SimCSE:
Simple Contrastive Learning of Sentence Embeddings

Author: Tianyu Gao, Xingcheng Yao, Danqi Chen

Presenter: Sunyoung Song

Contrastive Learning

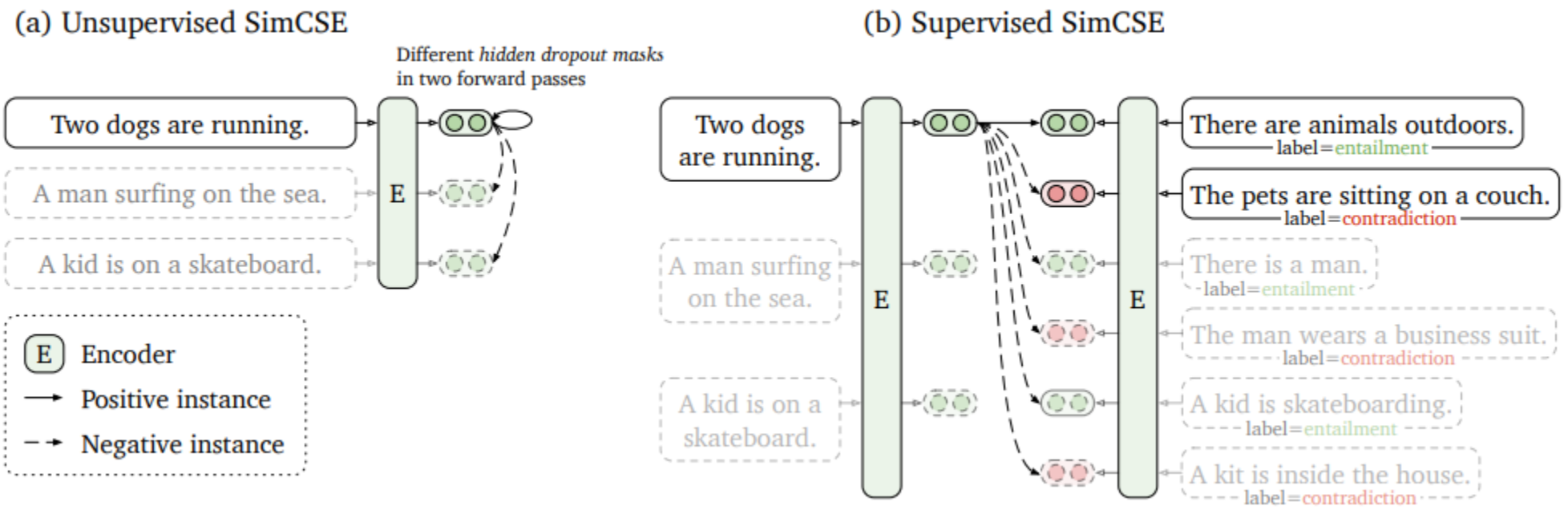
- 의미적으로 가까운 것들은 끌어당기고, 아닌 것들은 밀어냄으로써 효과적인 표현을 배우는 기법

- Contrastive learning은 모든 관측치를 클래스로 간주하고 softmax를 사용할 수 없기 때문에 다른 관측치들의 일부만을 샘플링하여 손실함수를 계산하는 noise contrastive estimation (NCE) 방식을 사용

- NCE loss란?
  - CBOW와 Skip-Gram 모델에서 사용하는 비용 계산 알고리즘
  - 전체 데이터셋에 대해 softmax 함수를 적용하는 것이 아니라 샘플링으로 추출한 일부에 대해서만 적용하는 방법

$$L_{infoNCE} = -log\frac{exp(sim(q,k_+)/\tau)}{exp(sim(q,k_+)/\tau) + \sum_{i=0}^{K} exp(sim(q,k_i)/\tau)}$$

SimCSE


(a) Unsupervised SimCSE
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
(b) Supervised SimCSE
Two dogs are running.
A man surfing on the sea.
A kid is on a skateboard.
E
E
There are animals outdoors.
label=entailment
The pets are sitting on a couch.
label=contradiction
There is a man.
label=entailment
The man wears a business suit.
label=contradiction
A kid is skateboarding.
label=entailment
A kit is inside the house.
label=contradiction

SimCSE

1. Unsupervised SimCSE

- Positive pair
  - Dropout을 noise로 사용
    - 동일한 문장을 사전 훈련된 encoder에 두 번 전달
    - 서로 다른 dropout mask가 적용되어 두 개의 다른 embedding을 positive pair로 얻음
    - Dropout는 완전 연결층에 배치된 dropout으로 기본값 $p = 0.1$

- Negative pair
  - 같은 mini-batch 내의 다른 문장들을 negative pair로 사용

- loss

$$\ell_i = -\log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}},$$

τ : temperature hyperparameter
$sim(h_i, h_i^+)$: cosine similarity

분자: positive 와의 유사도
분모: negative 와의 유사도

SimCSE

2. supervised SimCSE

- Dataset: NLI datasets (SNLI + MNLI)
  - QQP, Flickr30k, paraNMT, NLI datasets으로 실험 진행 결과, NLI datasets이 학습에 가장 효과적이였음
  - NLI datasets label – entailment, neutral, contradiction

- Positive pair : entailment
- Negative pair: contradiction을 hard negative로 사용

- Loss

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}+e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)}$$

h_i : premise
h_i^+ : entailment
h_i^+ : contradiction

Alignment and Uniformity

- Representation의 quality를 측정하기 위해 Alignment와 Uniformity를 측정

- Alignment
  - Positive pair간의 거리가 얼마나 가까운지를 나타냄
  - 값이 작을수록 좋음
  - Positive pair인 P_{pos} 가 있을 때, pair의 embedding 사이의 거리를 계산
    - Representation을 정규화한 후 진행

$$\ell_{\text{align}} \triangleq \mathop{\mathbb{E}}_{(x,x^+)\sim p_{\text{pos}}} \|f(x) - f(x^+)\|^2.$$

- Uniformity
  - Embedding이 균일하게 분포하는지를 나타냄
  - 값이 작을수록 좋음
  - Embedding space가 hypersphere에서 넓고 고르게 분포하여 각 단어가 고유한 의미를 보존하는 것이 중요하기 때문에 uniformity를 측정함

$$\ell_{\text{uniform}} \triangleq \log \mathop{\mathbb{E}}_{x,y \overset{i.i.d.}{\sim} p_{\text{data}}} e^{-2\|f(x)-f(y)\|^2}$$

Experiments

- Unsupervised SimCSE의 positive를 위한 데이터 증강 방법에 따른 성능 비교

Data augmentation			STS-B
None (unsup. SimCSE)			**82.5**
Crop	*10%*	*20%*	*30%*
	77.8	71.4	63.6
Word deletion	*10%*	*20%*	*30%*
	75.9	72.2	68.2
Delete one word			75.9
w/o dropout			74.2
Synonym replacement			77.4
MLM 15%			62.2

- 자르기, 단어 삭제, 단어 교체 등의 일반적인 데이터 증강 방법이 모두 dropout noise를 능가하지 못했음

Experiments

- 두 개의 encoder를 사용하는 대신 하나의 encoder를 사용하는 것이 더 좋은 성능을 보임을 확인

Training objective	f_θ	$(f_{\theta_1}, f_{\theta_2})$
Next sentence	67.1	68.9
Next 3 sentences	67.4	68.8
Delete one word	75.9	73.1
Unsupervised SimCSE	**82.5**	80.7

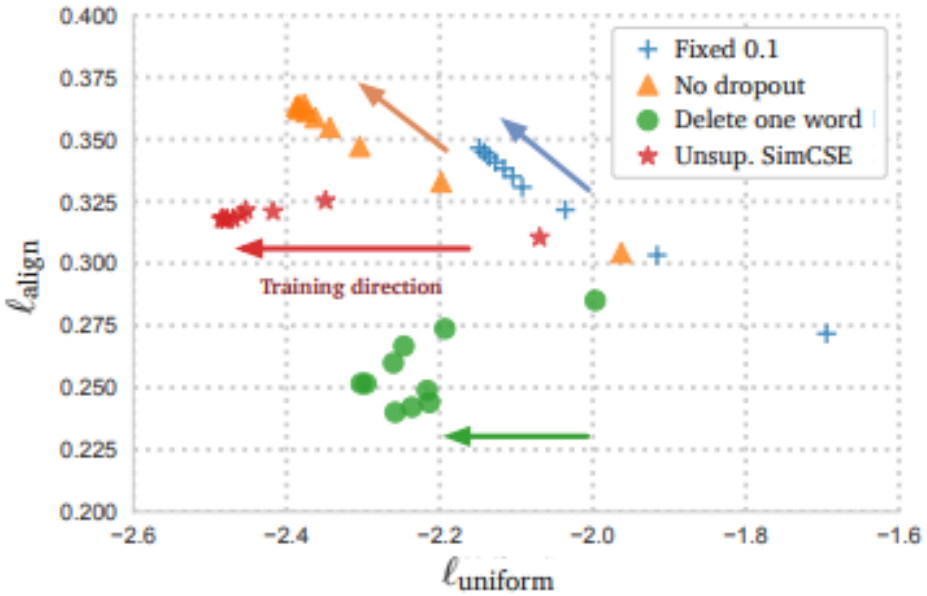
- 다양한 dropout mask를 실험

p	*0.0*	*0.01*	*0.05*	*0.1*
STS-B	71.1	72.6	81.1	**82.5**
p	*0.15*	*0.2*	*0.5*	*Fixed 0.1*
STS-B	81.4	80.5	71.0	43.6

  - Transformers의 기본 dropout인 0.1이 가장 좋은 성능을 보임
  - Fixed 0.1 (= 기본 dropout인 0.1을 사용하지만 pair에 동일한 dropout mask를 적용) 에서 급격한 성능 저하를 보임

Experiments

- Alignment와 Uniformity 실험
  - 훈련 중 10단계마다 시각화



- 모든 모델의 uniformity가 향상됨
- No dropout, Fixed 0.1은 alignment가 안 좋아졌지만, Unsupervised SimCSE는 alignment가 유지됨

Experiments

- STS (semantic textual similarity) task
  - 7개의 STS task에서 실험 진행
  - Unsupervised: 영어 Wikipedia에서 무작위로 선택된 문장(10^6)으로 학습
  - Supervised: MNLI와 SNLI dataset의 조합(314k)로 학습

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Unsupervised models								
GloVe embeddings (avg.)♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$\text{BERT}_{\text{base}}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$\text{BERT}_{\text{base}}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$\text{BERT}_{\text{base}}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-$\text{BERT}_{\text{base}}$♡	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-$\text{BERT}_{\text{base}}$	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
∗ SimCSE-$\text{BERT}_{\text{base}}$	**68.40**	**82.41**	**74.38**	**80.91**	**78.56**	**76.85**	**72.23**	**76.25**
$\text{RoBERTa}_{\text{base}}$ (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
$\text{RoBERTa}_{\text{base}}$-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
DeCLUTR-$\text{RoBERTa}_{\text{base}}$	52.41	75.19	65.52	77.12	78.63	72.41	**68.62**	69.99
∗ SimCSE-$\text{RoBERTa}_{\text{base}}$	**70.16**	**81.77**	**73.24**	**81.36**	**80.65**	**80.22**	68.56	**76.57**
∗ SimCSE-$\text{RoBERTa}_{\text{large}}$	**72.86**	**83.99**	**75.62**	**84.77**	**81.80**	**81.98**	**71.26**	**78.90**
Supervised models								
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
$\text{SBERT}_{\text{base}}$♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
$\text{SBERT}_{\text{base}}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
$\text{SBERT}_{\text{base}}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
CT-$\text{SBERT}_{\text{base}}$	74.84	83.20	78.07	83.84	77.93	81.46	76.42	79.39
∗ SimCSE-$\text{BERT}_{\text{base}}$	**75.30**	**84.67**	**80.19**	**85.40**	**80.82**	**84.25**	**80.39**	**81.57**
$\text{SRoBERTa}_{\text{base}}$♣	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
$\text{SRoBERTa}_{\text{base}}$-whitening	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
∗ SimCSE-$\text{RoBERTa}_{\text{base}}$	**76.53**	**85.21**	**80.95**	**86.03**	**82.57**	**85.83**	**80.50**	**82.52**
∗ SimCSE-$\text{RoBERTa}_{\text{large}}$	**77.46**	**87.27**	**82.36**	**86.66**	**83.93**	**86.70**	**81.95**	**83.76**

Experiments

- 다양한 pooling 방법에 따른 성능 비교

Pooler	Unsup.	Sup.
[CLS]		
w/ MLP	81.7	**86.2**
w/ MLP (train)	**82.5**	85.8
w/o MLP	80.9	**86.2**
First-last avg.	81.2	86.1

- Unsupervised: 훈련 중에만 MLP로 하는 것이 가장 성능이 좋았음
- Supervised: 어떤 방법을 사용하든 상관 없었음

Experiments

- Hard negative의 영향
  - Contradiction와 neutral hypothese를 hard negative로 같이 사용해봤을 때 α값을 다르게 하면서 실험

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}+\alpha^{\mathbb{1}_i^j}e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)},$$

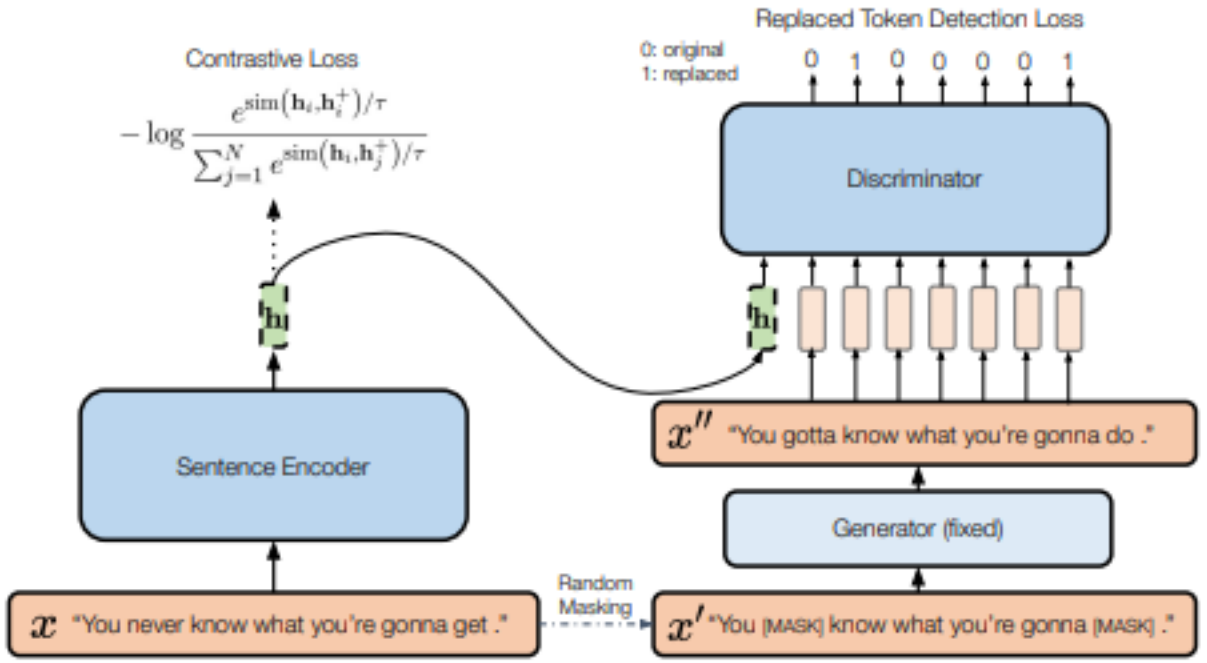
  - 그 결과, $\alpha = 1, 2$일 때 가장 성능이 좋았음

Hard neg	N/A	Contradiction			Contra.+ Neutral
α	-	0.5	1.0	2.0	1.0
STS-B	84.9	86.1	**86.2**	**86.2**	85.3

DiffCSE

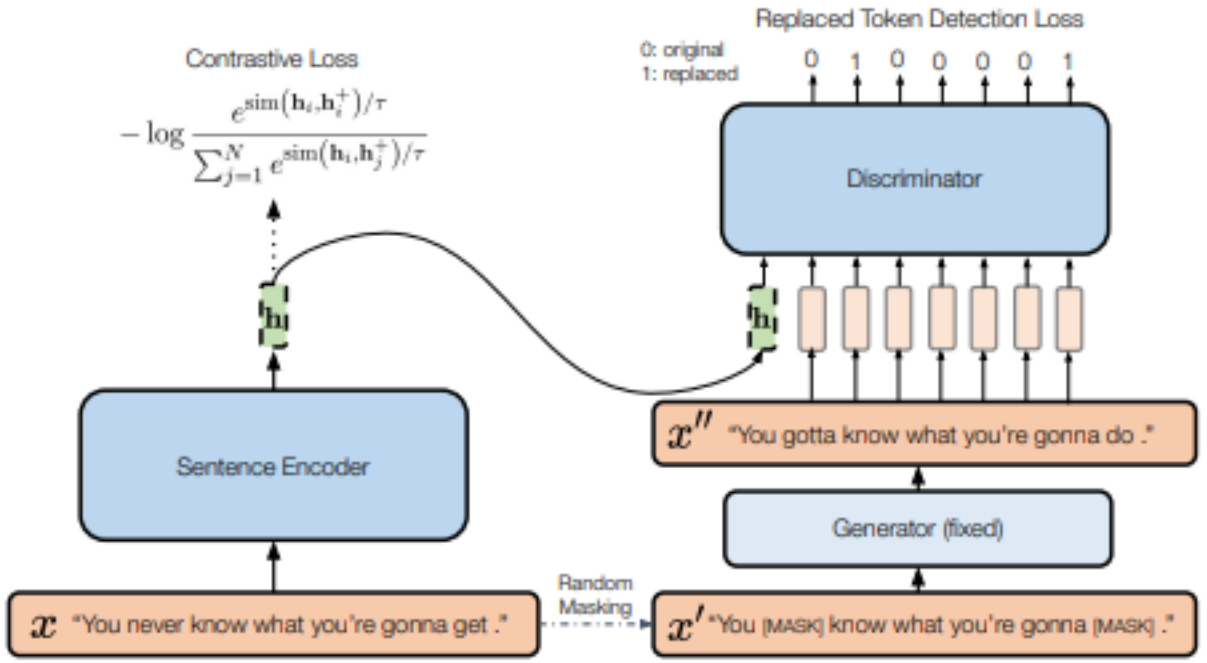
- DiffCSE: sentence representation을 학습하는 Unsupervised contrastive learning framework
- SimCSE와의 차이
  - SimCSE
    - MLM(Masked Language Model)이 모델의 성능을 떨어뜨렸기 때문에 MLM과 같은 transformation을 contrastive learning에 적용하는 것이 적절하지 않다고 주장했음
  - DiffCSE
    - MLM 기반의 transformation을 학습에 사용함
      - 그러기 위해 Conditional한 Discriminator를 활용
        - ELECTRA 모델은 Generator와 Discriminator로 이루어짐
        - DiffCSE는 ELECTRA 모델의 Discriminator를 Conditional한 Discriminator로 변형시켜주어 encoder를 학습함

DiffCSE



- Input sentence x 가 있을 때, x 를 contrastive learning 기법을 사용하여 sentence encoder를 통해 학습함
  - x와 동일한 encoder에 dropout mask만 다르게 적용하여 나온 hidden vector를 positive pair, N 크기만큼의 batch에서 다른 문장들과를 negative pair로 두어 학습

DiffCSE



1. Input sentence x 가 있을 때, 15% masking ratio로 토큰에 masking을 해줌
2. Masking을 해준 후, Generator를 통과하여 masking된 sentence를 생성
3. Sentence Encoder에서 나온 output hidden vector를 condition으로 걸어 (2)에서 생성된 sentence와 함께 Discriminator에 input으로 넣어줌
4. Discriminator는 Replaced Token Detection (RTD) 방식으로 학습을 진행

DiffCSE

- RTD 는 cross-entropy loss를 가지고 학습함
- ELECTRA 와는 달리 Generator는 freeze 시켜 학습을 하지 않음
- 오직 Discriminator만 학습함
- 학습 후 테스트 과정에서는 discriminator 부분을 버리고 sentence encoder만 활용함

- DiffCSE의 loss
  - Sentence encoder에서 나온 contrastive learning loss와 conditional한 Discriminator에서의 RTD loss를 결합하여 total loss를 정의함

    - Contrastive learning loss

$$\mathcal{L}_{\text{contrast}} = -\log \frac{e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_i^+\right)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_j^+\right)/\tau}},$$

    - RTD loss

$$\mathcal{L}_{\text{RTD}}^{x} = \sum_{t=1}^{T}\Big(-\mathbb{1}\left(x''_{(t)} = x_{(t)}\right)\log D\left(x'', \mathbf{h}, t\right) - \mathbb{1}\left(x''_{(t)} \neq x_{(t)}\right)\log\left(1 - D\left(x'', \mathbf{h}, t\right)\right)\Big)$$

    - DiffCSE의 Total loss

$$\mathcal{L} = \mathcal{L}_{\text{contrast}} + \lambda \cdot \mathcal{L}_{\text{RTD}}$$

DiffCSE

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
GloVe embeddings (avg.)♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$BERT_{base}$ (first-last avg.)◇	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$BERT_{base}$-flow◇	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$BERT_{base}$-whitening◇	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-$BERT_{base}$♡	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CMLM-$BERT_{base}$♠ (1TB data)	58.20	61.07	61.67	73.32	74.88	76.60	64.80	67.22
CT-$BERT_{base}$◇	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
SG-OPT-$BERT_{base}$†	66.84	80.13	71.23	81.56	77.17	77.23	68.16	74.62
SimCSE-$BERT_{base}$◇	68.40	82.41	74.38	80.91	78.56	76.85	**72.23**	76.25
∗ SimCSE-$BERT_{base}$(reproduce)	70.82	82.24	73.25	81.38	77.06	77.24	71.16	76.16
∗ DiffCSE-$BERT_{base}$	**72.28**	**84.43**	**76.47**	**83.90**	**80.54**	**80.59**	71.23	**78.49**
$RoBERTa_{base}$ (first-last avg.)◇	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
$RoBERTa_{base}$-whitening◇	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
DeCLUTR-$RoBERTa_{base}$◇	52.41	75.19	65.52	77.12	78.63	72.41	68.62	69.99
SimCSE-$RoBERTa_{base}$◇	**70.16**	81.77	73.24	81.36	80.65	80.22	68.56	76.57
∗ SimCSE-$RoBERTa_{base}$(reproduce)	68.60	81.36	73.16	81.61	80.76	80.58	68.83	76.41
∗ DiffCSE-$RoBERTa_{base}$	70.05	**83.43**	**75.49**	**82.81**	**82.12**	**82.38**	**71.19**	**78.21**

-> SoTA 달성

Thank You

감사합니다.