Improving Unsupervised Contrastive Learning for Sentence Embeddings

Ruixiang Wang

ruixwang@ebay.com

Master Thesis Final Talk July 5, 2022

Human Language Technology and Pattern Recognition
Computer Science Department, RWTH Aachen University
& eBay Research, Aachen

Background

- **eBay Internal Task: eProduct [Yuan & Chiang^{+} 21]**
  - **Task Description**
    - **Given a query q and documents database D (d_i for a single document): both are eBay item titles (a sequence of words)**
    - **For each query q, find top 10 relevant documents d_i from D**
  - **Example**
    - **Query**
      - ***Cisco 5500 Series Wireless Controller, model AIR-CT5508-K9, active licenses***
    - **Relevant documents**
      - ***Cisco 5500 Series Wireless Controller AIR-CT5508-K9 25 AP License***
      - ***Cisco 5500 Series Wireless Controller AIR-CT5508-K9 50 AP License***
      - ***Cisco 5500 Series AIR-CT5508-K9 5508 Wireless LAN Controller 25 AP License***
- **External Tasks: Duplicate Questions Retrieval [Thakur & Reimers^{+} 21]**
  - **Quora**
    - **The dataset is from question-answers platform which identify whether two questions are duplicates**
    - **Given a question as input query, retrieve top k similar questions as output**

Background

- **Decision Rule for All Tasks:**
  - **Given a query q, retrieve top k similar documents from a Database D ($d_i \in D$)**
  - **$sim(q, d_i)$ is a function to calculate similarity between query q and document d_i**
- **The approaches to calculate the similarity as follows:**

1. **Count-based approaches like BM25**
2. **Similarity of sentence embeddings $f_\theta(q)$ and $f_\theta(d_i)$**
3. **Cross-encoder produces an output value indicating the similarity when use concatenated (q, d_i) as input**

We focus on the second approach in this research, since sentence embeddings can

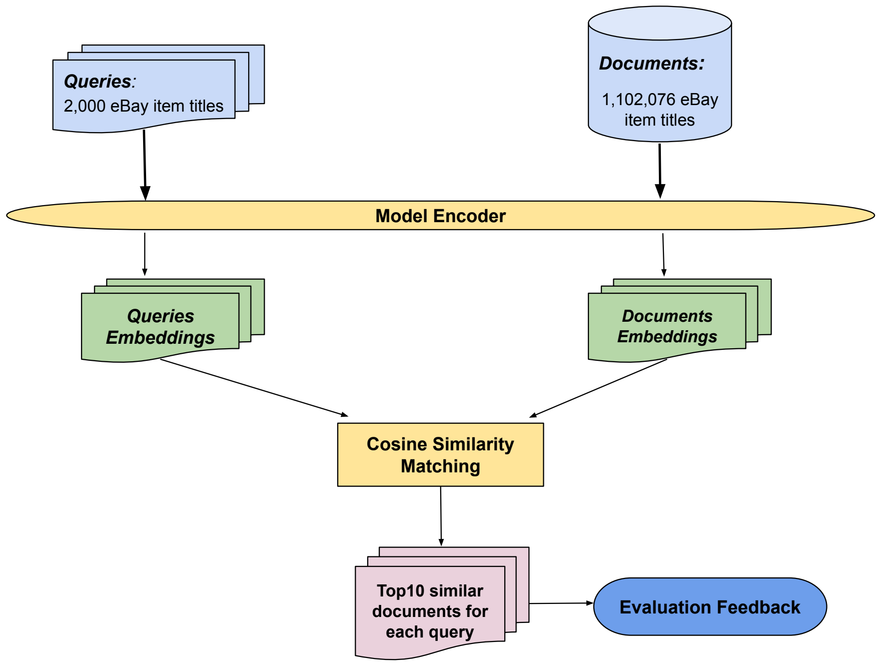
- **capture contextualized information**
- **allow for more efficient retrieval compared to cross-encoder**

Background

- **eBay Internal Task: eProduct embedding-based retrieval diagram**
  - **Use model to encode query q and all documents d_i to contextualized words embeddings**
  - **Through average pooling (mean of contextualized words embeddings of a sentence) to get sentence embeddings $f_\theta(q)$ and $f_\theta(d_i)$**
  - **Calculate cosine similarity based on these embeddings:**

$$sim(f_\theta(q), f_\theta(d_i)) = \frac{f_\theta(q) \cdot f_\theta(d_i)}{||f_\theta(q)|| \cdot ||f_\theta(d_i)||}$$

  - **Find top 10 similar documents for query q based on cosine similarity scores**



Overview

- Motivation
- Contrastive Learning
- State of the Art
  - SimCSE
  - ConSERT
  - BM25
- Methods
  - Autoencoder
  - Multi-task Learning
- Datasets Statistics
- Evaluation Metrics
- Experiments
- Conclusions

Motivation

- Why focus on using unsupervised learning to learn sentence embeddings?
  - Human annotation is costly and often unavailable in real-world
  - There are a lot of unlabelled data which can also be used
  - Investigate how far we can get with unlabelled data
- Contrastive learning methods can boost the performance of sentence embeddings when training with unlabelled data [Gao & Yao$^+$ 21, Yan & Li$^+$ 21, Chuang & Dangovski$^+$ 22]
- Count-based unsupervised method BM25 is a strong and tough-to-beat baseline in many retrieval tasks [Chen & Lakhotia$^+$ 21, Chang & Yu$^+$ 20, Rosa & Rodrigues$^+$ 21, Rau & Kamps 22]
  - Explore if we can close the gap between unsupervised sentence embeddings and unsupervised method BM25
  - Investigate if unsupervised method BM25 and unsupervised sentence embeddings complement each other in practical applications

Contrastive Learning

- **The goal of contrastive learning is to learn an embedding space in which similar sample pairs stay close to each other while dissimilar ones are far apart**
- **General Framework:**
  - **Given input sentence pairs $D = \{(x_i, x_i^+)\}_{i=1}^M$, where x_i and x_i^+ are are semantically related.**
    Each training sample x_i also has a set of negative samples X_i^- (not semantically related to x_i)
  - **Use NT-Xent loss [Chen & Kornblith$^+$ 20]: take a cross-entropy objective. The training objective for a mini-batch of N pairs $\{(x_i, x_i^+)\}_{i=1}^N$ is:**

$$\mathcal{L}_{\text{NT-Xent}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{sim(f_\theta(x_i), f_\theta(x_i^+))/\tau}}{\sum_{x_j \in X_i^- \cup \{x_i^+\}} e^{sim(f_\theta(x_i), f_\theta(x_j))/\tau}} \tag{1}$$

where τ is a temperature hyperparameter used to tune how concentrated the features are in the embedding space. $sim(f_\theta(x_i), f_\theta(x_i^+))$ is the cosine similarity

Constrastive Learning

How to generate positive ${x_i}^+$ for sample x_i?

- **Data Augmentation (dropout, feature cutoff, token cutoff, etc.)**

How to generate negatives X_i^- for sample x_i?

- **In-Batch Negatives**
  - **Consider all other input sentences (except x_i) in batch as negatives**
  - **Allows to efficiently use more negative samples in one batch**

State of the Art

- SimCSE [Gao & Yao$^+$ 21]
  - Main idea: Dropout noise of model as data augmentation to generate positive pairs for Contrastive Learning
  - Input: Sentence pairs $D = \{(x_i, x_i^+)\}_{i=1}^M$ where $x_i^+ = x_i$
  - Training process: Feed input into BERT-based model through the use of default independent of *dropout masks* to calculate NT-Xent loss
- ConSERT [Yan & Li$^+$ 21]
  - Main idea: Use multiple text-based data augmentation methods to generate positive pairs for Contrastive Learning
  - Input: Sentence pairs $D = \{(x_i, x_i^+)\}_{i=1}^M$ where $x_i^+ = x_i$
  - Training process:
    - Use BERT-based model as encoder, remove its default Dropout
    - Feed input to token embedding layer of encoder to generate embeddings
    - Apply text-based data augmentations to these embeddings to get new embeddings
    - Feed new embeddings into encoder to calculate NT-Xent loss

State of the Art

- **Count-based Method: BM25 [Robertson & Zaragoza 09]**

Given a query q_1^I and a set of N documents D the BM25 score of the document $[d_n]_1^{J_n}$ is:

$$S(q_1^I, [d_n]_1^{J_n}) = \sum_{i=1}^{I} \text{IDF}(q_i) \cdot \frac{\text{TF}(q_i, [d_n]_1^{J_n}) \cdot (k_1 + 1)}{\text{TF}(q_i, [d_n]_1^{J_n}) + k_1 \cdot (1 - b + b \cdot \frac{J_n \cdot N}{\sum_{n'=1}^{N} J_{n'}})}$$

with k_1 and b hyperparameters

$$\text{TF}(q_i, [d_n]_1^{J_n}) = \frac{\sum_{j=1}^{J_n} \delta(q_i, [d_n]_j)}{J_n}$$

and

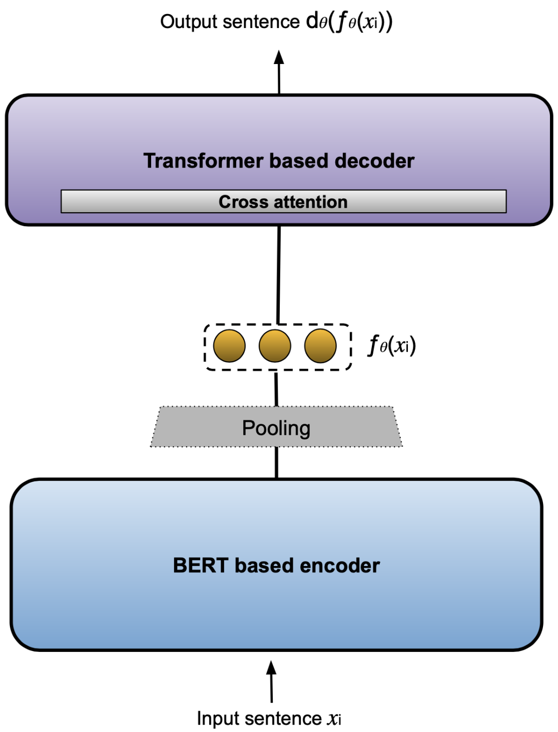
$$\text{IDF}(q_i) = \log\left(\frac{N}{\sum_{n=1}^{N} \delta(q_i \in d_n)}\right)$$

Methods

- Autoencoder is a promising approach to learn sentence representations in an unsupervised way [Shen & Mueller^{+} 20]
- General framework of Autoencoder
  - Autoencoder is based on encoder-decoder architecture
  - The encoder maps input sentence x_i to a new embedding $f_\theta(x_i)$ in a latent space, the decoder reconstructs $f_\theta(x_i)$ into $d_\theta(f_\theta(x_i))$ [Kingma & Welling 13]
  - The goal of autoencoder is to make output $d_\theta(f_\theta(x_i))$ and input x_i identical

Methods

- **CLM Autoencoder [Wang & Reimers^{+} 21]:**



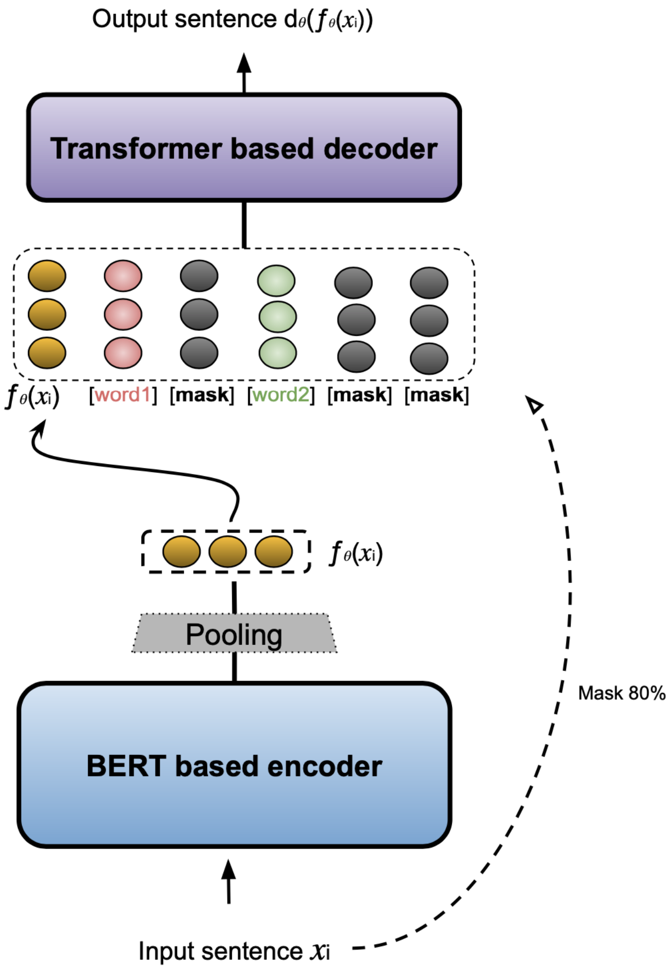
- **The Training loss for a mini-batch of N sentences:**

$$\begin{aligned}\mathcal{L} &= -\frac{1}{N}\sum_{i=1}^{N}\log P_\theta([x_i]_1^{T_i}|f_\theta(x_i)) \\ &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\log P_\theta\left([x_i]_t \mid [x_i]_0^{t-1}, f_\theta(x_i)\right)\end{aligned} \tag{2}$$

where $[x_i]_1^{T_i}$ indicates all tokens of input sentence x_i (T_i is the length of x_i) , $[x_i]_t$ is the t-th token of input sentence x_i , $[x_i]_0^{t-1}$ is a seqence of tokens $[x_i]_0[x_i]_1 \ldots [x_i]_{t-1}$ of input sentence x_i

Methods

- **MLM Autoencoder:**



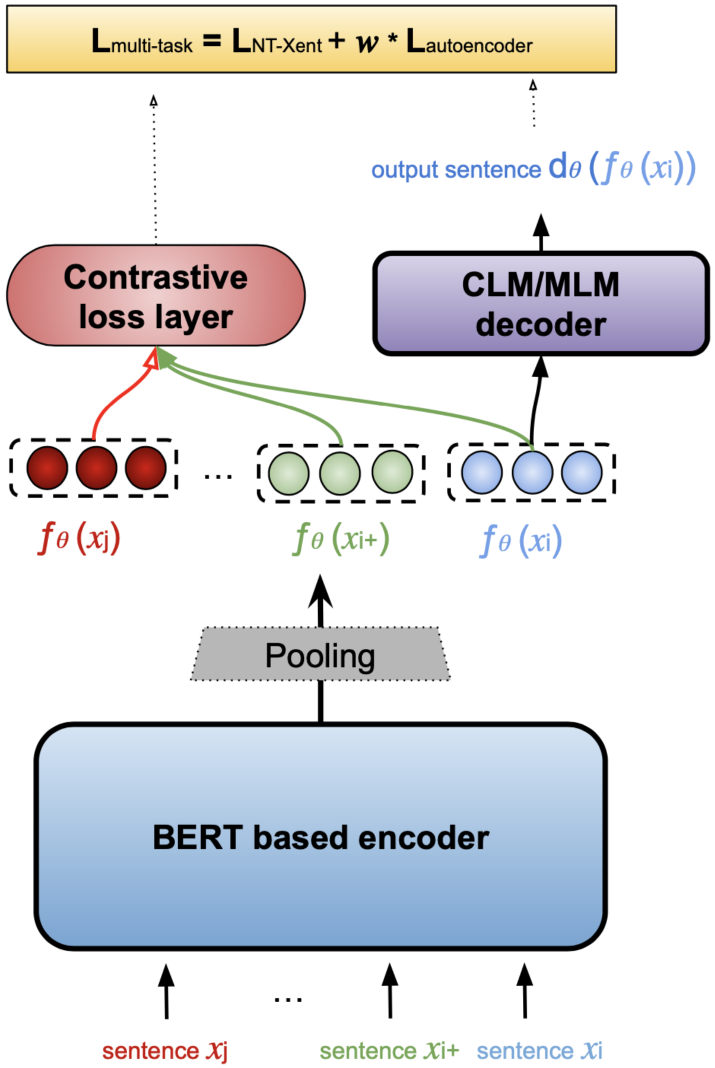
- **The Training loss for a mini-batch of N sentences:**

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{m\in M_i}\log P_\theta([x_i]_m | f_\theta(x_i), [x_i]_1^{T_i} \setminus M_i) \tag{3}$$

where M_i ($m \in M_i$) is the set of masked tokens of input sentence x_i, $[x_i]_m$ is the original token from x_i of masked position m, $[x_i]_1^{T_i}$ indicates all tokens of input sentence x_i

Methods

▶ **Multi-task learning for contrastive learning and autoencoder**



▷ **Contrastive learning loss converges very fast, adding autoencoder loss can make the whole training task more difficult**

▷ **Model can learn sentence similarity information as well as word level information from this combination**

Training Dataset Statistics

- **Training Datasets (unlabelled sentences)**

	Sentences	*Words per sentence*	*Running words*	*Vocabularies*
OpenWebText 1M	1,000,000	19.85	19,857,849	866,206
OpenWebText 100k	100,000	19.85	1,985,570	190,191
eBay Titles 1M	1,000,000	10.90	10,908,538	774,092
eBay Titles 100k	100,000	10.90	1,090,510	158,338
Quora titles 100k	100,000	11.41	1,141,440	85,014

- **Validation Datasets (unlabelled sentences)**

	Sentences	*Words per sentence*	*Running words*	*Vocabularies*
OpenWebText 10k	10,000	19.82	198,286	40,507
eBay Titles 10k	10,000	10.97	109,793	32,727
Quora titles 10k	10,000	11.52	115,206	20,038

OpenWebText sampled from OpenWebTextCorpus [Gokaslan & Cohen]. eBay Titles sampled from eBay internal database. Quora titles sampled from Quora questions dataset [Thakur & Reimers^{+} 21]

Evaluation Dataset Statistics

- **eProduct [Yuan & Chiang+ 21]**

eProduct	*Titles*	*Words per title*
Query@dev	2,000	10.97
Index	1,102,076	10.73

- **Semantic Textual Similarity (STS) [Conneau & Kiela 18]**

	STS-Avg	STS-B
Number of test samples	12,544	1,379

Each sample in these datasets contains a pair of sentences as well as a gold score between 0 and 5 indicating their semantic similarity

- **Duplicate Question Retrieval (Quora) [Thakur & Reimers+ 21]**

	Quora
#Queries	10,000
Avg. Query Lengths	9.53
#Documents	522,931
Avg. Document Lengths	11.44
Avg. D / Q	1.6

Avg. D/Q indicates the average relevant documents per query

Evaluation Metrics

- $r_{i@k}$ is the number of documents retrieved from k are groundtruth matches for query i
- g_i is the number of all groundtruth matches for query i
- $g_{i@k}$ is the capped number of groundtruth matches for query i. Note that $g_{i@k} \in [1, k]$, i.e. $g_{i@k} = min(k, g_i)$

- **Recall@k ($R@k$)**

$$R@k = \frac{1}{N} \sum_{i=1}^{N} \frac{r_{i@k}}{g_i} \tag{4}$$

- **Precision@k ($P@k$)**

$$P@k = \frac{1}{N} \sum_{i=1}^{N} \frac{r_{i@k}}{k} \tag{5}$$

- **Capped Recall@k ($R_{cap}@k$):**

$$R_{cap}@k = \frac{1}{N} \sum_{i=1}^{N} \frac{r_{i@k}}{g_{i@k}} \tag{6}$$

Evaluation Metrics

- **eProduct**
  - **Capped Recall@10 (R_{cap}@10)**
  - **Precision@10 (P@10)**
- **Quora**
  - **Recall@k (R@k), $k \in \{1, 3, 5, 10, 100, 1000\}$**
  - **Precision@k (P@k), $k \in \{1, 3, 5, 10, 100, 1000\}$**
- **Semantic Textual Similarity (STS)**

  Report the Spearman's correlation between the cosine similarity of the sentence representations and the human-annotated gold scores for STS-B and STS-Avg

Pretrained Models

- **Bert-base-uncased**
  - **BERT [Devlin & Chang$^+$ 18] is a transformers model pretrained on a large corpus of English data in a self-supervised fashion**
  - **The BERT model was pretrained on BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia (excluding lists, tables and headers)**
- **eBert**
  - **Same model architecture as BERT**
  - **The eBert model was pretrained on the BooksCorpus and English Wikipedia (same in BERT). Additionally, use 1B item titles from eBay e-commerce platform**
- **Word2vec**
  - **Word2vec [Mikolov & Chen$^+$ 13] model maps each word into a vector**
  - **Word2vec model was pretrained on eProduct dataset when do evaluation on eProduct task**

Basic Experimental Setup

- Pre-trained base models: bert-base-uncased model or eBert model
- Best model checkpoint selection: Use validation loss
- Shuffle training dataset before each epoch if train multiple epochs
- Report average score over 3 random seeds (1,7,42)
- Hyperparameters setting:

epochs	1
batch size	64
warmup steps	910
temperature	0.05
weight decay	0
max seq. length	32
learning rate	3e-5
pooling	avg.

Experiment I

- ***Compare ConSERT and SimCSE in eBay internal dataset and model***
- **Pre-trained base models: eBert model**
- **Both ConSERT and SimCSE use *eBay titles 100k* to fine-tune the base model**
- **Evaluation on eProduct and STS**

Model	eProduct[%]		STS-B[%]	STS-Avg[%]
	R_{cap}@10	P@10		
ConSERT$_1$	68.7	61.6	62.1	58.4
ConSERT$_2$	69.1	61.9	60.6	59.4
SimCSE	69.1	61.9	61.9	56.7
eBert	50.9	45.9	61.2	57.9
Word2vec	48.4	43.7	-	-
BM25	73.1	65.8	-	-

ConSERT$_1$ uses *Token Shuffling + Feature Cutoff* combination and ConSERT$_2$ uses *Token Shuffling + Embedding Dropout* combination

BM25 & SimCSE Comparison Results

- **Combination with SimCSE and BM25**
  - **Rescoring BM25 top1000 with SimCSE according to:**

$$\text{Score}(q, d_i) = \text{Score}_{\text{BM25}}(q, d_i) + \alpha \cdot \text{Score}_{\text{SimCSE}}(q, d_i) \tag{7}$$

    where q is a query and d_i is a document
  - **Select SimCSE model with the best random seed**

Model	eProduct[%]	
	$R_{cap}@10$	$P@10$
SimCSE	69.6	62.5
BM25	73.1	65.8
Rescoring	74.5	67.0

The combination of BM25 and SimCSE can achieve better performance, which indicates BM25 and SimCSE may complement each other in some cases

Experiment II

- ***Compare different sizes of training corpus for SimCSE in eBay internal task***
- **Use *eBert* model as base model and *eBay titles 1M/100k/50k/20k/10k* as training dataset**
- **Evaluation on eProduct with training 1 epoch**

Training Dataset (eBay titles)	1M	100k	50k	20k	10k
Training Epoch	1	1	1	1	1
Number of Steps	15625	1562	781	312	156
R_{cap}@10	65.3	69.1	68.8	68.2	66.4
P@10	58.6	61.9	61.7	60.9	59.4

eBay titles 100k has the best performance when training with only 1 epoch. Smaller datasets like eBay titles 50k/20k can also achieve decent results on eProduct. Larger number of training steps (e.g. eBay titles 1M) may lead to overfitting

Influence of Training Corpus Size for SimCSE

- Evaluation on eProduct with training 3 epochs

Training Dataset (eBay titles)	1M	100k	50k	20k	10k
Training Epoch	3	3	3	3	3
Number of Steps	46875	4686	2343	936	468
R_{cap}@10	61.6	67.9	68.5	69.4	67.9
P@10	55.3	60.9	61.4	62.1	60.5

eBay titles 20k has the best performance when training with 3 epoch. Larger training steps may damage the performance on eProduct

Experiment III

- ***Compare different domains of training datasets for SimCSE and Autoencoder in eBay internal task***
- **Use *eBert* or *Bert-base-uncased* as base model and *eBay titles 100k, OpenWebText 100k, Quora titles 100k* as training dataset**
- **Train 3 epochs, use 2 decoder layers for autoencoder based methods**
- **Evaluation on eProduct with base model eBert**

Model	Training Dataset	eProduct[%]	
		R_{cap}@10	P@10
SimCSE	eBay titles 100k	67.9	60.9
	OpenWebText 100k	71.2	63.7
	Quora titles 100k	70.5	63.0
CLM autoencoder	eBay titles 100k	68.3	61.4
	OpenWebText 100k	66.5	59.9
	Quora titles 100k	66.1	59.7
MLM autoencoder	eBay titles 100k	68.9	61.9
	OpenWebText 100k	67.0	60.3
	Quora titles 100k	66.4	59.8

Autoencoder based methods have better performance on in-domain dataset, contrastive learning method SimCSE has better performance on out-domain dataset

Different Domains of Training Dataset Evaluation

- Evaluation on eProduct with base model Bert-base-uncased

Model	Training Dataset	eProduct[%] R_{cap}@10	eProduct[%] P@10
SimCSE	eBay titles 100k	65.7	59.2
	OpenWebText 100k	64.0	57.5
	Quora titles 100k	65.3	58.7
CLM autoencoder	eBay titles 100k	66.9	60.0
	OpenWebText 100k	62.4	56.2
	Quora titles 100k	63.2	56.9
MLM autoencoder	eBay titles 100k	67.5	60.5
	OpenWebText 100k	63.7	57.1
	Quora titles 100k	63.3	56.8

- Autoencoder based methods learn words information from in-domain datasets, since their objective is to reconstruct input sentence word by word
- The relationship between SimCSE and domain of training corpus on eProduct is still unclear
- *Hypothesis*: eBert was pre-trained on eBay titles 1B, fine-tuning eBert with in-domain eBay titles 100k using SimCSE method may lead to overfitting. Out-domain datasets could add some noise to make the trained model more robust

Experiment IV

- ***This experiment aims to compare contrastive learning method and autoencoder methods as well as their combination in eBay internal task***
- **Use *eBert* model as base model and *eBay titles 100k* as training dataset**
- **Train 3 epochs, use 2 decoder layers for autoencoder**
- **Evaluation on eProduct**

Model	eProduct[%]	
	R_{cap}@10	P@10
SimCSE[1]	69.1	61.9
SimCSE	67.9	60.9
CLM autoencoder	68.3	61.4
MLM autoencoder	68.9	61.9
SimCSE + CLM autoencoder	69.7	62.6
SimCSE + MLM autoencoder	68.6	61.6
eBert	50.9	45.9
BM25	73.1	65.8

SimCSE[1] indicates using SimCSE and training for 1 epoch
We use multi-task loss with different weight $w \in \{0.1, 0.01, 0.001, 0.0001\}$ and select best results for the combination

Experiment V

- ***Compare contrastive learning method and autoencoder methods as well as their combination in Quora task***
- **Use *Bert-base-uncased* model as base model and *Quora titles 100k* as training dataset**
- **Use recall@10 of quora devset for best model checkpoint selection**
- **Train 3 epochs, use 2 decoder layers for Autoencoder**
- **Evaluation on Quora**

Model	R@100[%]	R@10[%]
SimCSE[1]	96.5	86.8
SimCSE	97.2	88.3
CLM autoencoder	95.9	84.9
MLM autoencoder	91.2	77.5
SimCSE + CLM autoencoder	97.4	88.8
SimCSE + MLM autoencoder	97.3	88.3
Bert-base-uncased	86.1	71.7
BM25	97.3	88.9

SimCSE[1] indicates using SimCSE and training for 1 epoch
We use multi-task loss with different weight $w \in \{0.1, 0.01, 0.001, 0.0001\}$ and select best results for the combination

Conclusions

- **The unsupervised method BM25 still outperforms unsupervised sentence embeddings on some retrieval tasks, but they could be complementary in some cases**
- **Larger number of training steps may quickly lead to overfitting for SimCSE**
- **The domain effect of training corpus for SimCSE on eProduct is still an open question**
- **Multi-task learning may help to improve the performance of sentence embeddings on some retrieval tasks**

Thank you for your attention

Ruixiang Wang

`ruixwang@ebay.com`

`http://www-i6.informatik.rwth-aachen.de/`

References

[Chang & Yu+ 20] W. Chang, F. X. Yu, Y. Chang, Y. Yang, S. Kumar. Pre-training tasks for embedding-based large-scale retrieval. *CoRR*, Vol. abs/2002.03932, 2020. 6

[Chen & Kornblith+ 20] T. Chen, S. Kornblith, M. Norouzi, G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020. 7

[Chen & Lakhotia+ 21] X. Chen, K. Lakhotia, B. Oguz, A. Gupta, P. S. H. Lewis, S. Peshterliev, Y. Mehdad, S. Gupta, W. Yih. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *CoRR*, Vol. abs/2110.06918, 2021. 6

[Chuang & Dangovski+ 22] Y.-S. Chuang, R. Dangovski, H. Luo, Y. Zhang, S. Chang, M. Soljačić, S.-W. Li, W.-t. Yih, Y. Kim, J. Glass. Diffcse: Difference-based contrastive learning for sentence embeddings, 2022. 6

[Conneau & Kiela 18] A. Conneau, D. Kiela. Senteval: An evaluation toolkit for universal sentence representations, 2018. 16

[Devlin & Chang+ 18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, Vol., 2018. 19

[Gao & Yao+ 21] T. Gao, X. Yao, D. Chen. SimCSE: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, Vol., 2021. 6, 9, 34

[Gokaslan & Cohen] A. Gokaslan, V. Cohen. Openwebtext corpus. 15

[Kingma & Welling 13] D. P. Kingma, M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, Vol., 2013. 11

[Mikolov & Chen+ 13] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio, Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. 19

[Rau & Kamps 22] D. Rau, J. Kamps. How different are pre-trained transformers fornbsp;text ranking? In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II*, 207–214, Berlin, Heidelberg, 2022. Springer-Verlag. 6

[Robertson & Zaragoza 09] S. Robertson, H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, Vol. 3, No. 4, pp. 333–389, apr 2009. 10

[Rosa & Rodrigues+ 21] G. M. Rosa, R. C. Rodrigues, R. de Alencar Lotufo, R. Nogueira. Yes, BM25 is a strong baseline for legal case retrieval. *CoRR*, Vol. abs/2105.05686, 2021. 6

[Shen & Mueller+ 20] T. Shen, J. Mueller, R. Barzilay, T. Jaakkola. Educating text autoencoders: Latent representation guidance via denoising. In *International Conference on Machine Learning*, pp. 8719–8729. PMLR, 2020. 11

[Thakur & Reimers+ 21] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, I. Gurevych. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models, 2021. 2, 15, 16

[Wang & Reimers^{+} 21] K. Wang, N. Reimers, I. Gurevych. TSDAE: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. In *EMNLP*, 2021. 12

[Yan & Li^{+} 21] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, W. Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*, Vol., 2021. 6, 9, 36

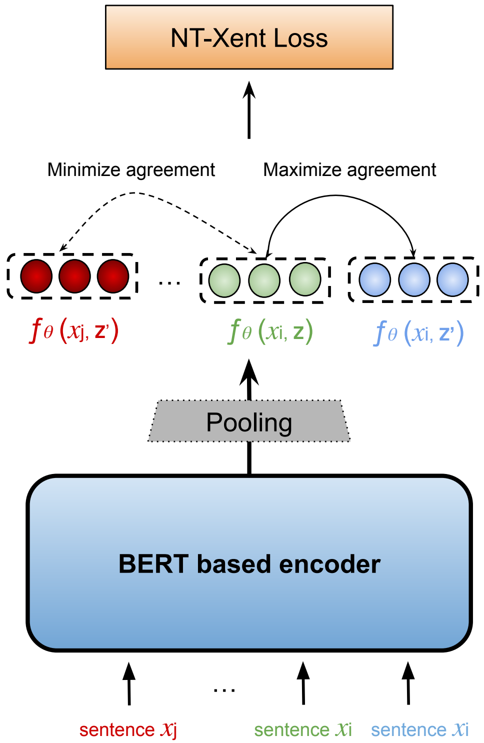
[Yuan & Chiang^{+} 21] J. Yuan, A.-T. Chiang, W. Tang, A. Haro. eProduct: A million-scale visual search benchmark to address product recognition challenges. *arXiv preprint arXiv:2107.05856*, Vol., 2021. 2, 16

State of the Art

- **SimCSE [Gao & Yao$^+$ 21]**
  - **Main idea: Dropout noise of model as data augmentation to generate positive pairs**
  - **Input: Sentence pairs $\{(x_i, x_i^+)\}$ where $x_i^+ = x_i$**
  - **Training process: Feed these identical positive pairs into BERT-based model through the use of independent *dropout masks* z, z'**
  - **Training objective: we denote by $f_\theta(x_i, z)$ the generated embedding for x_i through the dropout mask z. For a mini-batch of N sentences using non-symmetric NT-Xent loss**

State of the Art

- **Structure of SimCSE**

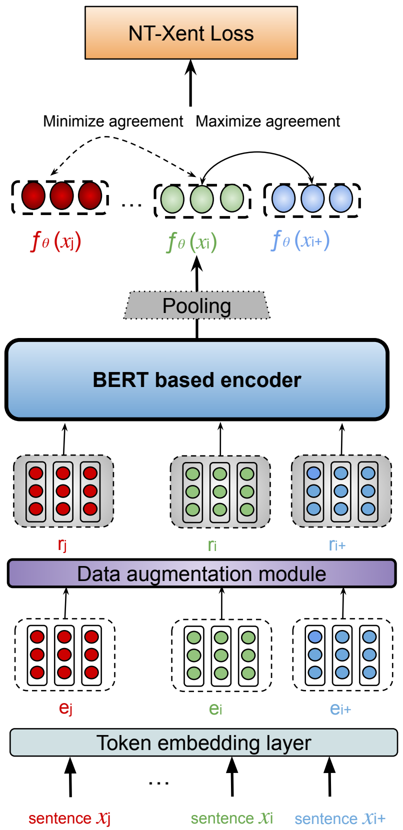


State of the Art

- **ConSERT [Yan & Li$^+$ 21]**
  - **Main idea: Use multiple text-based data augmentation methods to generate positive pairs**
  - **Input: Sentence pairs $\{(x_i, x_i^+)\}$ where $x_i^+ = x_i$**
  - **Training process:**
    - **Use BERT-based model T as encoder, remove its default Dropout**
    - **Apply token embedding layer of T to sentence pairs $\{(x_i, x_i^+)\}$ and get two same embeddings: $\{(e_i, e_i^+)\}$ where $e_i, e_i^+ \in \mathbb{R}^{L\times d}$, L is the sequence length and d is the hidden dimension**
    - **Apply different data augmentation strategies to $\{(e_i, e_i^+)\}$ (including token shuffling, cutoff, etc.) to get new embeddings $\{(r_i, r_i^+)\}$**
    - **Feed $\{(r_i, r_i^+)\}$ to BERT-based model T to get final embedding $\{(f_\theta(x_i), f_\theta(x_i^+))\}$ through average pooling**
  - **Training objective: For a mini-batch of N pairs using symmetric NT-Xent loss.**

State of the Art

- **Structure of ConSERT**



NT-Xent Loss with Binary Cross Entropy Loss

- **Replace NT-Xent loss with Binary Cross Entropy loss for contrastive learning**
  - **Binary Cross Entropy (BCE) loss**

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log g(sim(f_\theta(x_i), f_\theta(x_i^+))) + \sum_{x_i^- \in X_i^-} \log(1 - g(sim(f_\theta(x_i), f_\theta(x_i^-))))\right)$$

with $g(x) = \sigma(x^\tau)$. where τ is a temperature hyperparameter used to tune how concentrated the features are in the representation space. $sim(f_\theta(x_i), f_\theta(x_i^+))$ is the cosine similarity

Compare BCE loss and NT-Xent loss in SimCSE

- **Use *Bert-base-uncased* as base model and *OpenWebText 1M* as training dataset**
- **BCE & NT-Xent Loss Comparison Results**
  - **Evaluation on Semantic Textual Similarity (STS)**

Model	STS-B[%]	STS-Avg[%]
$\text{SimCSE}_{\text{NT-Xent}}$	71.4	71.9
$\text{SimCSE}_{\text{BCE}}$	43.3	44.2
Bert-base-uncased	47.3	51.6

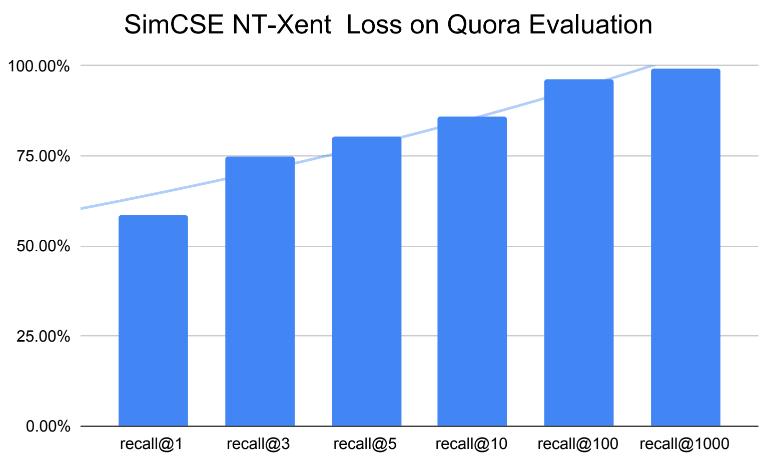
  - **Evaluation on Quora and CQADupStack**

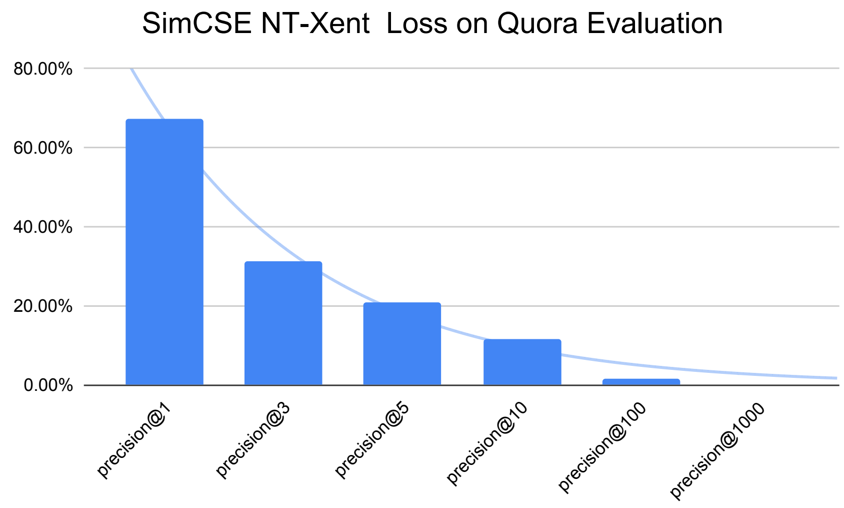
Model	R@100[%]	
	Quora	CQADupStack
$\text{SimCSE}_{\text{NT-Xent}}$	96.2	38.2
$\text{SimCSE}_{\text{BCE}}$	59.8	8.7
Bert-base-uncased	86.1	19.0
BM25	97.3	60.6

- **Binary Cross Entropy loss may not be directly applied to contrastive learning**

SimCSE with NT-Xent Loss Quora Results

- **SimCSE$_{\text{NT-Xent}}$ results on Quora Evaluation with different $k \in \{1, 3, 5, 10, 100, 1000\}$**





BM25 & SimCSE Comparison Results

- **Analysis of eProduct retrieval results**
  - **In this case SimCSE outperforms BM25**

Query	Vintage Fisher-Price Elephant Rattle Baby Toy Take Along Crib Stroller #619
BM25	Vintage Fisher-Price Elephant Rattle Baby Toy Take Along Crib Stroller #619 Blue Cute Baby Crib Stroller Rattles Seat Take Along Travel Arch Development Toys Fisher-Price Ocean Wonders Take-Along Projector Soother Baby - Kids Toy New
SimCSE	Vintage Fisher-Price Elephant Rattle Baby Toy Take Along Crib Stroller #619 Blue VINTAGE FISHER PRICE LOOK AT ME ELEPHANT RATTLE #429-1977 Vintage Baby Rattle Toy 1988 Discovery Toys elephant learning toy infant

Hypothesis: the dense retrieval model may get semantic information and modeling of term importance

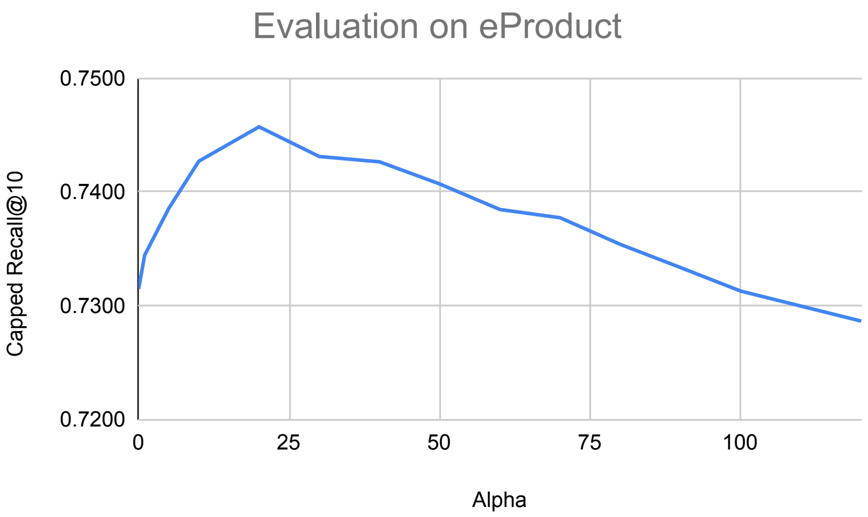
  - **In this case BM25 outperforms SimCSE**

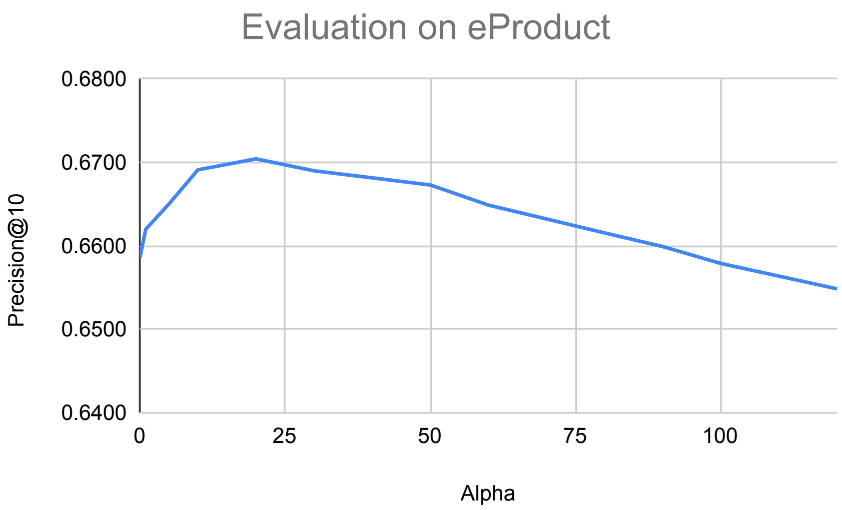
Query	ASICS GEL-Sonoma 3 Trail Running Shoes - Navy - Womens
BM25	Asics Gel Sonoma 3 Women's T774N 9667 Shoes Trail Running Grey Aqua Asics Women's GEL - Sonoma Trail Running Shoe - Assorted Sizes & Colors Asics Gel Sonoma 2 Running Sneakers Womens Trail Shoes Flat Heel
SimCSE	Asics Women's GEL - Sonoma Trail Running Shoe - Assorted Sizes & Color Asics Gel Sonoma 2 Running Sneakers Womens Trail Shoes Flat Heel B-604 Asics Women's GEL-Sonoma 2 Trail Running Shoes 9

Hypothesis:the dense retrieval model may confuse product numbers and product specifications

BM25 & SimCSE Comparison Results

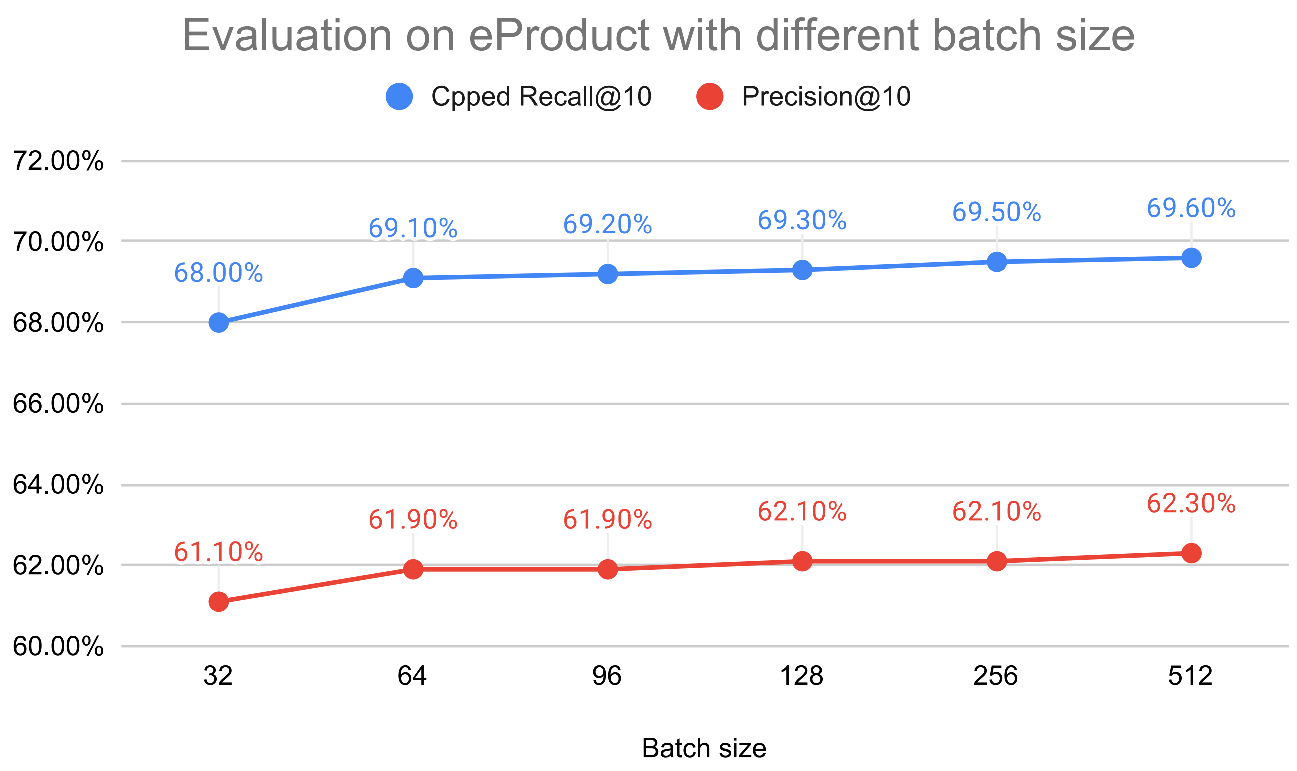
- **Combination with SimCSE and BM25**
  - **Combination score evaluate on eProduct using different α**





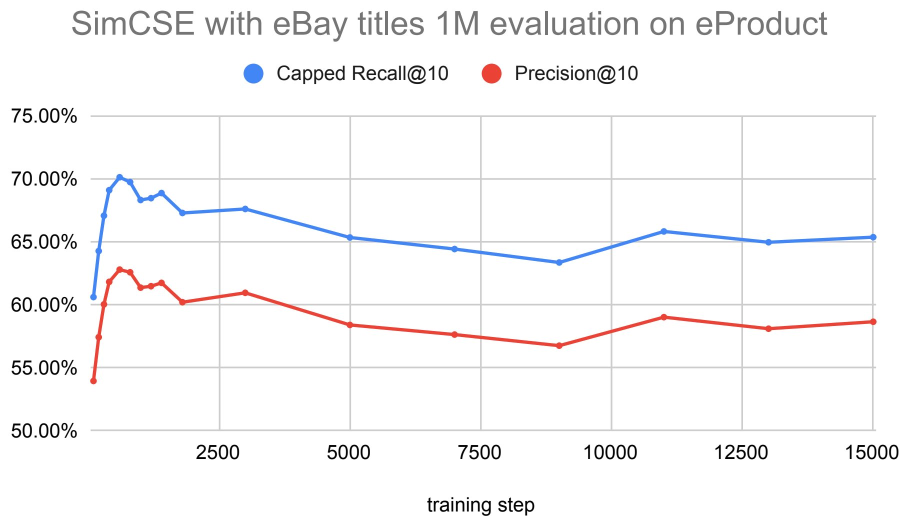
Influence of Batch size for SimCSE

- **Different batch size gives various number of negatives since SimCSE uses in-batch negatives**
- **Report eProduct results when training with *eBay titles 100k* and eBert model using different batch size 32, 64 (*default*), 96, 128, 256, 512**



SimCSE With eBay Titles 1M Training Process

- **SimCSE with eBay titles 1M loss values and evaluation results during training**



Different ePoduct Titles Comparison Results

- Data statistics of eProduct titles

Training Dataset	#Sentences	#Duplicates
eProduct titles 1M	1,000,000	80.326
eProduct titles 100k	100,000	2,020
unqiue eProduct titles 1M	1,000,000	0
unqiue eProduct titles 100k	100,000	0

- Comparison with ePoduct titles & unique eProduct titles (remove duplicates titles) using SimCSE on eProduct evaluation

Model	Training Dataset	eProduct[%]	
		$R_{cap}@10$	$P@10$
SimCSE$_{\text{eBERT}}$	eProduct titles 1M	56.2	50.6
	eProduct titles 100k	60.1	54.3
	unique eProduct titles 1M	59.1	53.1
	unique eProduct titles 100k	62.4	56.3