*mi*CSE: Mutual Information Contrastive Learning for Low-shot Sentence Embeddings

Tassilo Klein
SAP AI Research
tassilo.klein@sap.com

Moin Nabi
SAP AI Research
m.nabi@sap.com

Abstract

This paper presents *mi*CSE, a mutual information-based contrastive learning framework that significantly advances the state-of-the-art in few-shot sentence embedding. The proposed approach imposes alignment between the attention pattern of different views during contrastive learning. Learning sentence embeddings with *mi*CSE entails enforcing the structural consistency across augmented views for every sentence, making contrastive self-supervised learning more sample efficient. As a result, the proposed approach shows strong performance in the few-shot learning domain. While it achieves superior results compared to state-of-the-art methods on multiple benchmarks in few-shot learning, it is comparable in the full-shot scenario. This study opens up avenues for efficient self-supervised learning methods that are more robust than current contrastive methods for sentence embedding.[1]

1 Introduction

Measuring sentence similarity has been challenging due to the ambiguity and variability of linguistic expressions. The community's strong interest in the topic can be attributed to its applicability in numerous language processing applications, such as sentiment analysis, information retrieval, and semantic search (Pilehvar and Navigli, 2015; Iyyer et al., 2015). Language models perform well on these tasks but typically require fine-tuning on the downstream task and corpora (Reimers and Gurevych, 2019; Devlin et al., 2018; Pfeiffer et al., 2020; Mosbach et al., 2021). In terms of sentence embeddings, contrastive learning schemes have already been adopted successfully (van den Oord et al., 2018; Liu et al., 2021; Gao et al., 2021; Carlsson et al., 2021). The idea of contrastive learning is that positive and negative pairs are generated given a batch of samples. Whereas the positive pairs are obtained via augmentation, negative pairs are often created by random collation of sentences. Following the construction of pairs, contrastive learning forces the network to learn feature representations by pushing apart different samples (negative pairs) or pulling together similar ones (positive pairs). While some methods seek to optimize for selecting "hard" negative for negative pair generation (Zhou et al., 2022a), others investigated better augmentation techniques for positive pair creation. In this regard, many methods have been proposed to create augmentations to boost representation learning. Standard approaches for the augmentation aim at input *data level* (a.k.a *discrete* augmentation), which comprises word level operations such as swapping, insertion, deletion, and substitution (Xie et al., 2017; Coulombe, 2018; Wei and Zou, 2019). In contrast to that, *continuous* augmentation operates at the *representation level*, comprising approaches like interpolation or "mixup" on the embedding space (Chen et al., 2020; Cheng et al., 2020; Guo et al., 2019). Most recently, augmentation was also proposed in a more continuous fashion operating in a *parameter level* via simple techniques such as drop-out (Gao et al., 2021; Liu et al., 2021; Klein and Nabi, 2022) or random span masking (Liu et al., 2021). The intuition is that "drop-out" acts as minimal data augmentation, providing an expressive *semantic variation*. However, it will likely affect *structural alignment* across views. Since positive pairs are constructed from identical sentences, we hypothesize that the structural dependency over the views should be preserved by utilizing drop-out noise. Building on this idea, we maximize the *structural dependence* by enforcing distributional similarity over the attention values across the augmentation views. To this end, we employ maximization of the mutual information (MI) on the attention tensors of the positive pairs. However, since attention tensors can be very

[1] Source code and pre-trained models are available at: https://github.com/SAP-samples/acl2023-micse/

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6159–6177

high-dimensional, computing MI can quickly become a significant burden if not intractable. This paper proposes a simple solution to alleviate the computational burden of MI computation, which can be deployed efficiently. Similar to (Fan et al., 2020), we adopt the Log-Normal distribution to model attention. Empirical evidence confirms this model as a good fit while facilitating the optimization objective to be defined in closed form. In this case, mutual information can be provably reformulated as a function of correlation, allowing native GPU implementation. As discussed above, the proposed approach builds upon the contrastive learning paradigm known to suffer from model collapse. This issue becomes even more problematic when enforcing MI on the attention level, as it tightens the positive pairs via regularizing the attention. Therefore the selection of negative pairs becomes more critical in our setup. To this end, we utilize momentum contrastive learning to generate harder negatives (He et al., 2020). A "tighter" binding on positive pairs and repulsion on "harder" negative pairs empowers the proposed contrastive objective, yielding more powerful representations.

Combining ideas from momentum contrastive learning and attention regularization, we propose *mi*CSE, a conceptually simple yet empirically powerful method for sentence embedding, with the goal of integrating semantic and structural information of a sentence in an information-theoretic and Transformer-specific manner. We conjecture the relation between attention maps and a form of syntax to be the main driver behind the success of our approach. We speculate that our proposed method injects structural information into the model as an inductive bias, facilitating representation learning with fewer samples. The adopted structural inductive biases provide a "syntactic" prior as an implicit form of supervision during training (Wilcox et al., 2020), which promotes few-shot learning capabilities in neural language models. To validate this, we introduced a low-shot setup for training sentence embeddings. In this benchmark, we finetune the language model *only* with a small number of training samples. Note that this is a very challenging setup. The inherent difficulty can be attributed to the need to mitigate the domain shift in the low-shot self-supervised learning scheme. We emphasize the importance of this task, as in many real-world applications, only small datasets are often available. Such cases include NLP for low-resource languages or expert-produced texts (e.g., medical records by doctors), personalized LM for social media analysis (e.g., personalized hate speed recognition on Twitter), etc. Our proposed method significantly improves over the state-of-the-art in the low-shot sentence embedding benchmark. This is the first work that explores how to combine semantic and structural information through attention regularization and empirically demonstrates this benefit for low-shot sentence embeddings.

Previous works: Recently, VaSCL (Zhang et al., 2022a), ConSERT (Yan et al., 2021a), PCL (Wu et al., 2022a) and (Chuang et al., 2022) proposed contrastive representation learning with diverse augmentation strategies on positive pair. However, we proposed a principled approach for enforcing *alignment* in positive pairs at contrastive learning without discretely augmenting the data. Similar to us, ESimCSE (Wu et al., 2021) and MoCoSE (Cao et al., 2022a) proposed to exploit a momentum contrastive learning model with negative sample queue for sentence embedding to boost *uniformity* of the representations. However, unlike us, they do not enforce any further tightening objective on the positive pairs nor consider few-shot learning. Very recently, authors in InforMin-CL (Chen et al., 2022) and InfoCSE (Wu et al., 2022b)proposed information minimization-based contrastive learning. Specifically, the authors propose to minimize the information entropy between positive embeddings generated by drop-out augmentation. Our model differs from this paper and the method in (Bachman et al., 2019; Yang et al., 2021; Zhang et al., 2020; Sordoni et al., 2021; Wu et al., 2020), which focuses on using mutual information for self-supervised learning. A key difference compared to these methods is that they estimate MI directly on the representation space. In contrast, our method computes the MI on attention. Other related work include (Zhang et al., 2022b; Zhou et al., 2022b; Zhang et al., 2022c; Liu et al., 2022).

The contributions of the proposed work are: **First**, we propose to inject structural information into language models by adding an attention-level objective. **Second**, we introduce Attention Mutual Information (AMI), a sample-efficient self-supervised contrastive learning. **Third**, we introduce low-shot learning for sentence embedding. We show that our method performs comparably to the state-of-the-art in the full-shot scenario and significantly better in few-shot learning.

2 Method

The proposed approach aims to exploit the structure of the sentences in a contrastive learning scheme. Compared to conventional contrastive learning that solely operates at the level of *semantic* similarity in the embedding space, the proposed approach injects *structural* information into the model. This is achieved by regularizing the attention space of the model during training. We let $\mathcal{D}$ denote a dataset consisting of string sequences (sentences) from corpus $\mathcal{X}$ with $\mathcal{D} = \{x_1, x_2, ..., x_{|\mathcal{X}|}\}$, where we assume x_i to be a tokenized sequence of length n with $x_i \in \mathbb{N}^n$. For mapping the input data to the embedding space, we use a bi-encoder f_θ parametrized by θ. Bi-encoders entail the computation of embeddings for similarity comparison, whereby each sentence in a pair is encoded separately. Hence, the instantiation of a bi-encoder on augmented input data induces multiple views. For the following, we let $v \in \{1, 2\}$ denote the index of the view, where each view corresponds to a different augmentation. Consequently, encoding a data batch $\mathcal{D}_b$ yields embedding matrices $E_v \in \mathbb{R}^{|\mathcal{D}_b| \times U}$, where U denotes the dimensionality of the embeddings. Employing a Transformer, encoding the input data yields the embedding matrices and the associated attention tensors W_v. Then learning representation of the proposed approach entails the optimization of a joint loss:

$$\min_{\theta} \mathcal{L}_C(E_1, E_2) + \mathcal{L}_D(W_1, W_2) \tag{1}$$

with $(E_1, W_1), (E_2, W_2) = f_\theta(\mathcal{D}_b)$. Here, $\mathcal{L}_C$ is responsible for the semantic alignment, corresponding to the standard InfoNCE (van den Oord et al., 2018) loss that seeks to pull positive pairs close together while pushing away negative pairs in the embedding space. In contrast, $\mathcal{L}_D$ is responsible for the syntactic alignment, operating on the attention space. However, in comparison to $\mathcal{L}_D$ is employed only on positive pairs' attention tensors.

2.1 Embedding-level Momentum-Contrastive Learning (InfoNCE)

The InfoNCE-loss seeks to pull positive pairs together in the embedding space while pushing negative pairs apart. Specifically, InfoNCE on embeddings pushes for the similarity of each sample and its corresponding augmented embedding. Negatives pairs are constructed in two ways, reflected by the two terms in the denominator of Eq. 2. First, in-batch negative pairs are constructed by pairing each sentence with another random sentence (sharing no semantic similarity), pushing for dissimilarity. Second, using embeddings obtained from a momentum encoder known as MoCo (He et al., 2020; Cao et al., 2022a). The momentum encoder is a replication of the encoder f_θ, whose parameters are updated more slowly. Specifically, while the parameters of f_θ encoder are updated via back-propagation, the parameters of the momentum encoder are updated using an exponential moving average from the former. The negative embeddings are produced from samples from previous batches, which are stored in queue $\mathcal{Q}$ and are forward-passed through the momentum encoder. Then the InfoNCE (van den Oord et al., 2018) loss ($\mathcal{L}_C$) is defined as:

$$-\sum_{i}^{|\mathcal{D}_b|} \log \frac{d(\boldsymbol{e}_i, {}^{+}\mathbf{e}_i)}{\sum\limits_{j:i\neq j}^{|\mathcal{D}_b|} d(\boldsymbol{e}_i, \boldsymbol{e}_j) + \sum\limits_{j}^{|\mathcal{Q}|} d(\boldsymbol{e}_i, \boldsymbol{q}_j)}, \tag{2}$$

where $\boldsymbol{e}_i \in E_1$ and ${}^{+}\boldsymbol{e}_i \in E_2$ denote the embeddings of different augmentations of x_i. Furthermore, $d(\boldsymbol{x}, \boldsymbol{y}) = \exp(sim(\boldsymbol{x}, \boldsymbol{y})/\tau)$ with $sim(.)$ the cosine similarity metric, q_j denoting representations obtained from momentum encoder, and $\tau \in \mathbb{R}$ is a temperature scalar.

2.2 Attention-level Mutual Information (AMI)

Preliminaries and notations: We first briefly review the attention mechanism and explain the notation used in the rest of this section. A Transformer stack consists of a stack of L layers, with input data cascading up the layer stack. Each layer comprises a self-attention module and a feed-forward network in its simplest form. Passing sentences through the encoder stack entails simultaneous computation of attention weights. These attention weights indicate the relative importance of every token. To this end, key-value pairs are computed for each token of the input sequence within each self-attention module. This entails the computation of three different matrices: key matrix K, value matrix V, and query matrix Q. The values of the attention weights W are obtained according to $W = \text{softmax}(f(Q, K)) \in \mathbb{R}^{n \times n}$, where $f(.)$ is a scaled dot-product. Output features are then generated as obtained according to WV. To attend to different sub-spaces (Vaswani et al., 2017) simultaneously, the attention mechanism is replicated H times, referred to as multi-head attention. During training the encoder, the self-attention tensors W values are subject to a random

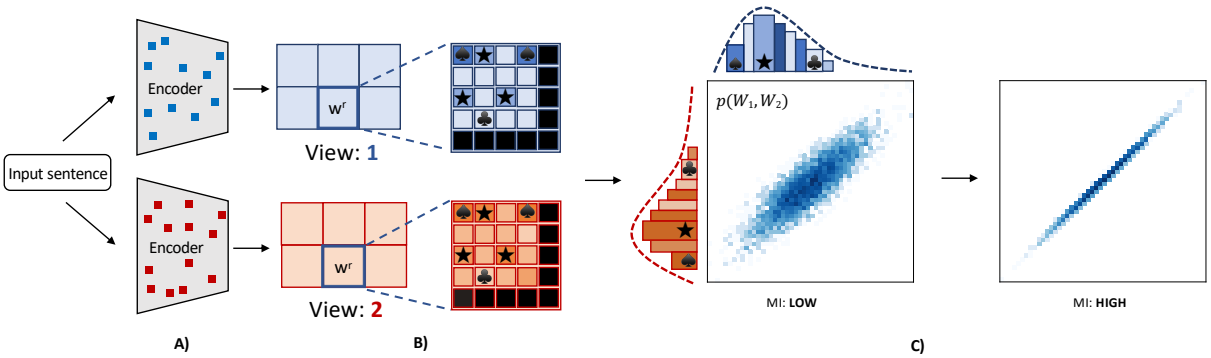


FIGURE 1. **Schematic illustration of AMI pipeline: A)** Starting from an input sentence, two views are created by drop-out augmentation (indicated with red and blue). Each view produces a different attention tensor. **B)** The attention tensor is sliced into tiles, and sampling is then conducted on aligned tiles. High correlation across attention-aligned tiles allows sampling without a significant shift in the attention distribution at a modest accuracy compromise. **C)** Subsequently, assuming a log-normal distribution of the attention tensor, the joint distribution is computed, and mutual information is maximized.

deterministic process, with randomness arising due to drop-out. Hence, the proposed approach seeks to optimize structural alignment by maximizing mutual information between the attention tensors $W_v = [\boldsymbol{w}_1, ..., \boldsymbol{w}_{|\mathcal{D}_b|}]$ of the augmentation views. We propose a four-step pipeline to regularize the joint attention space. For a schematic illustration of the AMI pipeline, see Fig. 1.

1) Attention Tensor Slicing: Given that augmentation has different effects on the attention distribution depending on the depth (layer) and the position (head) in the Transformer stack, we propose to slice the attention tensor. Chunking the attention has multiple advantages. On the one hand, this allows for preserving the locality of distribution change. This is important as it can be empirically observed that distribution divergence between views decreases with increasing depth in the encoding stack. On the other hand, restricting the space permits using a simple distributional model such as bivariate distribution compared to a mixture distribution for the whole stack.

For the sake of economy in notation and avoid notational clutter, we will restrict the attention tensor of a single encoded sample in the following. To this end, a slicing function $\pi : \mathbb{R}^{L\times H\times n\times n} \rightarrow \mathbb{R}^{R\times n\times n}$ cuts the attention tensor for each input sample into R (indexed) elements: $\pi(\boldsymbol{w}_i) = [\boldsymbol{w}_i^1, ..., \boldsymbol{w}_i^R] \in \mathbb{R}^{n\times n}$ with $\boldsymbol{w}_i^r = (w_{j,k})_{1\leq j,k\leq n}$ and $r \in R$. For a schematic illustration of how the attention tensor is sliced into tiles, see Fig. 2.

2) Attention Sampling: Different sentences in the batch are typically in token sequences of different lengths. To accommodate the different lengths and facilitate efficient training, sequences are typically padded with `[PAD]`-token for length equality. Although this allows for efficient batch encoding on GPU, attentions arising from `[PAD]`-tokens have to be discarded when looking at statistical relationships. To accommodate for the different lengths of tokenized sequences, perform a sampling step for attention values within each grid cell $\boldsymbol{w}_i^r$. To this end, we leverage multinomial distribution $P_{mult}(p_1, .., p_{n^2})$, where s correspond to the

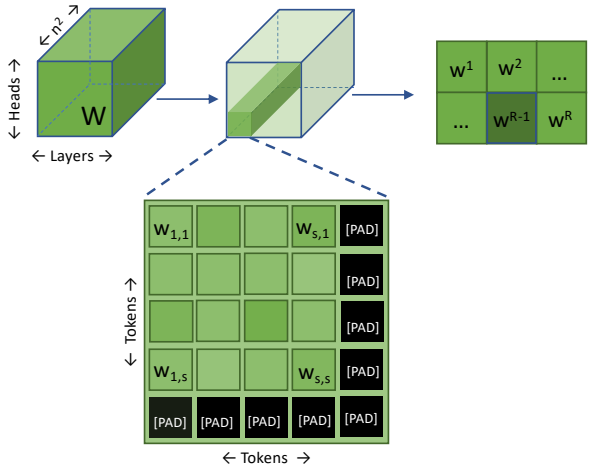


FIGURE 2. **Attention tensor slicing:** Instantiating a transformer stack on an input yields an attention tensor W comprising token attention weights across layers and heads. Slicing the attention entails tiling the tensor. Batch processing of sequences of different lengths is accommodated by padding (`[PAD]`).

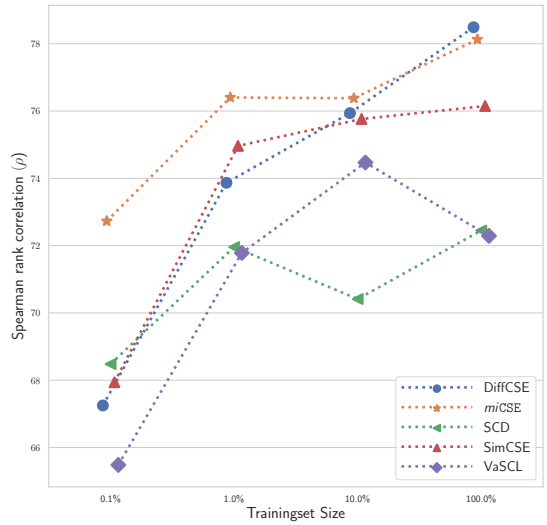


FIGURE 3. Few-shot performance of different algorithms. DiffCSE (Chuang et al., 2022) (—), *mi*CSE (—), SCD (Klein and Nabi, 2022) (—), SimCSE (Gao et al., 2021) (—), VaSCL (Zhang et al., 2022a) (—). Performance is shown in Spearman's correlation average of STS. Training set size: 0.1%, 1.0%, 10.0%, 100.0% of the data. Best viewed in color.

number of non-padding tokens with $1 \leq s \leq n$. Specifically, we sample from the s^2 attention values pool, each with a probability of $\frac{1}{s^2}$, with the remaining elements associated with probability 0. As a result, we obtain a set $J_r = \{j_1, ..., j_m\}$ consisting of m indices of the attention tensors for each slice $r \in R$:

$$J_r \sim P_{mult}(\underbrace{1/s^2, ..., 1/s^2}_{1, .., s^2}, \overbrace{0, ..., 0}^{(n-s)^2, ..., n^2}) \quad (3)$$

It should be noted that for the same slice r across the views, the same index set is used for sampling: $\tilde{\boldsymbol{w}}^r = \bigcup_{j \in J_r} \boldsymbol{w}^r[j]$ and ${}^+\tilde{\boldsymbol{w}}^r = \bigcup_{j \in J_r} {}^+\boldsymbol{w}^r[j]$.

3) Attention Mutual Information Estimation: We propose using mutual information to measure the similarity of attention patterns for different views. Specifically, we follow (Fan et al., 2020) and adopt the Log-Normal distribution for modeling the attention distribution, which is prudent for several reasons. First, Empirical observation confirms attention asymmetry. Second, by utilizing a non-symmetric distribution, it becomes possible to break down the attention tensor W into K and Q, thereby allowing for non-symmetrical attention. Third, adopting the log-normal models facilitates the optimization objective to be defined in closed form and hence easy to optimize, particularly on GPUs. Mutual information for two normally distributed tuple vectors $(\boldsymbol{z}_1, \boldsymbol{z}_2)$ can be written as a function of correlation (I.M. and A.M., 1957):

$$I(\boldsymbol{z}_1, \boldsymbol{z}_2) = -\frac{1}{2}\log(1 - \rho^2) \quad (4)$$

where ρ corresponds to the correlation coefficient computed from from $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. Hence, we compute the mutual information for each slice r and sample x_i as $MI_i^r = I(\log(\tilde{\boldsymbol{w}}_i^r), \log({}^+\tilde{\boldsymbol{w}}_i^r))$. The $log(.)$ function accommodates the Log-Normal to Normal random variable transformation. For details on the implementation, see Alg. 1.

4) Mutual Information Aggregation: To compute the loss component for attention regularization, we need to aggregate the distributional similarities for the entire tensor. Aggregation is obtained by averaging the individual similarities obtained for each slice $r \in R$ and each sample x_i in the batch. With $\lambda \in \mathbb{R}$ some weighting scalar, the attention alignment loss term is:

$$\mathcal{L}_D(W_1, W_2) = -\frac{\lambda}{|R| \cdot |\mathcal{D}_b|} \sum_{i}^{\mathcal{D}_b} \sum_{r}^{R} MI_i^r \quad (5)$$

3 Experiments

In this section, we describe the experimental setting used for the evaluation, present our main results, and discuss different aspects of our method by providing several empirical analyses.

3.1 Experimental Setup

Model and Hyperparameters: Training is started from a pre-trained transformer LM. Specifically, we employ the Hugging Face (Wolf et al., 2020) implementation of $\text{BERT}_{\text{base}}$. For each approach evaluated, we follow the same hyperparameters proposed by the authors. In the InfoNCE loss, we set $\tau = 0.05$. In order to determine the hyperparameter λ a coarse grid search $\{1.0, 0.1, ..., 1.0e{-}5\}$ was conducted to assess the magnitude. Upon determination, a fine grid search was conducted once with 10 steps. We set $\lambda = 2.5e - 3$ for training 100% of the data in a single episode with a batch size of 50 at a learning rate of $3.0e{-}5$ and 250 warm-up steps. The number of optimization steps is kept constant for training the different dataset sizes. For the training set of size $10^6 (= 100\%)$, we train for 1 epoch; for the size of $10^5 (= 10\%)$,

Semantic Textual Similarity (STS) Benchmark								
Model	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg.**
BERT	21.54	32.11	21.28	37.89	44.24	20.29	42.42	31.40
BERT$^{\diamondsuit}$(first-last avg)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
GloVe$^{\clubsuit}$(avg.)	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
BERT-flow$^{\diamondsuit}$	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT-whitening$^{\diamondsuit}$	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS (Zhang et al., 2020)	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
SG-OPT (Kim et al., 2021)	66.84	80.13	71.23	81.56	77.17	77.23	68.16	74.62
CT (Carlsson et al., 2021)	67.43	79.18	69.05	76.92	74.62	73.24	68.38	72.69
SCD† (Klein and Nabi, 2022)	66.94	78.03	69.89	78.73	76.23	76.30	73.18	74.19
Mirror-BERT† (Liu et al., 2021)	69.10	81.10	73.00	81.90	75.70	78.00	69.10	75.40
SimCSE (Gao et al., 2021)	68.69	82.05	72.91	81.15	79.39	77.93	70.93	76.15
MoCoSE†(Cao et al., 2022b)	71.58	81.40	74.47	83.45	78.99	78.68	72.44	77.27
InforMin-CL† (Chen et al., 2022)	70.22	83.48	75.51	81.72	79.88	79.27	71.03	77.30
MixCSE† (Zhang et al., 2022b)	71.71	83.14	75.49	83.64	79.00	78.48	72.19	77.66
ConSERT$^{\dagger,*}_{large}$ (Yan et al., 2021b)	70.69	82.96	74.13	82.78	76.66	77.53	70.37	76.45
VaSCL†,* (Wang et al., 2022)	69.08	81.95	74.64	82.64	**80.57**	80.23	71.23	77.19
DCLR†,* (Zhou et al., 2022a)	70.81	83.73	75.11	82.56	78.44	78.31	71.59	77.22
ArcCSE†,* (Zhang et al., 2022c)	72.08	84.27	76.25	82.32	79.54	79.92	72.39	78.11
PCL†,* (Wu et al., 2022a)	72.74	83.36	76.05	83.07	79.26	79.72	72.75	78.14
ESimCSE†,* (Wu et al., 2021)	**73.40**	83.27	**77.25**	82.66	78.81	80.17	72.30	78.27
DiffCSE†,* (Chuang et al., 2022)	72.28	**84.43**	76.47	**83.90**	80.54	**80.59**	71.29	**78.49**
*mi*CSE	71.71	83.09	75.46	83.13	80.22	79.70	**73.62**	78.13

TABLE 1. Sentence embedding performance on STS tasks is measured as Spearman's correlation using $\text{BERT}_{\text{base}}$, except for VaSCL, which uses RoBERTa. Unless states otherwise, [CLS]-embedding was used. ♣: results from (Reimers and Gurevych, 2019); ♢ results from (Gao et al., 2021); † by the respective authors; other results are by ourselves, denotes the proposed approach, **bold** denotes the best result, and $*$ denotes the use of *discrete augmentation*.

we train for 10 epochs, etc. The training was conducted using an NVIDIA V100 with a training time of around $1.5h$. The overall GPU budget from experimentation and hyperparameter optimization is estimated to be around 500 GPU/hours. The momentum encoder is associated with a sample queue of size $|\mathcal{Q}| = 384$. The momentum encoder parameters are updated with a factor of 0.995, except for the MLP pooling layer, which is kept identical to the online network. Additionally, we increase the drop-out for the momentum encoder network from the default rate (0.1) to 0.3.

Data and Evaluation: Following (Gao et al., 2021), we train the model unsupervised on sentences from Wikipedia. We create random sample sets of different sizes $\{10^6, 10^5, 10^4, 5.0{\cdot}10^3, 10^3\}$ to train the model in a few-shot learning scenario. We repeated the training set creation for each size 5 times with different random seeds.

Mutual Information Estimation: Following the observations in (Voita et al., 2019), we restrict the computation of the mutual information to the upper part of the layer stack. Specifically, we select the layers between 8 and 12 (= last layer in $\text{BERT}_{\text{base}}$). To accommodate input sequences of varying lengths and make computation more efficient, we pool together pairs of adjacent heads (without overlap) while preserving the layer separation. From each of the $(4 \times \frac{H}{2})$ chunks of pooled attentions, we random sample 150 joint-attention pairs for each embedding of the bi-encoder.

3.2 Experimental Results

Unsupervised Sentence Embedding: We compare *mi*CSE to previous state-of-the-art sentence embedding methods on STS tasks. For comparisons, we favored comparable architectures (bi-encoder) that facilitate seamless integration of the proposed approach and methods of comparable backbone. We also added methods that employ explicit *discrete augmentation* to provide a full picture of existing techniques for sentence embedding.

For semantic text similarity, we evaluated on 7 STS tasks: (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). These datasets come in sentence pairs with correlation labels in the range of 0 and 5, indicating the se-

Algorithm 1 Mutual Information estimation

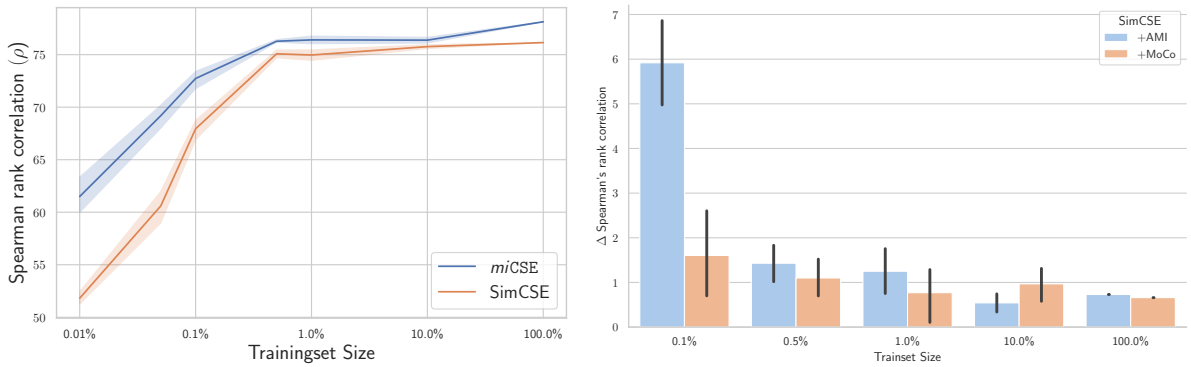
Input: Batch $\mathcal{D}_b$, encoder f_θ, multinomial sampler p_{mult}
Output: Average mutual information $\frac{1}{|R|\cdot|\mathcal{D}_b|}\sum_{i,r}^{\mathcal{D}_b,R} MI_i^r$
$(E_1, W_1), (E_2, W_2) \leftarrow f_\theta(\mathcal{D}_b)$ ▷ Transformer encoding creating views
for $i \leftarrow 1...|\mathcal{D}_b|$ **do**
  $\boldsymbol{w}_i, {}^{+}\boldsymbol{w}_i \leftarrow \text{EXTRACT}(W_1, W_2, i)$ ▷ Extract attention tensor for each sample
  $\{{}^{(+)}\boldsymbol{w}_i^1, ..., {}^{(+)}\boldsymbol{w}_i^R\} \leftarrow \pi({}^{(+)}\boldsymbol{w}_i)$ ▷ Slicing the attention tensors
  $s \leftarrow$ number of text tokens in x_i
  for $r \leftarrow 1...|R|$ **do**
    $J_r \leftarrow p_{mult}(1/s^2, ...1/s^2, 0)$ ▷ Sampling indices of valid attentions
    $MI_i^r \leftarrow \text{AMI}(\bigcup_{j\in J_r} \boldsymbol{w}_i^r[j], \bigcup_{j\in J_r} {}^{+}\boldsymbol{w}_i^r[j])$
  end for
end for
procedure $\text{AMI}(\boldsymbol{w}_1, \boldsymbol{w}_2)$
  $\boldsymbol{z}_1, \boldsymbol{z}_2 \leftarrow \log(\boldsymbol{w}_1), \log(\boldsymbol{w}_2)$ ▷ Log-Normal to Normal transform
  $\rho \leftarrow \cos(\boldsymbol{z}_1 - \bar{\boldsymbol{z}}_1, \boldsymbol{z}_2 - \bar{\boldsymbol{z}}_2)$ ▷ Compute correlation coefficient on centered attentions
  Return $-\frac{1}{2}(1-\rho^2)$ ▷ Mutual information for tensor slice
end procedure

mantic relatedness of the pairs. Specifically, we employ the SentEval toolkit (Conneau and Kiela, 2018) for evaluation. All our STS experiments are conducted in a *fully unsupervised* setup, not involving any STS training data. The benchmark measures the relatedness of two sentences based on the cosine similarity of their embeddings. The evaluation criterion is Spearman's rank correlation (ρ). For comparability, we follow the evaluation protocol of (Gao et al., 2021), employing Spearman's rank correlation and aggregation on all the topic subsets. Results for the sentence similarity experiment are presented in Tab. 1. As can be seen, the proposed approach is slightly lower in terms of average performance than state-of-the-art algorithms such as DiffCSE. However, it should be noted that these aforementioned methods use extensive discrete augmentation techniques, such as word repetition, deletion, and others, while the proposed method in this work does not employ any form of discrete data augmentation. This renders the proposed method more general and less ad-hoc in nature. While it is technically feasible for our method to incorporate discrete augmentation, it was deliberately excluded in this study for the sake of generalization with the intention of further exploration in future research. A more in-depth analysis shows the best performance on the SICK-R benchmark, where it outperforms the second-best approach SCD by $(+0.44)$ and third-best PCL by $(+0.87)$. We highlight the comparison to the closest method SimCSE, where the proposed approach has an average gain of $(+3.94)$. This improvement is due to the two additional components (i.e., AMI and MoCo) we add to this baseline method.

Low-shot Sentence Embedding: In this experiment, the performance of several SOTA sentence embedding approaches is benchmarked elaboratively. Similar to Sec. 3.2, we evaluate 7 STS tasks, STS Benchmark, and SICK-Relatedness with Spearman's ρ rank correlation as the evaluation metric. However, in contrast to the previous section, models are trained on different subsets of the data, namely $\{100\%, 10\%, 1\%, 0.1\%\}$ of the Wikipedia dataset used in (Gao et al., 2021). Results for the low-shot sentence similarity experiment can be presented in Fig. 3. As can be seen, the proposed approach gains by increasing the training set size and consistently outperforms all the baselines in all training subsets. Interestingly, our proposed method reaches the performance of SimCSE trained on the entire dataset with only 0.5% of the data. We believe it shows the impact of exploiting structural information for data augmentation during training. It should be noted that the performance gain is most significant when conducted on a single token rather than token averaging. We attribute this to token averaging, which to a certain degree,

Semantic Textual Similarity				
Model	**0.1%**	**1%**	**10%**	**100%**
CT (Carlsson et al., 2021)	68.46 ± 2.33	66.21 ± 4.06	72.06 ± 1.46	72.69
AMI+CT	71.12 ± 1.11	72.20 ± 0.49	73.20 ± 0.78	73.55
Mirror-BERT (Liu et al., 2021)	40.13 ± 5.08	42.17 ± 1.69	42.47 ± 3.66	43.32
AMI+Mirror-BERT	43.99 ± 1.26	45.26 ± 2.60	44.72 ± 1.36	47.48
Mirror (avg.) (Liu et al., 2021)	71.48 ± 1.19	71.80 ± 1.18	70.38 ± 1.18	69.81
AMI+Mirror-BERT (avg.)	71.49 ± 0.95	72.54 ± 0.49	70.68 ± 1.19	71.34
SimCSE (Gao et al., 2021)	67.94 ± 1.16	74.96 ± 0.65	75.76 ± 0.24	76.15
AMI+SimCSE	$\mathbf{73.85 \pm 0.49}$	76.21 ± 0.28	76.31 ± 0.46	76.88
*mi*CSE	73.68 ± 0.89	$\mathbf{76.40 \pm 0.48}$	$\mathbf{76.38 \pm 0.35}$	**78.13**

TABLE 2. Sentence embedding few-shot learning performance on STS tasks measured as Spearman's correlation using $\text{BERT}_{\text{base}}$. Unless states otherwise, [CLS]-embedding was used, the number corresponds to the average performance, **bold** denotes best performance, (●) denotes the integration of the proposed approach.



(A) Few-shot Performance

(B) Component Analysis

FIGURE 4. Few-shot performance analysis of models trained with different ratios of dataset size. Performance is shown in Spearman's correlation average of the STS benchmark. **Left**: Few-shot performance of SimCSE (Gao et al., 2021) (—) and the proposed approach *mi*CSE (—). **Right**: Few-shot ablation study with y-axis showing change (Δ) in Spearman's rank correlation ρ, showing the effect of adding components w.r.t. the SimCSE baseline.

is equivalent to attention regularization. On the *extremely* low data regime, the proposed approach shows very strong performance up (+11) compared to SimCSE - see Fig. 4a. It suggests resilience of our method to very small batch training.

3.3 Experimental Analysis of components

Given that AMI is a regularizer on Transformer attention, we evaluate the applicability in conjunction with other contrastive learning methods. We evaluate the following approaches CT (Carlsson et al., 2021), Mirror-BERT (Liu et al., 2021), and SimCSE (Gao et al., 2021). Evaluation is conducted on 7 STS tasks, STS Benchmark, and SICK-Relatedness with Spearman's ρ rank correlation as a metric. Results for the low-shot sentence similarity experiment are presented in Tab. 2. As can be seen, our proposed AMI can boost the performances of all approaches in all settings. Additionally, it shows the most significant boost in performance in combination with SimCSE. In addition, we observe that the impact of AMI grows with declining training set size. Combined with SimCSE, AMI leads to a performance gain of up to $(+5.91)$ at 0.1% of the data. We also observe that adding AMI to all the approaches significantly reduces the variance for all methods. This can probably be attributed to the regularization effect of the proposed AMI component. In addition, we conducted an ablation study to assess the effect of AMI and MoCo w.r.t. the baseline SimCSE - see Tab. 3. As shown in Fig. 4b, AMI and MoCo improve the baseline at different data ratios. Again, AMI provides a particularly strong performance boost in the low-data regime. In contrast, the impact of MoCo diminishes with decreasing training set size.

Semantic Textual Similarity				
Model	**0.1%**	**1%**	**10%**	**100%**
SimCSE (Gao et al., 2021)	67.94 ± 1.16	74.96 ± 0.65	75.76 ± 0.24	76.15
AMI+SimCSE	$\mathbf{73.85 \pm 0.49}$	76.21 ± 0.28	76.31 ± 0.46	76.88
MoCo+SimCSE	69.54 ± 1.61	75.73 ± 0.91	$\mathbf{76.73 \pm 0.29}$	76.81
*mi*CSE	73.68 ± 0.89	$\mathbf{76.40 \pm 0.48}$	76.38 ± 0.35	**78.13**

TABLE 3. Few-shot ablation study using [CLS]-embedding on STS tasks measured as Spearman's correlation using $\text{BERT}_{\text{base}}$. Performance corresponds to the average across all STS benchmarks, **bold** denotes best performance.

We emphasize that our approach gets the best of both worlds by integrating these two components. This can be directly exploited for different few-shot setups by adjusting the hyper-parameter λ.

Discussion on the *Structure* and *Attention*: The proposed approach aligns the attention patterns for drop-out augmented input pairs. We posit that conducting such a regularization enforces constraints w.r.t. the structure (e.g., syntax) of the sentence embeddings. This is motivated by recent literature findings, which suggest that the Transformer's attention captures structural information such as syntactic grammatical relationships of the sentences (Ravishankar et al., 2021; Clark et al., 2019; Raganato et al., 2018; Voita et al., 2019). Additionally, recent research explicitly targets the extraction of topologies from attention maps for diverse tasks on syntactic and grammatical structure (Kushnareva et al., 2021; Cherniavskii et al., 2022; Perez and Reinauer, 2022). Although no "one-to-one" mapping connects syntactic structures and attention patterns, the attention tensor, at the bare minimum, encodes a "holistic notion" of the syntactic structure of sentences. While this study refrains from making any definitive claim on the matter, a preliminary analysis wrt. role of syntax in our proposed method is conducted (see Appendix).

Discussion on the *discrete* argumentation: Discrete augmentation serves as a suitable strategy for expanding datasets to enhance learning robustness and partially address the issue of data scarcity. Although augmentation contributes to improved robustness, additional measures are required to tackle the information gap challenge in few-shot learning scenarios. Therefore, our current study deliberately excluded discrete augmentation to minimize any interference it may have with our low-shot learning algorithm. The primary rationale behind this decision is that while discrete augmentation is known to alleviate data scarcity by replicating missing information, it often leads to a superficial correlation between test and training data, rather than enhancing the model's few-shot learning capability. Consequently, we excluded augmentation to maintain control over *mi*CSE's behavior and validate its effectiveness without any negative consequences. The significant superiority of miCSE over augmentation-based approaches (such as DiffCSE) in the low-shot setup is evident from Fig. 3. Nevertheless, the proposed approach inherently facilitates the integration of discrete augmentation, offering the potential to enhance results in both few and full-shot learning scenarios. However, it is crucial to acknowledge that their structural similarities must be respected when applying augmentation strategies to positive pairs. One promising option is to utilize the augmentation strategies proposed by ESimCSE (Wu et al., 2021), which involve word *duplication* and *deletion* to address length biases. This can be followed by enforcing AMI on the shared attention subspaces of the augmented instances. Although we do not explore this approach in our current paper, it presents an intriguing avenue for future research.

4 Conclusion

We proposed a method to inject structural similarity into language models for self-supervised representation learning for sentence embeddings. The proposed approach integrates the inductive bias at the level of Transformer attention by enforcing mutual information on positive pairs obtained by drop-out augmentation. Leveraging attention regularization makes the proposed approach much more sample efficient. Consequently, it outperforms methods with a significant margin in low-shot learning scenarios while having state-of-the-art performance in full-shot to comparable approaches.

5 Limitations

The proposed AMI component is effective in the low-data regime but cannot be generalized to all cases. Future work will investigate the role of syntax in the structural regularization of attention and the extension of the proposed approach to discrete augmentation.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task (SemEval 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43.

Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32.

Rui Cao, Yihao Wang, Yuxin Liang, Ling Gao, Jie Zheng, Jie Ren, and Zheng Wang. 2022a. Exploring the impact of negative samples of contrastive learning: A case study of sentence embedding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3138–3152, Dublin, Ireland. Association for Computational Linguistics.

Rui Cao, Yihao Wang, Yuxin Liang, Ling Gao, Jie Zheng, Jie Ren, and Zheng Wang. 2022b. Exploring the impact of negative samples of contrastive learning: A case study of sentence embedding. In *Findings of the ACL*.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Shaobin Chen, Jie Zhou, Yuling Sun, and He Liang. 2022. An information minimization contrastive learning model for unsupervised sentence embeddings learning. In *COLING*.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. AdvAug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5961–5970, Online. Association for Computational Linguistics.

Daniil Cherniavskii, Eduard Tulchinskii, Vladislav Mikhailov, Irina Proskurina, Laida Kushnareva, Ekaterina Artemova, Serguei Barannikov, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2022. Acceptability judgements via examining the topology of attention maps.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations.

Claude Coulombe. 2018. Text data augmentation made simple by leveraging nlp cloud apis. *ArXiv*, abs/1812.04718.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. 2020. Bayesian attention modules. In *Advances in Neural Information Processing Systems*, volume 33, pages 16362–16376. Curran Associates, Inc.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *ArXiv*, abs/1905.08941.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735.

Gel'fand I.M. and Yaglom A.M. 1957. Calculation of amount of information about a random function contained in another such function. In *Amer. Math. Soc. Transl. Ser.: Series 2*, volume 12.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2528–2540, Online. Association for Computational Linguistics.

Tassilo Klein and Moin Nabi. 2022. Scd: Self-contrastive decorrelation for sentence embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL).

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. Artificial text detection via examining the topology of attention maps. In *EMNLP*, pages 635–649.

Fangyu Liu, Yunlong Jiao, Jordan Massiah, Emine Yilmaz, and Serhii Havrylov. 2022. Trans-encoder: Unsupervised sentence-pair modelling through self- and mutual-distillations. In *ICLR*.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1459, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. pages 216–223.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Ilan Perez and Raphael Reinauer. 2022. The topological bert: Transforming attention into topology for natural language processing.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Roberto Navigli. 2015. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228:95–128.

Alessandro Raganato, Jörg Tiedemann, et al. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. The Association for Computational Linguistics.

Vinit Ravishankar, Artur Kulmizev, Mostafa Abdou, Anders Søgaard, and Joakim Nivre. 2021. Attention can reflect syntactic structure (if you let it). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3031–3045, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Alessandro Sordoni, Nouha Dziri, Hannes Schulz, Geoff Gordon, Philip Bachman, and Remi Tachet Des Combes. 2021. Decomposed mutual information estimation for contrastive representation learning. In *International Conference on Machine Learning*, pages 9859–9869. PMLR.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Wei Wang, Liangzhu Ge, Jingqiao Zhang, and Cheng Yang. 2022. Improving contrastive learning of sentence embeddings with case-augmented positives and retrieved negatives. *arXiv preprint arXiv:2206.02457*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Ryosuke Kohita, Roger Levy, and Miguel Ballesteros. 2020. Structural supervision improves few-shot learning and syntactic generalization in neural language models. *arXiv preprint arXiv:2010.05725*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*.

Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. 2020. On mutual information in contrastive learning for visual representations. *arXiv preprint arXiv:2005.13149*.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022a. Pcl: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings.

Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022b. Infocse: Information-aggregated contrastive learning of sentence embeddings.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2021. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. *arXiv preprint arXiv:2109.04380*.

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021a. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021b. Consert: A contrastive framework for self-supervised sentence representation transfer. In *ACL*.

Chuanguang Yang, Zhulin An, Linhang Cai, and Yongjun Xu. 2021. Mutual contrastive learning for visual representation learning. *arXiv preprint arXiv:2104.12565*.

Dejiao Zhang, Wei Xiao, Henghui Zhu, Xiaofei Ma, and Andrew Arnold. 2022a. Virtual augmentation supported contrastive learning of sentence representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 864–876, Dublin, Ireland. Association for Computational Linguistics.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. pages 1601–1610.

Yanzhao Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. 2022b. Unsupervised sentence representation via contrastive learning with mixing negatives. *AAAI*.

Yuhao Zhang, Hongji Zhu, Yongliang Wang, Nan Xu, Xiaobo Li, and Binqiang Zhao. 2022c. A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space. In *ACL*.

Kun Zhou, Beichen Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2022a. Debiased contrastive learning of unsupervised sentence representations. *arXiv preprint arXiv:2205.00656*.

Kun Zhou, Beichen Zhang, Xin Zhao, and Ji-Rong Wen. 2022b. Debiased contrastive learning of unsupervised sentence representations. In *ACL*, pages 6120–6130.

A Appendix

In the following sections, we add additional details omitted in the main paper due to space restrictions. First, we show an analysis of the relationship between syntactic structure and semantics. Next, we illustrate the cosine similarity distribution according to human judgment (ground truth) in Sec. C. Next, in Sec. D, we visualize the 2D histogram of joint distributions between views. In Sec. E, we present detailed results of the few-shot performance of *mi*CSE in contrastive and non-contrastive setup. Finally, the exact relation between mutual information and correlation is presented in Sec. F.

B Analysis on Structure vs. Semantic

In light of the lack of a rigorous benchmark for analyzing structure(syntax) in sentence embedding, we performed two qualitative analyses visualized in Fig. 5 and Fig 6.

Let us consider the following three sentences and their linearized syntax tree to understand better the notions of negatives and (dis-)similar syntax.

Anchor / Positive:
Life is good
Negative (`similar Syntax`):
Good is expensive
Negative (`dissimilar Syntax`):
Live a good life

For each sentence, we computed the dependency tree. Subsequently, we linearize the tree structure for comparison, as can be done with tools such as spaCy[2]. Positive samples have an identical tree and negative samples have non-identical trees with their part-of-speech tags:

Anchor / Positive:
`nsubj(1,0) - ROOT(1,1) - acomp(1,2) - punct(1,3).`
Negative (`similar Syntax`):
`nsubj(1,0) - ROOT(1,1) - acomp(1,2) - punct(1,3).`
Negative (`dissimilar Syntax`):
`ROOT(0,0) - det(3,1) - amod(3,2) - npadvmod(0,3) - punct(0,4).`

Here `nsubj` corresponds to "nominal subject," `acomp` to "adjectival complement," `det` to "determiner," `npadv` to "noun phrase as adverbial modifier" and `punct` to "punctuation."

Our empirical observations are:

Observation (i) *There is a higher semantic and syntactic similarity between positive pairs compared to the negative pairs*: Our contrastive learning approach assumes that positive pairs exhibit more syntactic similarity than negative pairs (i.e., syntactic inductive bias). To validate this hypothesis, we plot the semantic similarity against syntactic similarity for both positive and negative pairs. Specifically, we analyzed the embeddings and attention values of the trained model with SimCSE and the proposed approach. Input to the models was randomly sampled sentences from Wikipedia. Interestingly enough, although training the proposed model involves maximization of MI over the attention w.r.t. positive pairs, we also observe the reflection of syntactic information in the negative pairs. As shown in Fig. 5, the negative pairs end up in the low left corner, whereas the positive pairs are in the upper right corner.

Observation (ii): *Negative pairs with similar syntax show higher attention similarity, compared to pairs with dissimilar syntax*: For a more in-depth analysis of this, we further sub-divided the negative pairs into two groups: *a)* negative pairs with similar dependency trees, *b)* negative pairs with dissimilar dependency trees. For simplicity, we adopted a binary similarity scheme - "similar" implies an identical dependency tree, whereas "dissimilar" corresponds to a non-identical dependency tree. To highlight the inter-group syntax similarity, samples of each group were normalized w.r.t. the centroid of the opposite group. As shown in Fig 6 (by the increased distance between the cluster centers), the proposed approach encodes a notion of syntactic similarity. Note that this margin appeared solely due to enforcing the AMI on attention for the positive pairs, leading to a notion of "syntax" on negative pairs.

[2]https://spacy.io/

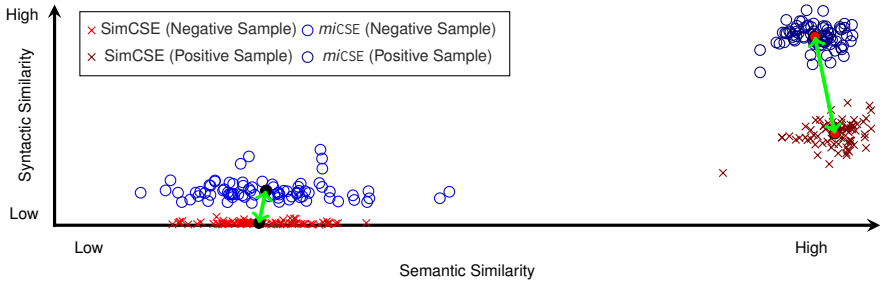


FIGURE 5. Sentence embeddings of positive and negative contrastive pairs in terms of semantic and syntax, comparing SimCSE and *mi*CSE. Semantic similarity is measured in terms of cosine similarity, syntactic similarity measured with mutual information on attention-level. (●) and (●) denote centroids of positive and negative centroids, (↔) their distance.

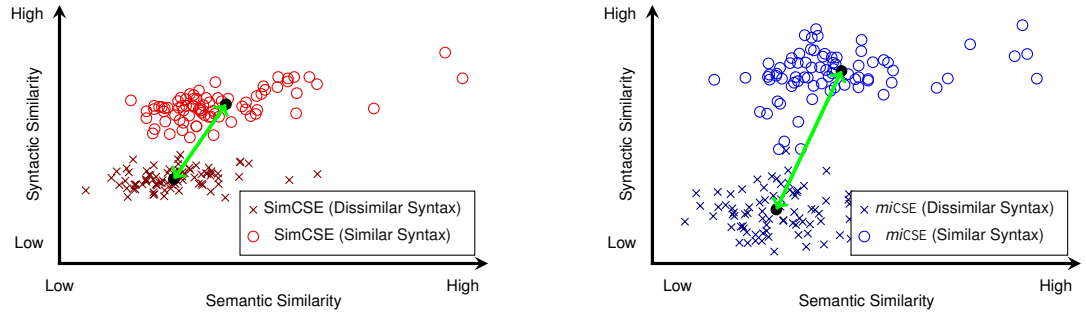


FIGURE 6. Comparison of negative contrastive pairs sentence embeddings in terms of semantic and syntax. **Left**: SimCSE **Right**: *mi*CSE. Semantic similarity measured in terms of cosine similarity, syntactic similary measured with mutual information on attentions. Negative pairs sub-divided into pairs with similar/dissimilar dependency trees. (●) denote cluster centroids, (↔) distance between centroids. Ranges are aligned.

C Cosine-similarity Distribution

To directly show the strengths of our approaches on STS tasks, we illustrate the cosine similarity on embeddings distributions of STS-B pairs in combination with human ratings in Fig. 7. The STS dataset comes in sentence pairs with correlation labels in the range of 0 and 5, indicating the semantic relatedness of the pairs. Here, the x-axis is the sample similarity of sentences according to human judgment (ground truth), and the y-axis represents the cosine similarity between pairs using embeddings. Color coding corresponds to ground-truth similarity. Compared to the baseline model (SimCSE), *mi*CSE better distinguishes sentence pairs with different levels of similarities, as can be seen from the stronger correlation between embedding distance and human rating. This property leads to better performance on STS tasks. In addition, we observe that *mi*CSE generally shows a more scattered distribution while preserving a lower variance on semantically similar sentence pairs. This observation further validates that *mi*CSE can potentially achieve a better alignment-uniformity balance.

D Visualization of Joint Distribution

To analyze the impact of the proposed approach compared to the baseline SimCSE at the attention level, we visualized the joint distribution of the attention values created by the two views created by the bi-encoder. The joint distribution and mutual information are closely related. More specifically, given two random variables X and Y, the associated mutual information can be expressed in terms of the joint distribution as:

$$I(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \qquad (6)$$

where $p(x,y)$ denotes the joint-distribution and $p(x), p(y)$ the marginals. Assuming random variables are normally distributed, the joint distribution of random variables is distinctly shaped depending on the correlation coefficient ρ. See Sec. F details on the relationship between entropy and the correlation coefficient. In the extreme case of totally unrelated marginals $\rho = 0$, the joint distribution assumes a circular shape having the lowest possible mutual information. On the other end of the

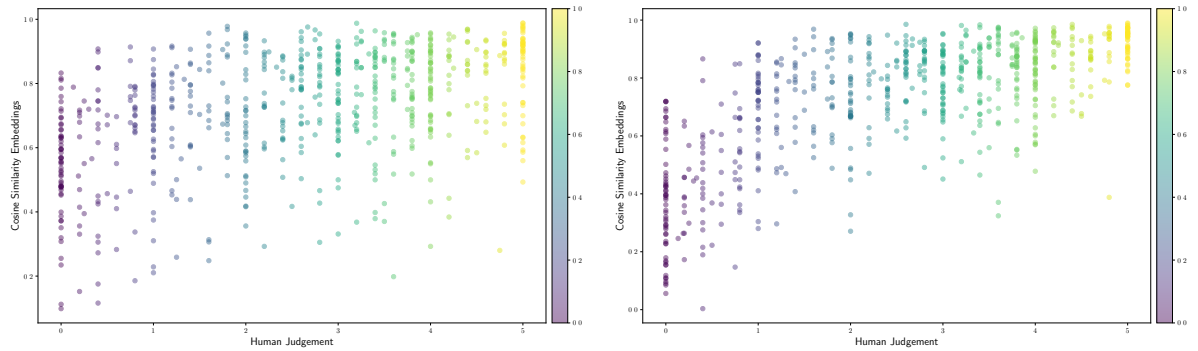


FIGURE 7. Scatter plots of cosine similarities between sentence pairs in STS. Pairs are shown based on ground-truth human scores (higher means more similar) along the x-axis; the y-axis is the cosine similarity. Color coding corresponds to ground-truth similarity. **Left:** SimCSE, **Right:** *mi*CSE (best viewed in color)

spectrum, in the case of perfect correlation, the joint distribution assumes collinearity (45° diagonal), with mutual information assuming maximal value. We sliced the attention tensor into 12 slices to avoid visual clutter, pooling together every 3 adjacent heads and every 4 adjacent layers. Slicing the tensor at a higher resolution leads to visually very similar results. The axes of the joint distribution (2d histogram) correspond to the marginals' distribution. As *mi*CSE maximizes the mutual information, one can observe a reduction in the scatter of the joint distribution compared to SimCSE.

E Detailed Comparison with SimCSE

Our proposed method is built on top of contrastive learning. Thus it intrinsically relies on the existence of the negative pairs. To complement the performance comparison of contrastive learning in Fig. 4a, we designed an experiment to analyze the extent to which attention regularization alone (AMI) can compensate for the lack of negative pairs. To that end, we conducted training with positive pairs only. See Tab. 4 and Fig. 9 for results. The integration of mutual attention information boosts the performance by up to $(+15)$ across all training set sizes. It suggests the potential application of our proposed attention regularization for non-contrastive learning.

F Bivariate Normal Mutual Information

General Log-Normal Properties: Similar to the normal distribution, the log-normal distribution $\log \mathcal{N}(w|\mu_w, \sigma_w^2)$ has two parameters μ_w and σ_w capturing mean and variance. It follows that applying the $\log$ transformation on a random variable w, we yield random variable $z = \log(w)$, which is normally distributed: $z \sim \mathcal{N}(\mu_z, \sigma_z^2)$.

Mutual Information: Given a vectors of tuples (X_1, X_2) containing i.i.d. points sampled the joint bivariate normal distribution of $p(A, B) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu} \in \mathbb{R}^2, \Sigma \in \mathbb{R}^{2\times 2}$. It can be shown that there exists an exact relationship between mutual information and the correlation coefficient ρ (I.M. and A.M., 1957) derived from X_1 and X_2. To that end, we expand the notation:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 & \mu_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{7}$$

The marginal and the joint entropy terms for Gaussian distributed variables can be written as:

$$H(X_i) = \frac{1}{2}\log(2\pi \boldsymbol{e}\sigma_i^2) = \frac{1}{2} + \frac{1}{2}\log(2\pi) + \log(\sigma_i), \quad i \in \{1, 2\} \tag{8}$$

$$H(X_1, X_2) = \frac{1}{2}\log\left[(2\pi \boldsymbol{e})^2|\Sigma|\right] = 1 + \log(2\pi) + \log(\sigma_1\sigma_2) + \frac{1}{2}(1-\rho^2). \tag{9}$$

Given that Mutual Information can be written in terms of entropy as:

$$I(X_1, X_2) = H(X_1) + H(X_2) - H(X_1, X_2) \tag{10}$$

Then it follows by inserting Eq. 8,9 in Eq. 10:

$$I(X_1, X_2) = -\frac{1}{2}(1-\rho^2) \tag{11}$$

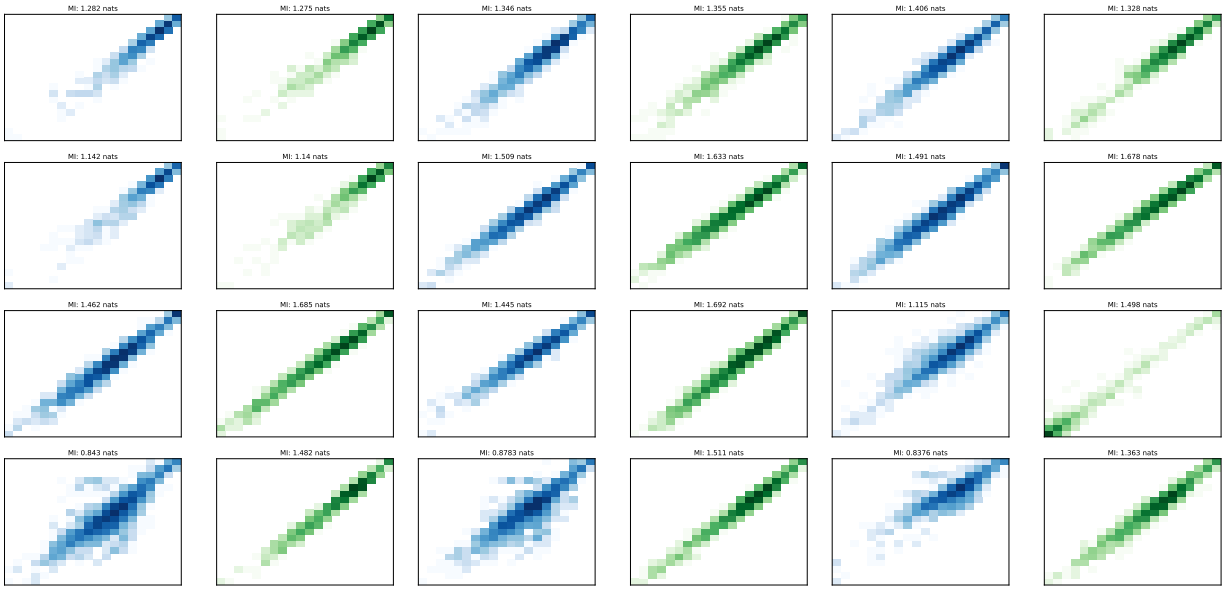


FIGURE 8. Joint distribution between two augmentation induced views. Images depict 12 attention slices per methods, obtained by slicing the attention tensor for the input sentence "the best thing you can do is to know your stuff." Increasing depth in layer stack from left to right, top to bottom. (●) SimCSE, (●): *mi*CSE (best viewed in color)

Semantic Textual Similarity				
Model	**0.1%**	**1%**	**10%**	**100%**
SimCSE (with negatives)	66.69 ± 1.03	74.08 ± 0.81	75.01 ± 0.23	76.15
$*$ *mi*CSE (with negatives)	73.85 ± 0.49	76.21 ± 0.28	76.31 ± 0.46	78.13
SimCSE (w/o negatives)	43.02 ± 4.48	41.30 ± 1.63	42.56 ± 6.87	40.18
$*$ *mi*CSE (w/o negatives)	57.00 ± 1.32	56.41 ± 3.38	53.38 ± 4.70	54.34

TABLE 4. Sentence embedding few-shot learning performance on STS tasks measured as Spearman's correlation. Top: performance in contrastive setup with in-batch negatives. Bottom: performance with positive samples only. The number corresponds to the average performance across all benchmarks.

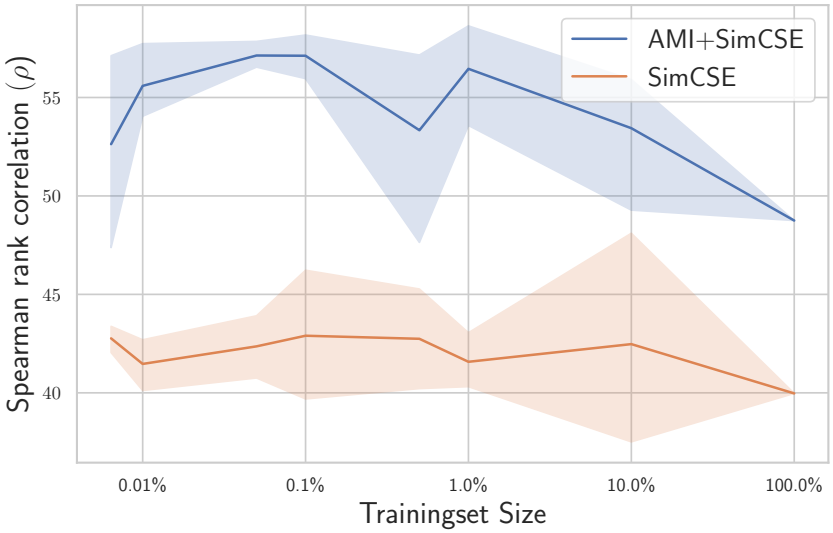


FIGURE 9. Few-shot performance of SimCSE (Gao et al., 2021) (—) and the proposed approach AMI in combination with SimCSE (—). Performance is shown in Spearman's correlation average of the STS benchmark at different ratios of dataset sizes used for training. Training in non-contrastive setting with positive-only pairs.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 5

☐ A2. Did you discuss any potential risks of your work?
Not applicable. No potential risk

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☑ A4. Have you used AI writing assistants when working on this paper?
Grammarly for grammar correction and spelling correction

B ☑ Did you use or create scientific artifacts?

Section 3.1, 3.2

☑ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. All open-source

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.2

C ☑ Did you run computational experiments?

Section 3.1 + Section 3.2

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 3.1 + Section 3.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.1 + Section 3.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3.1 + Section 3.2

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.