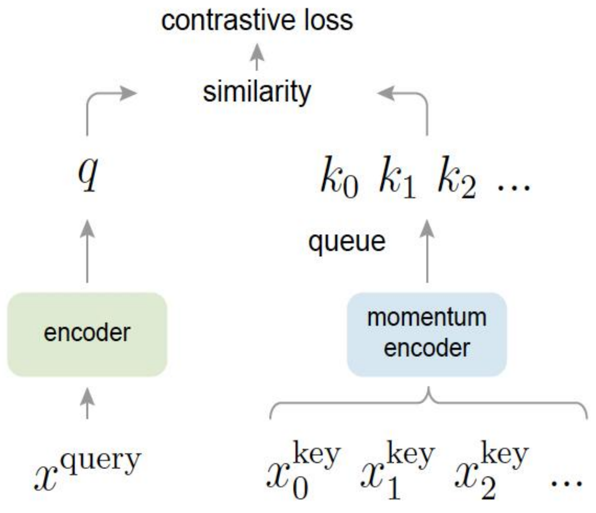
1. Northwest University
2. Shaanxi Normal University
3. University of Leeds

{caorui, wangyihao, liangyuxin}@stumail.nwu.edu.cn
{gl, jzheng}@nwu.edu.cn, renjie@snnu.edu.cn, z.wang5@leeds.ac.uk

* Authors contributed equally to this work.
† Corresponding author.

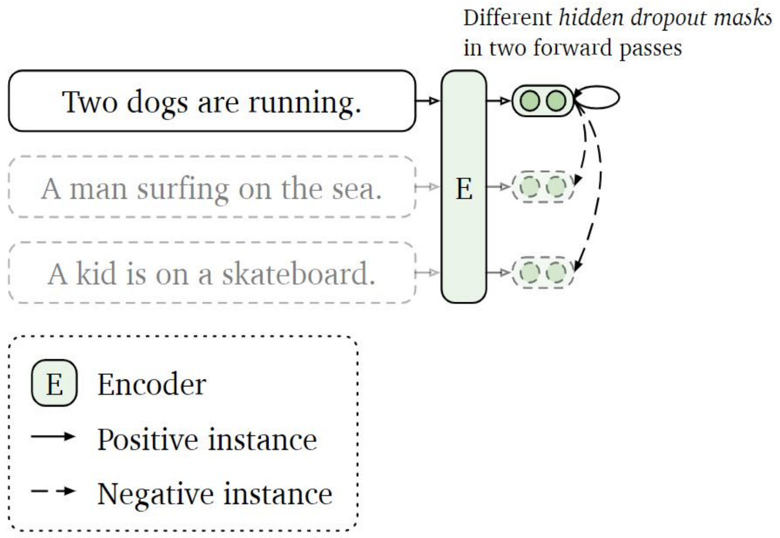
1. Introduction

Contrastive Learning is on 🔥 … bringing similar items closer and pushing dissimilar items away



MoCo[1]

- negative sample queue
- momentum encoder
- add MLP layer in ver.2
- cosine learning rate schedule


Different *hidden dropout masks* in two forward passes
Two dogs are running.
A man surfing on the sea.
A kid is on a skateboard.
E
E Encoder
Positive instance
Negative instance

SimCSE[2]

- dropout mask twice as positive pair
- supervised and unsupervised
- balanced alignment and uniformity

Motivation: Can we apply MoCo-style contrastive structure on unsupervised textual tasks?
If could:
1. How to boost performance?
2. How to pervent model from clasping?
3. How much negative sample do textual task needs?

[1] He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9729-9738.
[2] Gao T, Yao X, Chen D. SimCSE: Simple Contrastive Learning of Sentence Embeddings[C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021: 6894-6910.

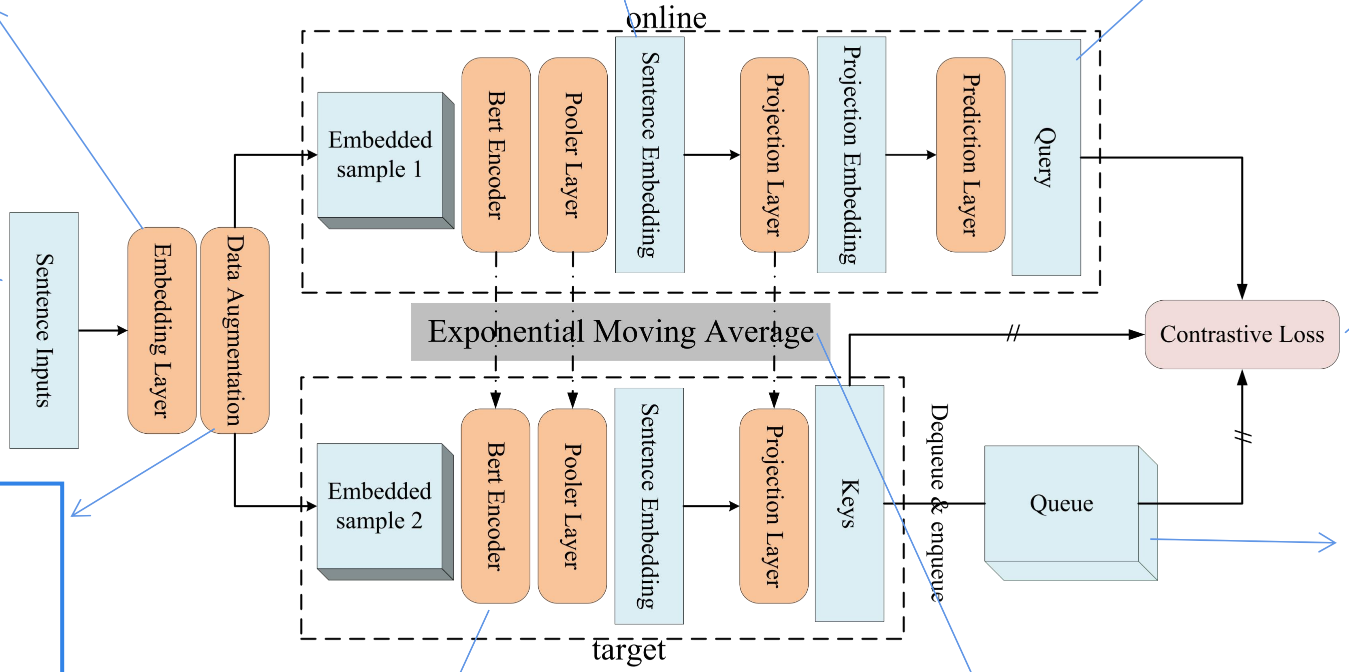
2. Method

Loading pre-trained BERT model from Huggingface*, use [CLS] token as sentence embedding

For evaluation ustage.

Pooler layer and Projection Layer are identical.
Prediction Layer is basically a MLP layer with Tanh activation.
We combine the structure of projection and prediction with different configurations and train them with the same hyperparameters

Pre-train:
1 million sentences from the English Wikipedia, same as SimCSE



InfoNCE loss:

$$\mathcal{L} = -\log \frac{exp\,(q \cdot k/\tau)}{exp\,(q \cdot k/\tau) + \sum_l exp\,(q \cdot l/\tau)}$$

Augmentation Methods:
1. Dropout as SimCSE
2. FGSM
3. Position_shuffle
4. Token dropout
5. Feature dropout
6. Typos
7. Synonym replace
8. Paraphrasing
9. Backtranslation (en->de->en)

Negative Sample Queue:
A first-in-first-out queue to store target branch output. Only a small portion of the queue is randomly initialized at the beginning, and then enqueue and dequeue normally until the end.

A Vanilla BERT encoder. We use the online one for downstream tasks.

EMA:

$$\theta_t \leftarrow \eta\theta_t + (1-\eta)\theta_o$$

Let the encoder of the lower branch update the parameters slowly, so that the negative sample embedding obtained at each step is generated from the encoder with similar parameters as the previous step

* https://huggingface.co/models

3. Experiment

3.1 Main Results

We conduct experiments on seven standard semantic text similarity (STS) tasks. To acclearate the training, we preprocess the training data to directly load while training.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Unsupervised Models (Base)								
GloVe (avg.)	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
BERT (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
BERT-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-BERT	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-BERT	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
RoBERTa (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
RoBERTa-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
DeCLUTR-RoBERT	52.41	75.19	65.52	77.12	78.63	72.41	68.62	69.99
SimCSE	68.40	**82.41**	74.38	80.91	78.56	76.85	72.23	76.25
MoCoSE	**71.48**	81.40	**74.47**	**83.45**	**78.99**	**78.68**	**72.44**	**77.27**
Unsupervised Models (Large)								
SimCSE-RoBERTa	72.86	83.99	75.62	**84.77**	**81.80**	**81.98**	71.26	78.90
SimCSE-BERT	70.88	84.16	76.43	84.50	79.76	79.26	**73.88**	78.41
MoCoSE-BERT	**74.50**	**84.54**	**77.32**	84.11	79.67	80.53	73.26	**79.13**

Table 1: Spearman correlation of MoCoSE on seven semantic text similarity tasks. We compared with the state-of-the-art method SimCSE. MoCoSE achieves the best results with both BERT-base and BERT-large pre-trained models.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Unsupervised Model (Base)								
GloVe (avg.)	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Skip-thought	76.50	80.10	93.60	87.10	82.00	92.20	73.00	83.50
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	**92.80**	69.54	84.94
BERT-[CLS]embedding	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
SimCSE-RoBERTa	81.04	**87.74**	93.28	86.94	**86.60**	84.60	73.68	84.84
SimCSE-BERT	**81.18**	86.46	94.45	88.88	85.50	89.80	74.43	**85.81**
MoCoSE-BERT	81.07	86.43	**94.76**	**89.70**	86.35	84.06	**75.86**	85.46
Unsupervised Model (Large)								
SimCSE-RoBERTa	82.74	87.87	93.66	88.22	**88.58**	**92.00**	69.68	86.11
MoCoSE-BERT	**83.71**	**89.07**	**95.58**	**90.26**	87.96	84.92	**76.81**	**86.90**

Table 2: Performance of MoCoSE on the seven transfer tasks. We compare the performance of MoCoSE and other models on the seven transfer tasks evaluated by SentEval, and MoCoSE remains at a comparable level with the SimCSE.

Learning rate:		Batch size:	
MoCoSE-BERT-base	3e-5	MoCoSE-BERT-base	64
MoCoSE-BERT-large	1e-5	MoCoSE-BERT-large	32
Weight decay	1e-6	Validate per step	100
Negative queue size	512	EMA decay weight	0.75~0.95

Average Spearman's correlation of our best model is 77.27%

3. Experiment

3.2 Abalation Study 1

Symmetric Two-branch Structure

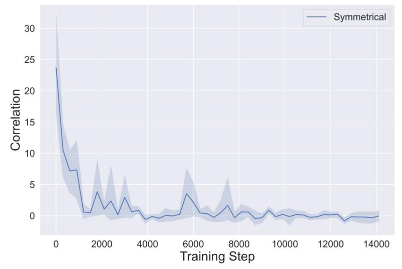


Figure 5: Experiment on a symmetric two-branch structure with EMA decay weight set to 0.

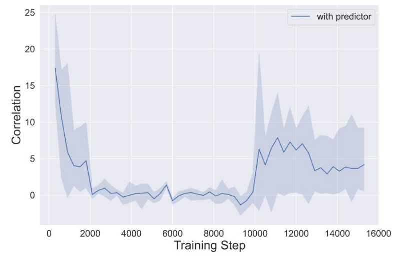
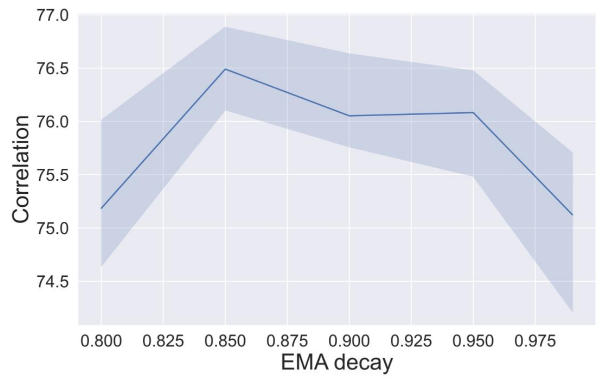


Figure 6: Experiment after adding predictor on the online branch with EMA decay weight set to 0.

Proj.	Pred.	Corr.	Proj.	Pred.	Corr.
	1	60.46		1	66.96
0	2	62.67	2	2	66.29
	3	63.62		3	61.57
	1	76.74		1	31.51
1	**2**	**76.89**	3	2	43.97
	3	76.24		3	39.13

Table 4: The impact of different combinations of projection and predictor on the model.

EMA Hyperparameters



EMA	0.5	0.8	**0.85**	0.9	0.95	0.99
Avg.	75.76	75.19	**76.49**	76.05	76.08	75.12

Table 3: Effect of EMA decay weight on model performance. The best results are obtained with the EMA decay weight at 0.85.

Compared to the choice of EMA decay weight in CV (generally as large as 0.99), the value of 0.85 in our model is smaller, which means that the model is updated faster.

Different Data Augmentations

Augmentation Methods	Avg.
Dropout only	76.76
+ **FGSM** (ε=5e-9)	**77.04**
+ Position_shuffle (True)	73.80
+ Token dropout (prob=0.1)	41.32
+ Feature dropout (prob=0.01)	76.33
+ Feature dropout (prob=0.1)	71.62
+ Typos	22.32
+ Synonym replace (roberta-base)	28.70
+ Paraphrasing (xlnet-base-cased)	60.45
+ Backtranslation (en->de->en)	69.35

Epsilon	1e-9	**5e-9**	1e-8	5e-8	No
Avg.	75.61	**76.64**	75.39	76.62	76.26

Table 9: Different parameters of FGSM in data augmentation affect the model results.

We use the nlpaug toolkit* in our data augmentation experiments. All the parameter listing is default value given by official.

* https://github.com/makcedward/nlpaug

3. Experiment

3.2 Abalation Study 2

Predictor Mapping Dimension

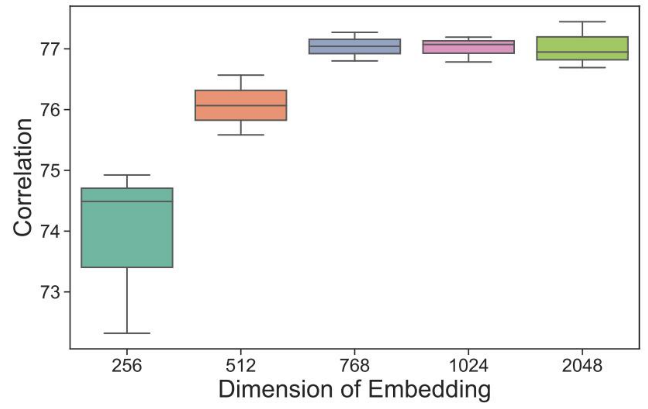


Figure 10: Impact of dimensions of the sentence embedding on the model with fixed queue size of 512.

When the dimension of embedding is low, this causes considerable damage to the performance of the model, while the dimension rises to certain range, the performance of the model stays steady.

Batch Size

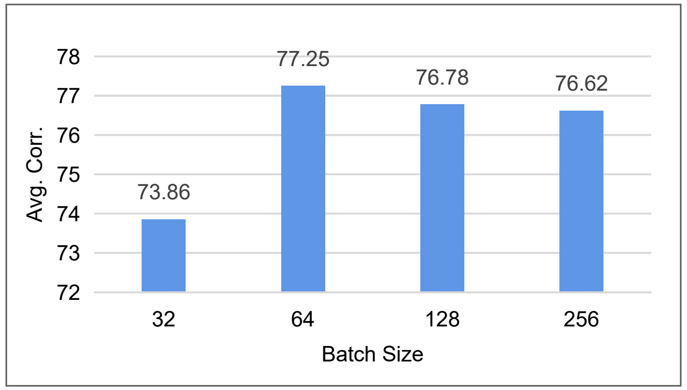


Table 6(b): Impact of batch size on the model with fixed queue size of 512.

The model performance does not improve with increasing batch size, which contradicts the general experience in image contrastive learning.

Distribution of Singular Values

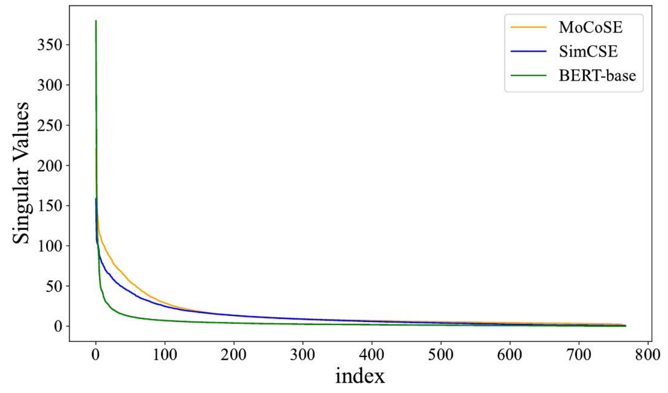
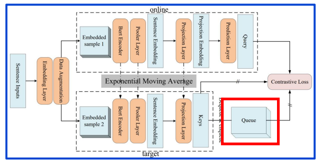


Figure 7: Singular value distributions of sentence embedding matrix from sentences in STS-B.

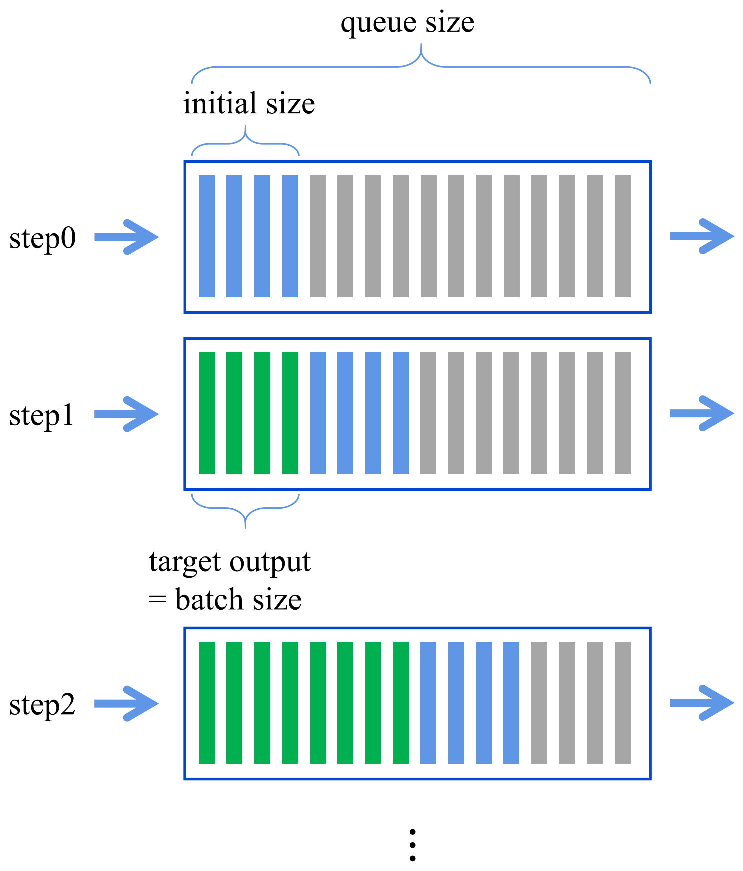
The model is able to alleviate the rapid decline of singular values compared to other methods, making the curve smoother, i.e., our model is able to make the sentence embedding more isotropic.

3. Experiment

3.3 The study of Negative Sample Queue

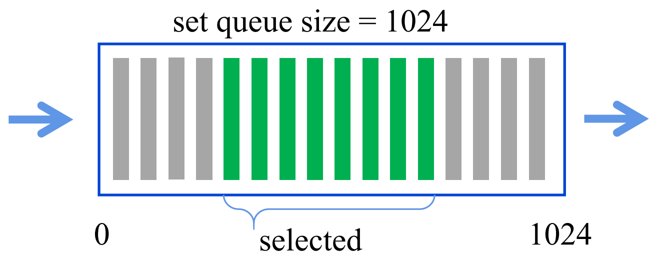


Add random inital process to First-in-First-out queue



Initial Size	Queue Size 128	256	512	1024	4096
w.o. init.	76.40	76.19	75.38	**76.63**	50.17
init. 1/4 queue	75.92	76.34	**77.30**	76.20	**50.42**
init. 1/2 queue	76.16	**76.39**	76.94	76.57	38.74
init. all (normal)	**76.87**	75.81	76.29	76.45	45.80

Table 7: Correlation performance of initializing different proportion of negative queue with different negative queue size.



Corr.	0∼ 512	256∼ 768	512∼ 1024	Without 256∼768	All
Avg.	76.10	**77.02**	75.71	76.18	76.86

Table 8: The impact of negative samples at different locations in the queue on the model performance.

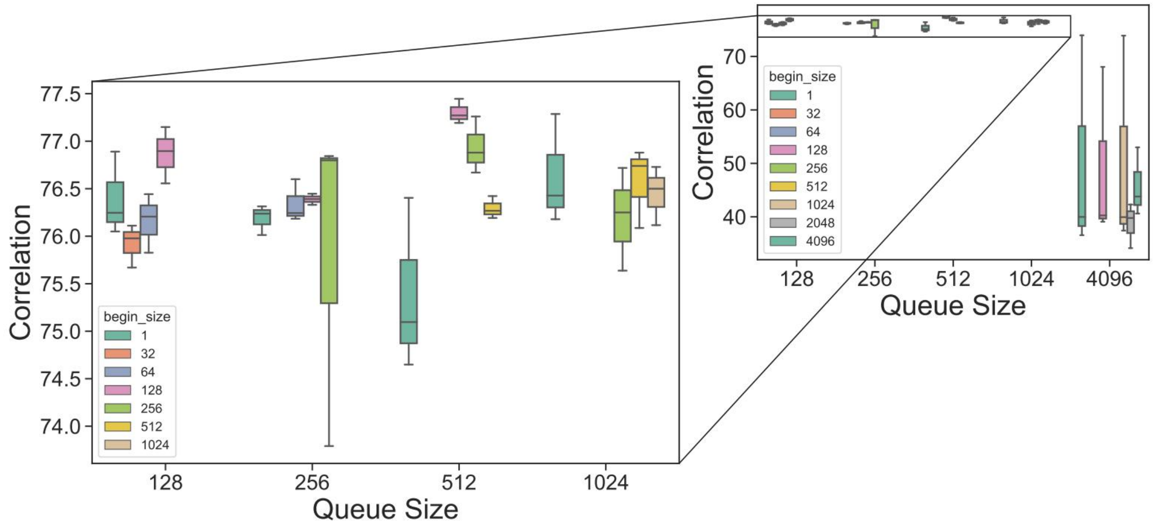


Figure 12: The impact of different initial negative sample queue sizes for different initial sizes on model performance. (left):Zoomed view. (right):Overview with different negative queue size. Results of different initial size under same queue size.

3. Experiment

3.3 The study of Negative Sample Queue

To testify there are historical information in negative sample queue influencing the model performance, we define a Maximum Traceable Distance Metric to help explore the phenomenon.

$$d_{trace} = \frac{1}{1-\eta} + \frac{queue\_size}{batch\_size}$$

The η refers to the decay weight of EMA.
The d_{trace} calculates the update steps between the current online branch and the oldest negative samples in the queue.

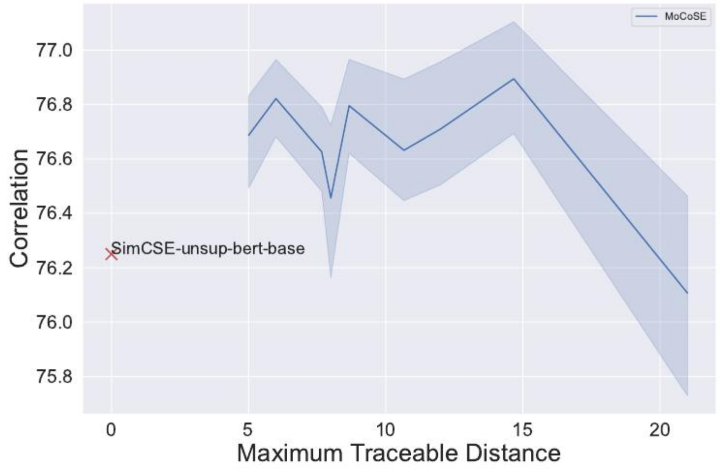


Figure 2: The relationship between traceable distance and model correlation.

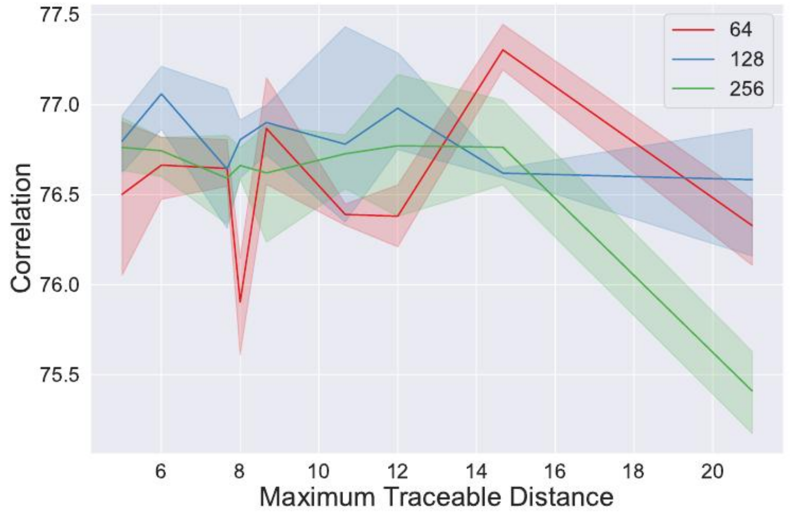


Figure 3: The batch size does not invalidate the traceable distance. The traceable distance needs to be maintained within a reasonable range even for different batch sizes.

4. Conclusion

Our work applies the MoCo-style contrastive learning model to the empirical study of sentence embedding:

1. Negative sample queue is not always better when larger in unsupervised text contrastive learning, which is different from the behavior in image. The experiments show that the model performs best when using the middle part of the queue.
2. We define the Maximum Traceable Distance (MTD) to measure the relationship between the EMA update parameter, batch size, and the size of the negative sample queue during training.
    - Through the experiment, we found that the change in MTD is reflected in the performance of uniformity and alignment of the learned text embedding, and the increase and decrease of MTD pushes uniformity and alignment away from their optimal combination region.
    - Recently, we use a kmeans clustering method to test whether can we decoupling the queue length with three ingredients appeared in MTD.

settings		normal	mean average	5 clusters	10 clusters	15 cluster	20 clusters
ema	0.85						
queue size	512	76.41	76.80	74.37	77.13	77.01	75.82
batch size	64						

    - The results show that the negative samples stored in the queue are clustered by a simple clustering methods to generate a number of pseudo clustering centers, which can also plays a better negative sample role. The clustering is independent of the length of the negative sample queue. This inspires us that perhaps we can use clustering methods to improve the expression of negative sample queue in text contrastive learning.

Thanks for watching!