# Debiased Contrastive Learning of Unsupervised Sentence Representations

Authors: Kun Zhou, Beichen Zhang, Wayne Xin Zhao, Ji-Rong Wen

Venue: ACL 2022

# 1. Introduction

- 최근 pre-trained language models(PLMs) 이 다양한 NLP task에서 높은 성능을 보이면서, semantic한 표현을 얻기 위해 많이 사용됨
- 하지만, 어떤 연구들에서 PLMs에서 얻은 sentence representations이 균일하게 분배되어 있지 않고, vector 공간에서 narrow cone 모양으로 나타남을 발견함. (Anistoropy함)
  → 이런 현상은 표현성을 크게 제한할 수 있음


→ 이런 문제를 해결하기 위해 contrastive learning을 사용하여 PLMs로 얻는 sentence representations을 개선하고자 함

# 1. Introduction

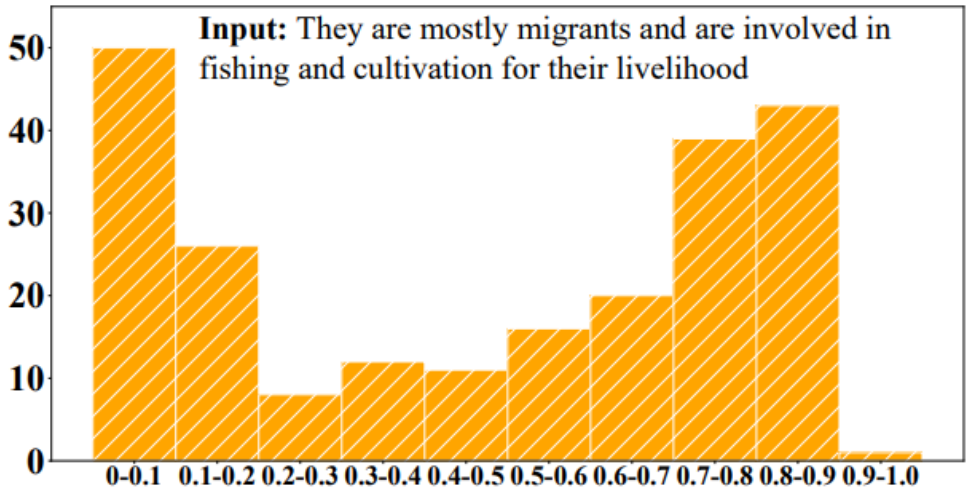
- Contrastive learning이란?
  - Alignment를 개선하기 위해 Positive끼리는 가깝게, uniformity를 개선하기 위해 negative끼리는 멀게 학습시키는 방법
    - Representation은 불필요한 디테일에 불변해야 하고, 최대한 많은 정보를 보존해야 함

    - Alignment
      - positive끼리의 거리가 얼마나 가까운지를 나타냄
      - 유사한 sample은 유사한 Representation 를 가진다.
    - Uniformity
      - representation이 균일하게 분포하는지를 나타냄
      - Representation의 분포는 정보를 최대한 보존한다.

  - Positive sample은 이전 연구들 중 가장 높은 성능을 보인 데이터 증강 기법을 사용
  - Negative sample은 배치 안에서 random하게 샘플링 → in batch negative
    - Labeling data가 부족하기 때문
    - 배치 안에서 random하게 샘플링 하는 방법은 **간단하고 편리하지만 sampling bias를 일으킨다는 문제점이 있음**

# 1. Introduction

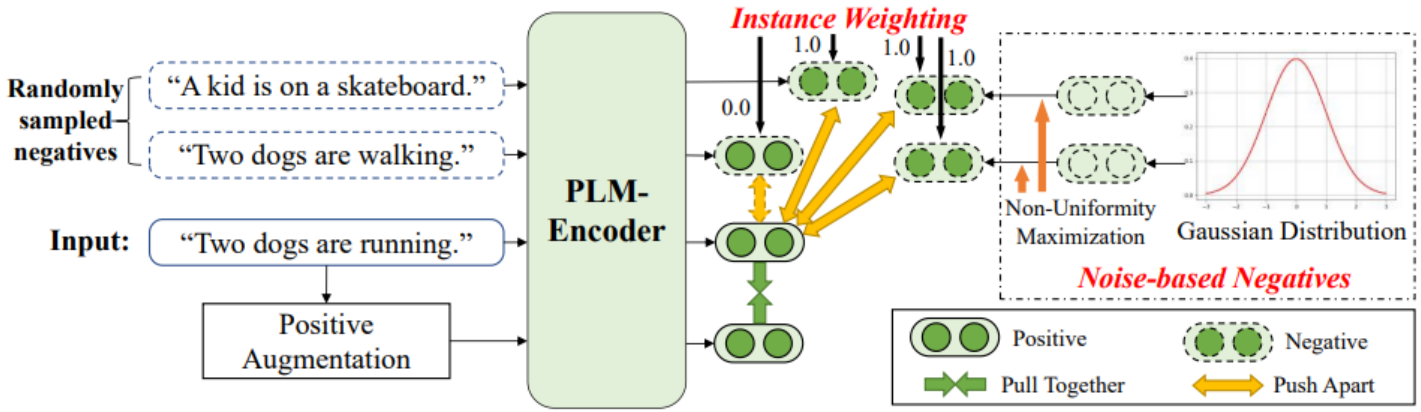
- Negative sampling의 문제점
  1. Sampling된 negative data가 original sentence와 의미적으로 유사한 false negative일 수 있음




  2. Sampling된 negative data의 representations은 narrow cone에서 나오는 representation임
     → 표현 공간의 전체적인 의미를 반영하기에 충분하지 않음

# 2. Method

- 이를 해결하기 위해 DCLR(Debiased Contrastive Learning of unsupervised sentence Representations) 을 제안




1. Gaussian noise 추가하여 negative 생성

2. Instance weighting
   - In batch negatives 중에서 original sentence와의 유사도를 확인
   - 유사도가 높은 negative를 False negative라고 보고 이에 대한 weight을 지정
   - 유사도가 높을 수록 낮은 weight을 줌

→ 최종적으로는 이 두가지 모두를 사용

# 2. Method

- Generating Noise-based Negatives
  - Random한 Gaussian 분포를 통해 k개의 새로운 negative set 초기화

$$\{\hat{h}_1, \hat{h}_2, \cdots, \hat{h}_k\} \sim \mathcal{N}(0, \sigma^2), \qquad (1)$$

$(h_i, h_i^+)$ as:

$$L_U(h_i, h_i^+, \{\hat{h}\}) = -\log \frac{e^{\text{sim}(h_i, h_i^+)/\tau_u}}{\sum_{\hat{h}_j \in \{\hat{h}_j\}} e^{\text{sim}(h_i, \hat{h}_i)/\tau_u}}, \quad (2)$$

  - 이에 맞는 경사하강법 최적화

$$\hat{h}_j = \hat{h}_j + \beta g(\hat{h}_j)/||g(\hat{h}_j)||_2, \qquad (3)$$
$$g(\hat{h}_j) = \nabla_{\hat{h}_j} L_U(h_i, h_i^+, \{\hat{h}\}), \qquad (4)$$

원래 gradient ascent 식
$x' = x + \beta g(x)$
$g(x) = \nabla f(x)$

    - $\beta$ : learning rate

# 2. Method

- Instance weighting
  - Negative와 original sentence의 유사도가 threshold보다 크면 0, 작으면 1로 weight 지정

$$\alpha_{h^-} = \begin{cases} 0, \mathrm{sim}_C(h_i, h^-) \geq \phi \\ 1, \mathrm{sim}_C(h_i, h^-) < \phi \end{cases} \quad (5)$$

  - Gaussian noise-based negative, Instance weighting 방법을 적용한 최종 loss function 정의

$$L = -\log \frac{e^{\mathrm{sim}(h_i, h_i^+)/\tau}}{\sum_{h^- \in \{\hat{h}\} \cup \{\tilde{h}^-\}} \alpha_{h^-} \times e^{\mathrm{sim}(h_i, h^-)/\tau}}, \quad (6)$$

Gaussian noise-based negative

Instance weighting

# 3. Experiments

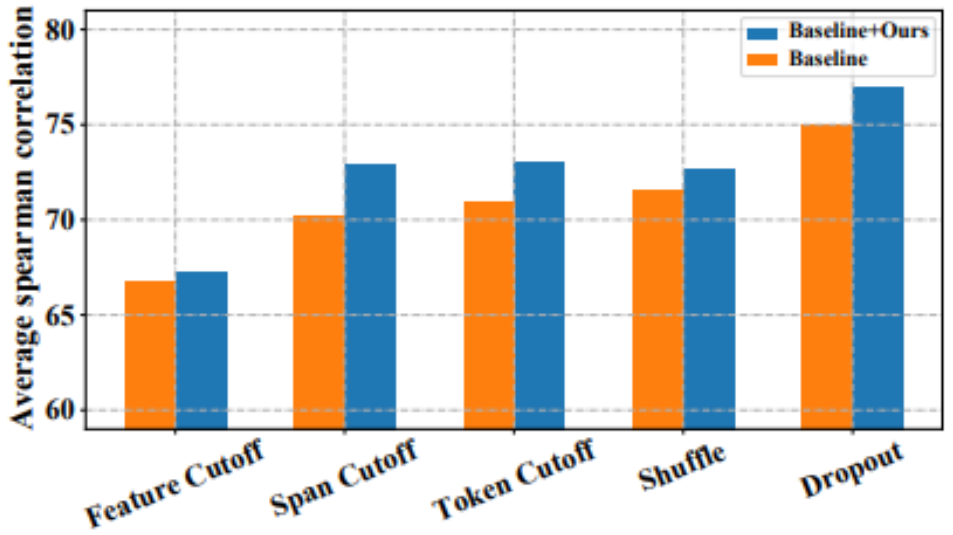
- 3-1) STS(Semantic Textual Similarity) Tasks
  - GloVe, USE
  - CLS, Mean, First-Last AVG
  - Flow, Whitening
  - Contrastive(BT), ConSERT, SG-OPT, SimCSE

  - **학습 데이터**: wikipedia에서 랜덤하게 100만 개의 문장 샘플링
  - **Backbone**: BERT-base, RoBERTa-base, BERT-large, RoBERTa-large
  - **Epoch**:3, **Temperature**: 0.05
  - **Optimizer**: Adam
  - **Batch size**: (base)128, (large)256
  - **Learning rate**: (base, BERT-large) 3e-5, (RoBERTa-large) 1e-5
  - **Instance weighting threshold**: 각각 0.9, 0.85, 0.9, 0.85
  - **Gaussian k**: (k*batch_size) 각각 k – 1, 2.5, 4, 5

| | Models | STS12 | STS13 | STS14 | STS15 | STS16 | STS-B | SICK-R | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| **Non-BERT** | GloVe (avg.)† | 55.14 | 70.66 | 59.73 | 68.25 | 63.66 | 58.02 | 53.76 | 61.32 |
| | USE† | 64.49 | 67.80 | 64.61 | 76.83 | 73.18 | 74.92 | 76.69 | 71.22 |
| **BERT-base** | CLS† | 21.54 | 32.11 | 21.28 | 37.89 | 44.24 | 20.30 | 42.42 | 31.40 |
| | Mean† | 30.87 | 59.89 | 47.73 | 60.29 | 63.73 | 47.29 | 58.22 | 52.57 |
| | First-Last AVG‡. | 39.70 | 59.38 | 49.67 | 66.03 | 66.19 | 53.87 | 62.06 | 56.70 |
| | +flow‡ | 58.40 | 67.10 | 60.85 | 75.16 | 71.22 | 68.66 | 64.47 | 66.55 |
| | +whitening‡ | 57.83 | 66.90 | 60.90 | 75.08 | 71.31 | 68.24 | 63.73 | 66.28 |
| | +Contrastive (BT)† | 54.26 | 64.03 | 54.28 | 68.19 | 67.50 | 63.27 | 66.91 | 62.63 |
| | +ConSERT | 64.64 | 78.49 | 69.07 | 79.72 | 75.95 | 73.97 | 67.31 | 72.74 |
| | +SG-OPT† | 66.84 | 80.13 | 71.23 | 81.56 | 77.17 | 77.23 | 68.16 | 74.62 |
| | +SimCSE | 68.40 | 82.41 | 74.38 | 80.91 | **78.56** | 76.85 | **72.23** | 76.25 |
| | +DCLR (Ours) | **70.81** | **83.73** | **75.11** | **82.56** | 78.44 | **78.31** | 71.59 | **77.22** |
| **BERT-large** | CLS† | 27.44 | 30.76 | 22.59 | 29.98 | 42.74 | 26.75 | 43.44 | 31.96 |
| | Mean† | 27.67 | 55.79 | 44.49 | 51.67 | 61.88 | 47.00 | 53.85 | 48.91 |
| | First-Last AVG | 57.73 | 61.17 | 61.18 | 68.07 | 70.25 | 59.59 | 60.34 | 62.62 |
| | +flow† | 62.82 | 71.24 | 65.39 | 78.98 | 73.23 | 72.72 | 63.77 | 70.07 |
| | +whitening | 64.34 | 74.60 | 69.64 | 74.68 | 75.90 | 72.48 | 60.80 | 70.35 |
| | +Contrastive (BT)† | 52.04 | 62.59 | 54.25 | 71.07 | 66.71 | 63.84 | 66.53 | 62.43 |
| | +ConSERT | 70.69 | 82.96 | 74.13 | 82.78 | 76.66 | 77.53 | 70.37 | 76.45 |
| | +SG-OPT† | 67.02 | 79.42 | 70.38 | 81.72 | 76.35 | 76.16 | 70.20 | 74.46 |
| | +SimCSE | 70.88 | 84.16 | 76.43 | 84.50 | 79.76 | 79.26 | 73.88 | 78.41 |
| | +DCLR (Ours) | **71.87** | **84.83** | **77.37** | **84.70** | **79.81** | **79.55** | **74.19** | **78.90** |
| **RoBERTa-base** | CLS† | 16.67 | 45.57 | 30.36 | 55.08 | 56.98 | 45.41 | 61.89 | 44.57 |
| | Mean† | 32.11 | 56.33 | 45.22 | 61.34 | 61.98 | 54.53 | 62.03 | 53.36 |
| | First-Last AVG‡ | 40.88 | 58.74 | 49.07 | 65.63 | 61.48 | 58.55 | 61.63 | 56.57 |
| | +whitening‡ | 46.99 | 63.24 | 57.23 | 71.36 | 68.99 | 61.36 | 62.91 | 61.73 |
| | +Contrastive (BT)† | 62.34 | 78.60 | 68.65 | 79.31 | 77.49 | 79.93 | 71.97 | 74.04 |
| | +SG-OPT† | 62.57 | 78.96 | 69.24 | 79.99 | 77.17 | 77.60 | 68.42 | 73.42 |
| | +SimCSE | **70.16** | 81.77 | 73.24 | 81.36 | 80.65 | 80.22 | 68.56 | 76.57 |
| | +DCLR (Ours) | 70.01 | **83.08** | **75.09** | **83.66** | **81.06** | **81.86** | **70.33** | **77.87** |
| **RoBERTa-large** | CLS† | 19.25 | 22.97 | 14.93 | 33.41 | 38.01 | 12.52 | 40.63 | 25.96 |
| | Mean† | 33.63 | 57.22 | 45.67 | 63.00 | 61.18 | 47.07 | 58.38 | 52.31 |
| | First-Last AVG | 58.91 | 58.62 | 61.44 | 69.05 | 65.23 | 59.38 | 58.84 | 61.64 |
| | +whitening | 64.17 | 73.92 | 71.06 | 76.40 | 74.87 | 71.68 | 58.49 | 70.08 |
| | +Contrastive (BT)† | 57.60 | 72.14 | 62.25 | 71.49 | 71.75 | 77.05 | 67.83 | 68.59 |
| | +SG-OPT† | 64.29 | 76.36 | 68.48 | 80.10 | 76.60 | 78.14 | 67.97 | 73.13 |
| | +SimCSE | 72.86 | 83.99 | 75.62 | 84.77 | 81.80 | 81.98 | 71.26 | 78.90 |
| | +DCLR (Ours) | **73.09** | **84.57** | **76.13** | **85.15** | **81.99** | **82.35** | **71.80** | **79.30** |

# 3. Experiments

- 3-2) Performance comparison using different positive augmentation strategies
  - 모든 방법에서 DCLR을 적용한 것이 더 성능이 좋음을 보임



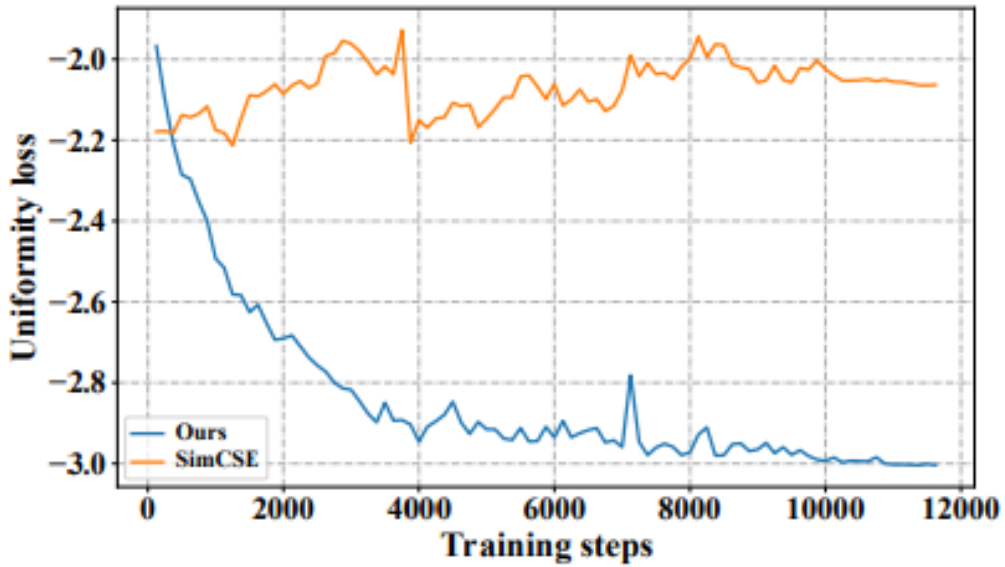
# 3. Experiments

- 3-3) Ablation
  - Instance Weighting을 제거했을 때, 더 큰 하락 폭을 보임
  - Random Noise, Knowledge Distillation, Self Instance Weighting 모두 DCLR보다 낮음
    - Random Noise: gradient-based optimization 없이 noise-based negative 생성
    - Knowledge Distillation: SimCSE를 teacher model로 사용
    - Self Instance Weighting: weight을 생성하기 위해 자기 자신을 보조 모델로 사용

| Model | STS-Avg. |
|---|---|
| BERT-base+Ours | **77.22** |
| w/o Noise-based Negatives | 76.17 |
| w/o Instance Weighting | 76.31 |
| BERT-base+Random Noise | 75.22 |
| BERT-base+Knowledge Distillation | 75.05 |
| BERT-base+Self Instance Weighting | 73.93 |

# 3. Experiments

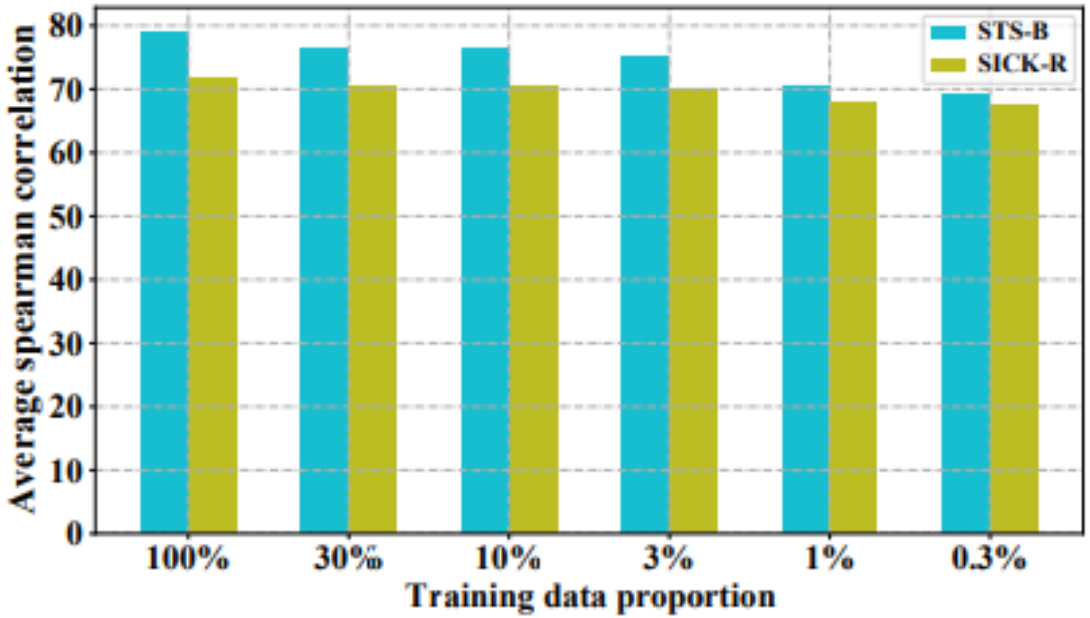
- 3-4) Uniformity Analysis
  - DCLR이 SimCSE보다 훨씬 빠르게 낮아지는 모습을 보임
  - Representation space에서 noise-based negative를 사용했기 때문
    - Gaussian noise-based negative가 uniformity 개선에 좋다는 것을 보임



# 3. Experiments

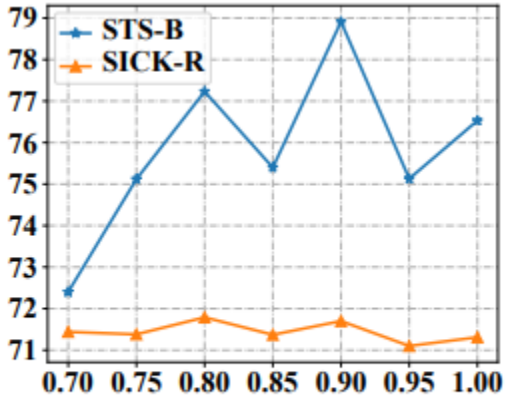
- 3-5) Performance under Few-shot Settings
  - Data가 부족할 때도 DCLR이 robust하고 reliable한 지 알기 위해 실험
  - Backbone model: BERT-base

  - 데이터가 줄어도 stable한 결과를 보였고, 데이터를 극도로 줄인 0.3%에서의 성능 차이는 9, 4 정도뿐이였음

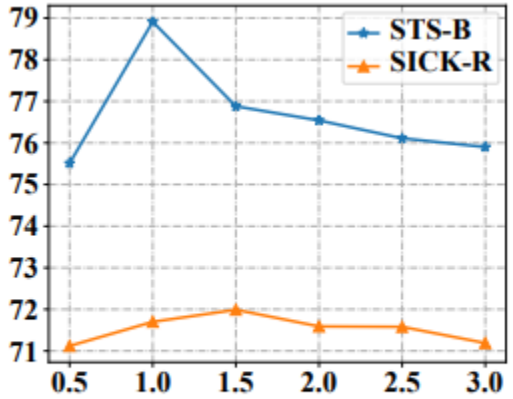


# 3. Experiments

- 3-6) Hyper-parameters Analysis
  - Weighting threshold
    - STS-B에서 너무 크거나 너무 작은 threshold는 오히려 성능 저하를 일으킴
    - Threshold가 0.9일 때 가장 좋은 성능을 보임
  - Negative Proportion
    - Noise-based negative의 수(k*batch_size)가 batch size와 가까울 때 가장 좋은 성능을 보임




(a) Weighting Threshold $\phi$




(b) Negative Proportion $k$

# 5. Conclusion

- DCLR은 random negative sampling에서 발생하는 sampling bias를 완화하기 위해 instance weighting 방법을 제안했음
- 또한, PLMs에서 얻은 representation이 anisotropy하기 때문에 이런 문제를 해결하기 위해 noise-based negative를 생성하여 사용하는 방법도 제안했음
- 두 가지 방법 모두 사용한 결과, 7개의 STS task에서 baseline model보다 좋은 성능을 보였음

# Thank You

감사합니다.