applied sciences

MDPI

Article

SelfCCL: Curriculum Contrastive Learning by Transferring Self-Taught Knowledge for Fine-Tuning BERT

Somaiyeh Dehghan * and Mehmet Fatih Amasyali

Department of Computer Engineering, Yildiz Technical University, Istanbul 34220, Turkey
* Correspondence: so.dehghan87@gmail.com

Abstract: BERT, the most popular deep learning language model, has yielded breakthrough results in various NLP tasks. However, the semantic representation space learned by BERT has the property of anisotropy. Therefore, BERT needs to be fine-tuned for certain downstream tasks such as Semantic Textual Similarity (STS). To overcome this problem and improve the sentence representation space, some contrastive learning methods have been proposed for fine-tuning BERT. However, existing contrastive learning models do not consider the importance of input triplets in terms of easy and hard negatives during training. In this paper, we propose the SelfCCL: Curriculum Contrastive Learning model by Transferring Self-taught Knowledge for Fine-Tuning BERT, which mimics the two ways that humans learn about the world around them, namely contrastive learning and curriculum learning. The former learns by contrasting similar and dissimilar samples. The latter is inspired by the way humans learn from the simplest concepts to the most complex concepts. Our model also performs this training by transferring self-taught knowledge. That is, the model figures out which triplets are easy or difficult based on previously learned knowledge, and then learns based on those triplets in the order of curriculum using a contrastive objective. We apply our proposed model to the BERT and Sentence BERT(SBERT) frameworks. The evaluation results of SelfCCL on the standard STS and SentEval transfer learning tasks show that using curriculum learning together with contrastive learning increases average performance to some extent.

Keywords: transfer learning; curriculum learning; contrastive learning; self-taught learning; sentence embedding; natural language processing; semantic textual similarity; BERT

check for updates

Citation: Dehghan, S.; Amasyali, M.F. SelfCCL: Curriculum Contrastive Learning by Transferring Self-Taught Knowledge for Fine-Tuning BERT. *Appl. Sci.* **2023**, *13*, 1913. https://doi.org/10.3390/app13031913

Academic Editor: Pengjie Ren

Received: 18 December 2022
Revised: 24 January 2023
Accepted: 28 January 2023
Published: 1 February 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The advent of BERT [1], a pre-trained conceptualized language model, was a paradigm shift in Natural Language Processing (NLP) particularly because of the introduction of the pre-training/fine-tuning mechanism [2,3]. That is, after pre-training in a self-supervised way on a tremendous amount of textual data, the BERT model can be rapidly fine-tuned on a certain downstream task with small labeled-data and little time, because the general linguistic patterns have already been learned during the pre-training phase [3].

One of the essential downstream tasks in NLP, for which BERT needs to be fine-tuned, is Semantic Textual Similarity (STS), which quantifies the closeness of semantic meaning of given texts that have the same meaning, but are not necessarily lexically similar. STS is indispensable for many NLP applications including information retrieval, text classification, machine translation, sentiment analysis, text summarization, market intelligence, named entity recognition, etc. For example, in information retrieval, STS is used to measure the relevancy between the retrieved document and the user query; in sentiment analysis of tweets, STS is used to estimate similarity scores between tweets; in question answering, STS is used to find correct answers that have no lexical overlap but are still semantically similar; in market intelligence, STS is used to select the best wording for questions to use when creating surveys, as the questions can be expressed in such a way that the information gathered remains valuable and relevant over time. For example, consider these

two questions: "How old are you?" and "What year were you born?". Both have the same meaning, but the latter will still be useful several years from now.

Although BERT has been very successful in some NLP tasks, sentence embeddings derived from BERT are susceptible to collapse without fine-tuning. That is, they are mapped in a small space and almost all pairs of sentences have a cosine similarity in the range of {0.6, 1} [4]. This problem arises from the fact that the frequency of words distorts the embedding space [4,5].

To overcome this problem, a thread of research has thus been working on applying contrastive learning techniques under the pre-train-then-fine-tune paradigm of BERT [4,6–14]. Contrastive learning pulls similar samples closer together, but pushes out dissimilar samples [15]. Basically, contrastive learning is inspired by the way humans learn about the world around them, using the principle of contrasting samples against each other. The idea behind it is similar to the "match the correct image" game for children, where they learn by comparing similar and dissimilar pictures. The issue here, however, is the difficulty level of the game. For example, as shown in Figure 1, we can match easy or hard pictures. The learning process begins with simple examples of a task and then gradually increases the difficulty of the task; this is called curriculum learning, and is based on the human learning strategy from easy to hard material.

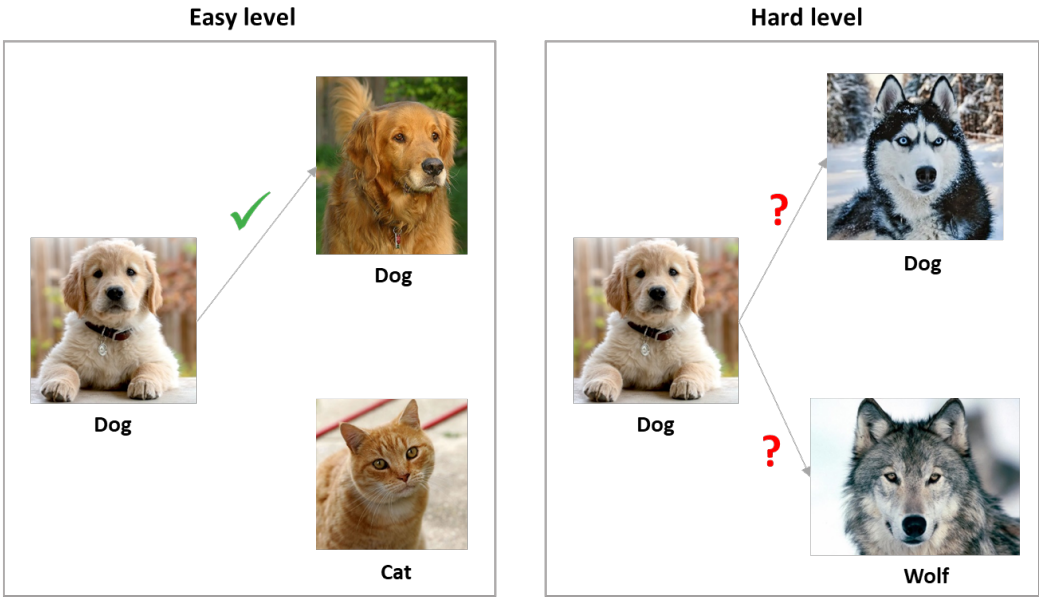


Figure 1. "Match the correct animal" by contrasting similar and dissimilar images.

Based on the above descriptions, this work aims to answer the following question: Can applying curriculum learning to contrastive learning increase the performance of contrastive learning? To answer this question, we propose Curriculum Contrastive Learning by Transferring Self-taught Knowledge for Fine-Tuning BERT. For simplicity, we abbreviate the foregoing as SelfCCL. Our model learns in the meaningful order from easy to hard triplets based on the contrastive objective in a self-taught manner. That is, SelfCCL transfers self-taught knowledge to sort input triplets by difficulty. In essence, self-taught learning is a subcategory of transfer learning, which is the ability of transferring the knowledge and skills learned in a previous task to a new task [16].

SelfCCL first uses unlabeled contrastive triplets (x_1, x_i^+, x_i^-) for triplet mining and scoring triplets to easy, semi-hard and hard through BERT itself. Then, BERT is fine-tuned in curriculum order using these self-taught labeled triplets based on contrastive objective. To sum up, the SelfCCL model figures out what matters for itself and then learns based on it.

For a detailed investigation, we perform three experiments. In the first experiment, we assess SelfCCL on seven standard STS tasks. In the second experiment, we assess SelfCCL

on seven standard transfer learning tasks. In the third experiment, we investigate the correlation between the cosine similarity of the sentence embeddings generated by SelfCCL and the human-annotated scores in the SICK dataset to observe the contrastive learning ability in distinguishing similar and dissimilar sentences.

Our principal contributions can be briefly notated as follows:

- We present a SelfCCL model for fine-tuning BERT based on the combination of curriculum learning and contrastive learning;
- Our model transfers self-taught knowledge to score and sorts input-data triplets;
- Our model surpasses the previous state-of-the-art models;
- The results reveal that the use of curriculum learning along with contrastive learning partially increases the average performance.

The remainder of the paper is organized as follows: In Section 2, a short summary of contextualized sentence embedding models is provided. In Section 3, a brief background on contrastive learning, curriculum learning, and self-taught transfer learning is given. In Section 4, our proposed model (SelfCCL) is presented. Thereafter, in Section 5, the experiments are given. Finally, conclusions and recommendations for future works are provided in Section 6.

2. Related Works

The traditional embedding models such as Word2Vec [17], GloVe [18], and FastText [19] only work on the word level. After that, the same idea was extended to learn sentence embeddings instead of learning word embeddings. Sentence embedding models can be divided into those that produce non-contextualized embeddings, and those that produce contextualized embeddings [20]. Early sentence embedding models such as Doc2Vec [21], Skip-thought [22], FastSent [23], Sent2Vec [24], and Quick-thought [25] are non-contextualized embeddings and have been relatively successful in some of NLP tasks. However, these embeddings provide an exact meaning to words, which is a major downside of them, as the meaning of words changes based on context. For example, in the following three sentences, the meaning of the word "date" can be changed based on context:

- Her favorite to eat is a date.
- They went on a date tonight.
- What is your date of birth?

Thus, learning high-quality sentence representations can still be challenging, because a desirable sentence embedding model is expected to be able to model complex features of word use (e.g., syntax and semantics) and to use linguistic contexts in a way that can handle polysemy [26].

Recently, deep contextualized Sentence-embedding models such as Facebook's InferSent [27], AllenAI's ELMo [26], USE [28], and Google's BERT [1] have been proposed, which are considered as important milestone achievements in word and sentence representation. These pre-trained machine learning models can encode a sentence into deep contextualized embeddings that depend on its intra-sentential context [20]. Among contextualized sentence embedding models, BERT has established itself as the most effective NLP model, performing excellently in many NLP applications with appropriate fine-tuning for particular downstream tasks [29].

BERT is a contextualized language model based on the Transformer architecture developed by researchers at Google AI. The Transformer-based architecture of BERT uses the amazing attention mechanism that learns contextual relationships between words in a sequence of text. In fact, BERT solves the long-term dependencies problem of Recurrent Neural Networks (RNNs) by attention mechanism, which is a technique for paying attention to specific words.

Recent works have shown that fine-tuning BERT has been very successful in many NLP tasks, including sentiment analysis [29], text classification [30], named-entity recognition (NER) [31], intent recognition [32], etc. For instance, Agrawal et al. in Ref. [31] attempted

to solve the nested-NER problem using transfer learning through fine-tuning BERT. They performed fine-tuning of various pre-trained BERT models on the datasets in which the nested labels are connected by flattening (e.g., Tottenham: organization + location). Similarly, in Ref. [32], Fernández-Martínez et al. fine-tuned BERT for intent recognition using vocabulary extension. They showed that adding adequate domain specific words to BERT's original tokenizer vocabulary can improve performance.

The biggest advantage of BERT is that it generates "contextualized" word embeddings, which means that it can assign each word a representation based on its context. The biggest downside of BERT is that it is slow to train as it has millions of parameters and needs a prohibitively large dataset in order to train to reasonable accuracy. In addition, BERT is based on a cross-encoder architecture, which makes it too slow for sentence-pair tasks such as clustering, since both sentences must be fed into the network, which requires an enormous amount of computation. For example, clustering 10,000 sentences requires approximately 65 h of training.

Ref. [33] have proposed a Bi-encoder BERT model, which is called Sentence-BERT (SBERT). SBERT aims to overcome this challenge through a siamese network, which creates sentence embeddings for each sentence and can then compare them using a cosine-similarity. SBERT makes semantic search feasible for a large number of sentences, reducing the time from 65 h to about 5 s for the same complexity of clustering 10,000 sentences.

SBERT uses NLI datasets such as SNLI [34] and Multi-Genre NLI (MultiNLI) datasets [35]. These datasets contain sentence pairs labeled as entailment, contradiction, or neutral. SBERT first enters each pair of sentences into BERT and obtains sentence embeddings u and v through a mean pooling operation. It then inputs a composite vector of these vectors in the form of $(u, v, |u - v|)$ into a three-way softmax classifier to predict the label of the given sentence pair.

3. Background

3.1. Contrastive Learning

Contrastive learning is self-supervised learning that encourages similar data points to stay close together in the embedding space, while dissimilar ones stay far apart. In other words, contrastive learning is an approach to create a model for finding similar and dissimilar things in machine learning.

Contrastive loss proposed by Refs. [36,37] is the first training objective that was used for contrastive learning. It takes a pair of embedding vectors (x_i, x_j) and a label, either 1 (if they belong to the same class) or 0 (if they belong to different classes). The objective function then tries to decrease the distance between two embedding vectors with label 1, and increase the distance between two embedding vectors with label 0. The contrastive loss formula is defined as:

$$L_{CL}(x_i, x_j, Y) = (1 - Y)\frac{1}{2}(D_{x_i,x_j})^2 \; + \; Y\frac{1}{2}\Big\{max(0, m - D_{x_i,x_j})\Big\}^2 \tag{1}$$

where (x_i, x_j) is a pair of embedding vectors, Y is a label that can be 0 or 1, m is a margin that defines the lower boundary distance between dissimilar samples, and $D(\cdot)$ is a Euclidean distance between a pair of embedding vectors of samples.

Triplet loss is an improvement of the contrastive loss first proposed in Ref. [38] for face recognition. It outperforms the former by using triplets of samples instead of pairs. Triplet loss consists of an anchor, a positive, and a negative in the form of (x_i, x_i^+, x_i^-). Anchor and positive belong to the same class while negative belongs to different classes. The loss is calculated over triplets of anchor-positive-negative, so that for each triplet the anchor-positive distance must be smaller than the anchor-negative distance. The formula for triplet loss is defined as follows:

$$L_{TL} = max\big(D\big(x, x^+\big) - D\big(x, x^-\big) + m,\ 0\big) \tag{2}$$

where $D(\cdot)$ is distance function, which can be Euclidean or cosine, and m is a parameter that specifies how far dissimilar samples should be from the anchor.

Triplet Mining

The performance of the triplet loss depends strongly on the selection of the triplets. On the other hand, selecting the triplet randomly leads to non-convergence [38,39]. To overcome this problem, Ref. [38] proposed triplet mining. As shown in Figure 2, depending on the distance between the anchor, positive, and negative points, there are three possible types of triplets: easy triplets, semi-hard triplets, and hard triplets.

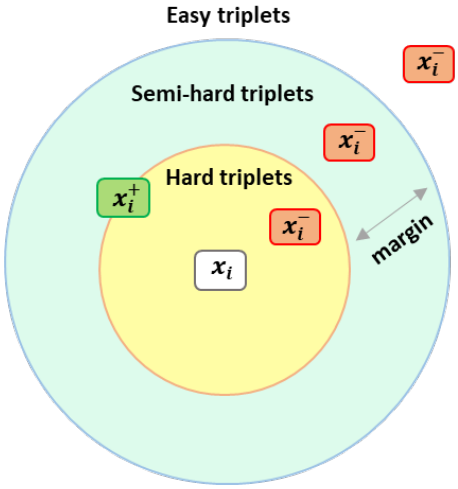


Figure 2. Three possible triplets depending on the distance between an anchor and a positive or a negative.

In the original FaceNet paper [38], the authors used semi-hard triplets in their work. However, offline triplet mining, e.g., before training starts, is extremely inefficient. To overcome this restriction, online triplet mining is used. The idea is to select triplets during training epochs within a batch of samples [38,40], so that for each anchor-positive pair within a batch, other in-batch samples are considered as negatives. Online triplet mining is also known as in-batch negative or batch-wise [41].

NT-Xent loss, which stands for Normalized Temperature-Scaled Cross-Entropy loss, is the most popular batch-wise contrastive loss, and is proposed in the SimCLR paper by Ref. [42]. It is an extended version of the multi-class N-pair loss [43]. This loss function takes positive pairs in the form of (x_i, x_i^+) and other possible pairs within the batch considered as negatives. The formula for NT-Xent loss is defined as follows:

$$L_{NT-Xent} = -\log \frac{e^{sim(x_i, x_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(x_i, x_j^+)/\tau}} \tag{3}$$

where $sim(\cdot)$ is the standard cosine similarity, and τ is a temperature parameter to scale the cosine similarity.

3.2. Curriculum Learning

Curriculum Learning (CL) is a strategy of training a machine learning model from simpler to more difficult patterns, inspired by the meaningful learning-order of human from easy concepts to complex concepts. The idea of curriculum learning was introduced earlier by Ref. [44] and first, formalized in machine learning by Bengio et al. in 2009 [45]. Curriculum learning strategy has been successfully employed in several areas of machine learning including computer vision, speech recognition, audio-visual representations learning, robotics, and NLP [46].

There are two common frameworks for curriculum learning, the data-level CL and the model-level CL. Data-level CL, the original formulation by Ref. [45], and also known as vanilla CL, deals with gradually increasing complexity of data during the training process while the model-level CL deals with gradually increasing the model capacity by adding more units, activating more units, or deblurring conventional filters during the progress of training process [46]. Most CL works in the literature use the data-level CL.

Generally speaking, the curriculum is defined by three elements: (1) the scoring function, (2) the pacing function, and (3) the order. Scoring function and the pacing function are also known as the difficulty measurer and the curriculum scheduler in the literature, respectively [47].

1. **Scoring function:** The scoring function determines the criterion for scoring data to easy and hard samples. For example, in natural language processing tasks, word frequency and sentence length are mostly used as a criterion for difficulty measurer [46]. So, expressed mathematically, the scoring function is a map from an input example, x, to a numerical score, $s(x) \in \mathbb{R}$, where a higher score corresponds to a more difficult example [48].
2. **Pacing function:** The pacing function $g(t)$ determines when harder samples are presented to the model during the training process. To put it simply, the pacing function determines the size of training data to be used at epoch t.
3. **The order:** The order corresponds to ascending (curriculum), descending (anti-curriculum), and random-curriculum. Anti-curriculum learning uses the scoring function, in which training examples are sorted in descending order of difficulty; thus, the more difficult examples are queried before the easier ones [49]. In the random-curriculum, the size of the batch is dynamically grown over time, while the examples within the batch are randomly ordered [48,50].

3.3. Transfer Learning and Self-Taught Learning

Training a machine learning model from scratch is computationally expensive since collecting sufficient training data is often costly and time-consuming. This is where transfer learning comes into play. Transfer learning is the ability of transferring the knowledge gained from a task to another task [16]. Put simply, transfer learning is the reuse of a pre-trained model for a model on a new task with often related source domains [51]. The use of transfer learning is very common in the fields of NLP and Computer Vision (CV). In NLP, BERT, for example, is used for as the starting point for other tasks such as sentiment analysis, text summarization, text classification, machine translation, and so on.

Transfer learning can be classified based on several factors. However, there is no definitive categorization for it in the literature. The most common classification is based on data, task, and domains of the source and target models [51,52]. Depending on the tasks, domains and type of data available, i.e., labeled/unlabeled, for source and target models, transfer learning methods fall into one of three categories: inductive, transductive, and unsupervised [51]. Table 1 shows these three categories.

Self-Taught learning is a type of inductive transfer learning, when a pre-trained model on an unlabeled source task is reused for a labeled target task. Self-taught learning was first introduced in Ref. [53] by Stanford researchers in 2007 for image classification. They used the prior knowledge learned from unlabeled data for a new supervised classification task.

Table 1. Taxonomy of transfer learning depending on data, task, and domain of source and target models.

Category	Source Data Labeled?	Target Data Labeled?	Source and Target Task	Source and Target Domains
Inductive	Can be labeled and unlabeled	Yes	Different but related	Same
Transductive	Yes	No	Same	Different but related
Unsupervised	No	No	Different but related	Different but related

4. SelfCCL: Curriculum Contrastive Learning by Transferring Self-Taught Knowledge for Fine-Tuning BERT

4.1. Methodology

We propose a SelfCCL model that learns in the meaningful order from the easiest to most difficult triplets, based on contrastive objective through the self-taught knowledge. Our model consists of two phases. We have depicted these two phases in Figure 3. In the first phase, Figure 3a, the input data triplets are scored trough self-taught transfer learning from BERT and sorted in the curriculum order. In fact, we use prior knowledge from BERT, learned during pre-trining, to score the data. No training or fine-tuning takes place at this stage. In the second phase, Figure 3b, the same BERT model is fine-tuned based on the curriculum order (ascending order) of these triplets using supervised contrastive objective.

In the following sections, we explain the training data, scoring function and pacing function of the curriculum based on self-taught transfer learning, as well as our contrastive training objective.

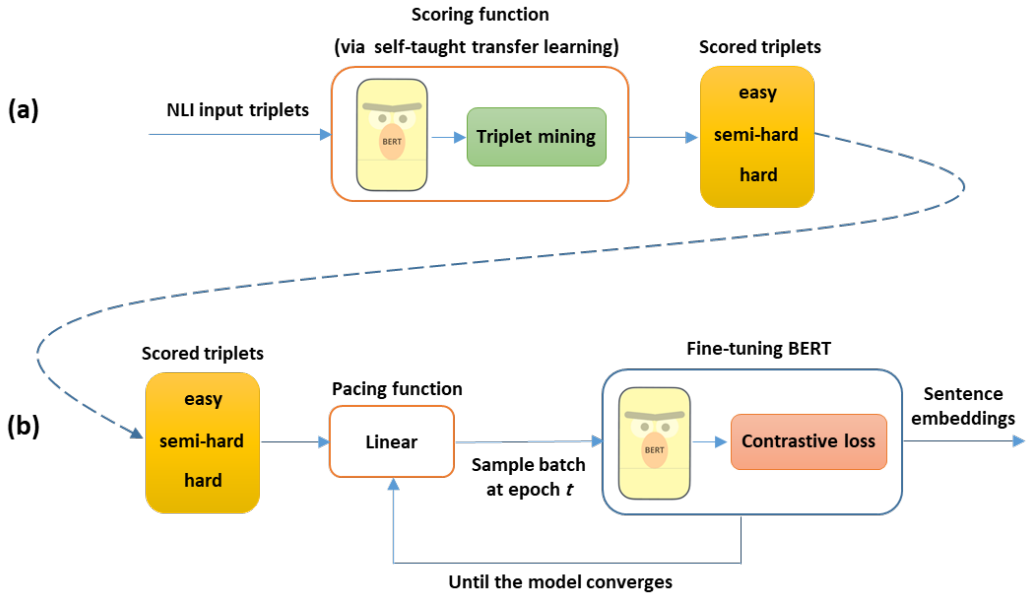


Figure 3. SelfCCL model; (**a**) Easy-to-hard triplets mining using BERT in a self-taught way; (**b**) Curriculum contrastive based fine-tuning of BERT.

4.2. Training Data

We use SNLI [34] and Multi-genre NLI (MNLI) datasets [35]. SNLI (570 K) and MNLI (433 K) contain sentence pairs (premise-hypothesis), labeled as entailment, neutral, or contradiction. We obtain only entailment and contradiction labels and use them in the form of (x_1, x_i^+, x_i^-), where entailment hypotheses are positives and contradiction hypotheses are negatives for premise (anchor). The total number of input triplets in our training dataset is about ∼314 K.

4.3. Curriculum Setting

For the Curriculum setting of our model, we use vanilla CL configuration by Bengio et al. [45]. The settings are as follows:

4.3.1. Scoring Function

The challenge facing curriculum learning is how to score training data in a way that reflects an easy-to-difficult arrangement. Indeed, this step requires additional labeling. To confront this challenge, we first establish the scoring criteria according to the triplet mining rules. We use the following criteria to classify triplets in the form of (x_i, x_i^+, x_i^-) into easy, semi-hard, and hard triplets:

$$\begin{aligned} &\textbf{Easy triplets:} && d(x, x^+) + m < d(x, x^-) \\ &\textbf{Semi-hard triplets:} && d(x, x^+) < d(x, x^-) < d(x, x^+) + m \\ &\textbf{Hard triplets:} && d(x, x^-) < d(x, x^+) \end{aligned}$$

where d is cosine distance and $m = 0.2$ is margin.

Next, we use BERT itself to apply these three triplet mining criteria to score input-triplets. That is, inspired by self-taught learning, we reuse the pre-trained BERT model for scoring input triplets. As shown in Figure 3a, NLI input-triplets in the form of (x_i, x_i^+, x_i^-) are fed into BERT and their embeddings are obtained. We use BERT-base (uncased) to generate sentence embeddings. Then, triplets are classified as easy, semi-hard, and hard triplets according to the mentioned criteria. This phase is performed offline. An example of this scoring phase can be found in Table 2.

Table 2. Triplet mining for triplet samples from SNLI [34].

Premise	Entailment	Contradiction	Score
A man with a beard,	A bearded man pulls a rope.	A man pulls his beard.	easy
wearing a red shirt	A bearded man is pulling on a rope.	The man was clean shaven.	easy
with gray sleeves and	A man is pulling on a rope.	A man in a swimsuit, swings on a rope.	semi-hard
work gloves,	The man is able to grow a beard.	A man is wearing a black shirt.	semi-hard
pulling on a rope.	A man pulls on a rope.	A bearded man is pulling a car with his teeth.	hard

4.3.2. Pacing Function

After we get the score of each triplet, we use the pacing function to schedule how to introduce the triplets during the training process. The pacing function $g(t)$ specifies the size of training input at each epoch t. We use the linear pacing function as follows:

$$g(t) = \left(\frac{t}{T}\right)^{\lambda} \times k \tag{4}$$

where T is the total number of training steps, k is the number of sorted training samples, λ is a smoothing parameter that controls the pace in the training procedure. $\lambda = \frac{1}{2}, 1, 2$ denote the root, linear, and quadratic pacing functions, respectively. Here, we use the linear setting as $\lambda = 1$.

4.4. Training Objective

For contrastive objective, we follow the contrastive framework in Ref. [54] and Ref. [12], which is a developing version of NT-Xent loss by adding a hard negative to it. This loss function is also a version of Multiple Negatives Ranking Loss (MNRL) (https://www.sbert.net/ (accessed on 13 December 2022)). So for a triplet in the form of (x_i, x_i^+, x_i^-) in a mini-batch, the loss is defined as:

$$L = -\log \frac{e^{sim(x_i, x_i^+)/\tau}}{\sum_{j=1}^{N} (e^{sim(x_i, x_j^+)/\tau} + e^{sim(x_i, x_j^-)/\tau})} \tag{5}$$

5. Experiments

5.1. Training Setups

For SelfCCL-BERT, we use $\text{SimCSE}_{\text{sup}}$ setup (https://github.com/princeton-nlp/SimCSE (accessed on 13 December 2022)) and for SelfCCL-SBERT, we use sentence-transformers. For both of our models, we start from pre-trained BERT-base (uncased) hosted on the Hugging Face Model Hub (https://huggingface.co/ (accessed on 13 December 2022)). We use the average embeddings of the first and last layers ("avg-first-last") and "mean" as pooling mode for SelfCCL-BERT and SelfCCL-SBERT, respectively. We run SelfCCL-BERT on 4 A100 GPUs and and SelfCCL-SBERT on 1 A100 GPU with CUDA 11, and train them for four epochs. We also tested more epochs, but overfitting was observed. In the original paper [1], the authors essentially recommend only two to four training epochs for fine-tuning BERT on specific NLP tasks, since general linguistic patterns were already learned during pre-training. We also use a batch size of 512 for SelfCCL-BERT and 350 for SelfCCL-SBERT.

5.2. Baseline and Previous Models

In our experiments, the SelfCCL model is compared to the preceding sentence encoder models. They are listed as supervised and unsupervised models in Table 3, separately. Refs. [4,6–15] use contrastive learning. BERT-Flow and BERT-whitening [5,55] are post-processing models that apply flow-network and whitening to enhance BERT, respectively. SBERT [33] and SBERT-base-nli-v2 (our reproduced model) are considered as baseline models.

The SimCSE [12] and SupMPN [15] models are the last two contrastive learning models that have performed the best compared to baseline SBERT and other previous models. SimCSE uses a contrastive loss function similar to our SelfCCL model so that it accepts triplets in the form of (x_i, x_i^+, x_i^-). SupMPN uses a SupMPNRL objective function, which accepts one anchor, multiple hard positives, and multiple hard negatives in the form of $(x_i, x_{i1}^+, \ldots, x_{i5}^+, x_{i1}^-, \ldots, x_{i5}^-)$. In these two methods, there is no curriculum order of input triplets.

5.3. Reproducing SBERT-Base-Nli-V2 Model

SBERT-base-nli-v2 checkpoint is the second version of "nli-bert-base" checkpoint from sentence-transformers, which is trained using a contrastive objective by Ref. [56]. They reported its evaluation results on STS tasks. However, they did not release its files. Additionally, Ref. [57] also reproduced SBERT-base-nli-v2, but they did not report any result for the STS tasks. Therefore, in order to make an accurate comparison, we reproduced it again. We used SNLI and MNLI entailment and contradiction hypotheses in the form of (x_i, x_i^+, x_i^-) and we used Multiple Negatives Ranking Loss (MNRL) as the objective function. We trained 3 epochs on 1 A100 GPU with a batch size of 350. Table 3 shows the specifications for our proposed models, SBERT-base-nli-v2 (our reproduced model), state-of-the-art (SOTA) SimCSE [12], and SOTA SupMPN [15].

Table 3. The specifications for SBERT-base-nli-v2, SimCSE, SupMPN, and our proposed models.

Model	GPU	Number of GPUs Used in Training	Number of Training Epochs	Batch Size	The Form of Input Triplets for Contrastive Objective	Using Curriculum Learning
$\text{SupMPN-BERT}_{\text{base}}$	Nvidia A100 80 GB	8	3	512	$(x_i, x_{i1}^+, \ldots, x_{i5}^+, x_{i1}^-, \ldots, x_{i5}^-)$	No
$\text{SimCSE-BERT}_{\text{base}}$	Nvidia 3090 24 GB	Not reported	3	512	(x_i, x_i^+, x_i^-)	No
SBERT-base-nli-v2	Nvidia A100 80 GB	1	3	350	(x_i, x_i^+, x_i^-)	No
$\text{SelfCCL-BERT}_{\text{base}}$	Nvidia A100 80 GB	4	4	512	(x_i, x_i^+, x_i^-)	Yes
$\text{SelfCCL-SBERT}_{\text{base}}$	Nvidia A100 80 GB	1	4	350	(x_i, x_i^+, x_i^-)	Yes

5.4. First Experiment: Evaluate the Model for STS Tasks

We evaluate the SelfCCL model on seven standard STS tasks: STS 2012–2016 [58–62], STS-Benchmark [63], and SICK-Relatedness [64]. STS 2012-2016 and STS-Benchmark assign gold labels between 0 and 5, and SICK-Relatedness assign gold labels between 1 and 5 to the sentence pairs. Gold-labels show the semantic relatedness of the sentence pairs. Spearman's rank correlation between the cosine-similarity of the sentence embeddings and the gold labels is reported for each benchmark in columns.

A modified version of the SentEval toolkit from Ref. [12] is used to evaluate models on STS tasks. SentEval [65] is a general library for assessing the quality of sentence embeddings (https://github.com/facebookresearch/SentEval (accessed on 13 December 2022)). Ref. [12] released a modification of SentEval in their Github repository. They added the "all" setting to the original SentEval so that the overall Spearman's correlation was computed for all topics in sub-tasks. In addition, additional regressors were removed from SentEval by Ref. [12] for training frozen sentence embedding on the STS-B and SICK-R datasets.

Results: As shown in Table 4, SelfCCL-BERT and SelfCCL-SBERT models outperform baseline SBERT, SBERT-base-nli-v2, and SOTA SimCSE. For SelfCC-BERT, the corresponding improvements are 0.23 point compared to the SOTA SimCSE and 6.91 compared to baseline SBERT. For SelfCCL-SBERT, the corresponding improvements are 0.17 points compared to SBERT-base-nli-v2 and 6.88 compared to baseline SBERT. It is clear that we could improve the performance of SOTA SimCSE to some extent. Although, our models are still behind the SOTA SupMPN on STS tasks, according to the Table 3, our models require low computational power compared to it, and so they are more cost-effective in terms of hardware requirements.

Table 4. Assessment results on the seven STS tasks. Spearman's rank correlation as $\rho \times 100$ is given for each task in the columns. The best average result is in bold in the last column. †: [33], ‡: [9], ♠: [6], ♣: [10], ★: [56], ♦: [4], ♡: [7], ♢: [8], ■: [13] , □: [14], ∗: [15], SBERT-base-nli-v2: reproduced by ourselves, and the rest of the results are taken from Ref. [12].

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
fastText embeddings ‡	58.85	58.83	63.42	69.05	68.24	68.26	72.98	59.76
$BERT_{base}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$BERT_{base}$-flow-NLI	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$BERT_{base}$-whitening-NLI	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
CT-$BERT_{base}$	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
SG-$BERT_{base}$ ♢	66.84	80.13	71.23	81.56	77.17	77.23	68.16	74.62
Mirror-$BERT_{base}$ ♣	69.10	81.10	73.00	81.90	75.70	78.00	69.10	75.40
$SimCSE_{unsup}$-$BERT_{base}$	68.40	82.41	80.91	78.56	78.56	76.85	72.23	76.25
TSDAE-$BERT_{base}$ ★	55.02	67.40	62.40	74.30	73.00	66.00	62.30	65.80
ConSERT-$BERT_{base}$ ♦	70.53	79.96	74.85	81.45	76.72	78.82	77.53	77.12
ConSERT-$BERT_{large}$ ♦	73.26	82.37	77.73	83.84	78.75	81.54	78.64	79.44
DiffCSE-$BERT_{base}$ ■	72.28	84.43	76.47	83.90	80.54	80.59	71.23	78.49
miCSE–$BERT_{base}$ □	71.77	83.09	75.46	83.13	80.22	79.70	73.62	78.13
$RoBERTa_{base}$ (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
CLEAR-$RoBERTa_{base}$ ♡	49.00	48.90	57.40	63.60	65.60	72.50	75.60	61.08
DeCLUTR-$RoBERTa_{base}$ ‡	52.41	75.19	65.52	77.12	78.63	72.41	68.62	69.99

Table 4. *Cont.*

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
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
$SBERT_{base}$ †	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
$SBERT_{base}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
$SBERT_{base}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
CT-$SBERT_{base}$	74.84	83.20	78.07	83.84	77.93	81.46	76.42	79.39
$SBERT_{base}$-nli-v2	75.33	84.52	79.54	85.72	80.82	84.48	80.77	81.60
SelfCCL-$SBERT_{base}$	75.50	84.81	80.05	85.53	81.07	84.77	80.67	**81.77**
SG-$BERT_{base}$ ♢	75.16	81.27	76.31	84.71	80.33	81.46	76.64	79.41
$SimCSE_{sup}$-$BERT_{base}$	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
SupMPN-$BERT_{base}$ *	75.96	84.96	80.61	85.63	81.69	84.90	80.72	**82.07**
SelfCCL-$BERT_{base}$	75.61	84.72	80.04	85.44	81.37	84.63	80.82	81.80

5.5. Second Experiment: Evaluate the Model for Transfer Learning Tasks

We compare the SelfCCL model to the previous models on the seven SentEval transfer learning tasks as follows:

MR [66]: Binary sentiment prediction on movie reviews.
CR [67]: Binary sentiment prediction on customer product reviews.
SUBJ [68]: Binary subjectivity prediction on movie reviews and plot summaries.
MPQA [69]: Phrase-level opinion polarity classification.
SST-2 [70]: Stanford Sentiment Treebank with binary labels.
TREC [71]: Question type classification with six classes.
MRPC [72]: Microsoft Research Paraphrase Corpus for paraphrase prediction.

More details about these seven tasks can be found in Ref. [65]. In this experiment, the default configuration of the SentEval toolkit [65] was used, in which sentence embeddings serve as features for a logistic regression classifier. Then, the logistic regression classifier is trained with different tasks in a 10-fold cross-validation and the prediction accuracy is calculated for the test-folds. The results are given in Table 5.

Results: As shown in Table 5, SelfCCL-BERT and SelfCCL-SBERT outperform SOTA SimCSE and all previous models except for the baseline SBERT. As explained in Section 2, the baseline SBERT performs better than all other models on these seven classification tasks because it was trained on the classification objective. For SelfCC-BERT, the corresponding improvement is 0.39 point compared to the SOTA SimCSE. For SelfCCL-SBERT, the improvement is 0.13 points compared to SBERT-base-nli-v2. SelfCCL-SBERT also achieved an approximately similar average accuracy as SOTA SupMPN in transfer learning tasks, while according to Table 3, it requires less computational power compared to SOTA SupMPN. To sum up, using both methods, contrastive learning and curriculum learning, also show an improvement in average accuracy compared to cases where only contrastive learning is used.

5.6. Third Experiment: Cosine Similarity Distribution

In this experiment, considering that the supervised contrastive based models have a higher average improvement, we study the correlation of cosine similarity between sentence embeddings generated by our models and baseline models (BERT, SBERT) and human-scored on the SICK dataset [64].

Table 5. Assessment results for the seven transfer learning tasks. The prediction accuracy is given for each task in the columns. The best average result is in bold in the last column. †: [33], ♠: [6], ♢: [8], ∞: [11], ■: [13], ∗: [15], SBERT-base-nli-v2: reproduced by ourselves, and the rest of the results are taken from [12].

Model	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
Unsupervised models								
Glove embeddings (avg.) †	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Skip-thought ♠	76.50	80.10	93.60	87.10	82.00	92.20	73.00	83.50
Avg. BERT embedding †	78.66	86.25	94.37	88.66	84.40	92.80	69.54	84.94
BERT-[CLS] embedding †	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
IS-BERT$_{\text{base}}$ ♠	81.09	87.18	94.96	88.75	85.96	88.64	74.24	85.83
CT-BERT$_{\text{base}}$ ∞	79.84	84.00	94.10	88.06	82.43	89.20	73.80	84.49
SimCSE$_{\text{unsup}}$-BERT$_{\text{base}}$	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.51
SimCSE$_{\text{unsup}}$-BERT$_{\text{base}}$-MLM	82.92	87.23	95.71	88.73	86.81	87.01	78.07	86.64
DiffCSE-BERT$_{\text{base}}$ ■	82.69	78.23	95.23	89.28	86.60	90.40	76.58	86.86
Supervised models								
InferSent-GloVe †	81.57	86.54	92.50	90.38	84.18	88.20	75.77	85.59
Universal Sentence Encoder †	80.09	85.19	93.98	86.70	86.38	93.20	70.14	85.10
SBERT$_{\text{base}}$ †	83.64	89.43	94.39	89.86	88.96	89.60	76.00	**87.41**
SBERT$_{\text{base}}$-nli-v2	83.09	89.33	94.98	90.15	87.92	87.00	75.25	86.82
SelfCCL-SBERT$_{\text{base}}$	83.02	89.46	95.05	90.18	87.97	87.00	75.94	86.95
SG-BERT$_{\text{base}}$ ♢	82.47	87.42	95.40	88.92	86.20	91.60	74.21	86.60
SimCSE$_{\text{sup}}$-BERT$_{\text{base}}$	82.69	89.25	84.81	89.59	87.31	88.40	73.51	86.51
SupMPN-BERT$_{\text{base}}$ ∗	82.93	89.26	94.76	90.21	86.99	88.20	76.35	86.96
SelfCCL-BERT$_{\text{base}}$	82.89	89.22	94.78	90.51	87.15	87.60	76.17	86.90

SICK, an acronym for Sentences Involving Compositional Knowledge, contains about 10,000 sentence pairs rich in lexical, syntactic, and semantic phenomena. Each sentence pair is annotated in two forms: relatedness and entailment.

The human relatedness score ranges from 1 to 5; the entailment relation is categorical, consisting of entailment, contradiction, and neutral. We use a version of SICK that is located in the SentEval Github repository. There are 4500 pairs in the train-split, 500 in the trial-split used for development, and 4927 in the test-split. We analyzed the test-split statistics. There are 720 pairs with a contradiction relation that average the human relatedness scores of them at about 3.58 (whose normalized value in the range of {0,1} via division by 5 is 71.60). That is, although they have contradiction relation, they are very similar to each other based on word embeddings. Table 6 shows some sentences pairs with contradiction relations from the SICK test dataset.

Table 6. Example sentence pairs of the SICK [64] test-split data with the contradiction relation, which have high human relatedness scores.

Sentence 1	Sentence 2	Relation	Human Relatedness Scores
A girl is brushing her hair.	There is no girl brushing her hair.	Contradiction	4.5
A chubby faced boy is not wearing sunglasses.	A chubby faced boy is wearing sunglasses.	Contradiction	3.9
The dog is on a leash and is walking in the water.	The dog is on a leash and is walking out of the water.	Contradiction	3.5
A black sheep is standing near three white dogs.	A black sheep is standing far from three white dogs.	Contradiction	3.5
A man dressed in black and white is holding up the tennis racket and is waiting for the ball.	A man dressed in black and white is dropping the tennis racket and is waiting for the ball.	Contradiction	4

We use "bert-base-uncased" (https://huggingface.co/bert-base-uncased (accessed on 13 December 2022)) and "nli-bert-base" (https://huggingface.co/sentence-transformers/nli-bert-base (accessed on 13 December 2022)) checkpoints for BERT and SBERT, respectively. We have demonstrated the results in Figure 4, where the x-axis represents human relatedness scores, which are normalized in the range {0,1} via division by 5, and the y-axis represents cosine similarity between sentence-embedding pairs generated by models. The color coding corresponds to the human relatedness scores.

Results: As show in Figure 4, almost all sentence pairs generated by BERT have similarities in the range between 0.6 and 1. The SBERT model, considering that it uses softmax-classification for classifying entailment, neutral, and contradiction pairs, pushes contradictions farther away from anchors. So, for sentence pairs of contradiction relation with a high relatedness score, SBERT embeddings have a low cosine similarity. Compared to original BERT-base and original SBERT-base, our models better discriminate similar and dissimilar sentences. So, there are stronger correlations between cosine similarity of embedding and human relatedness scores. The underlying reason is that the SelfCCL models use a contrastive objective, which increases the distance of negative sentences from anchors in comparison to the distance of positive sentences from anchors.

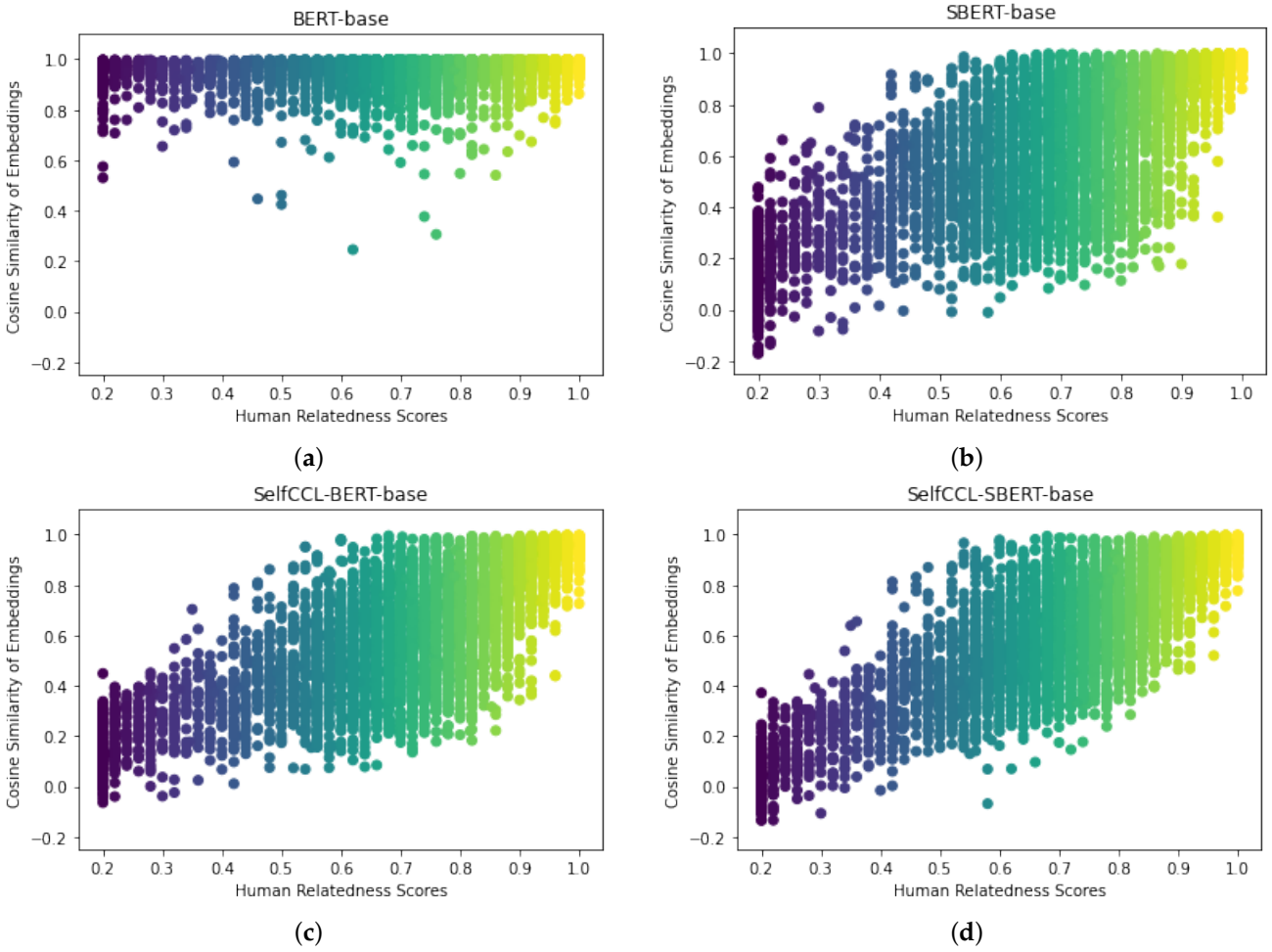


Figure 4. Scatter plots of cosine similarity of sentence embeddings and human relatedness scores on SICK test split for (**a**) BERT, (**b**) SBERT, (**c**) SelfCCL-BERT, and (**d**) SelfCCL-SBERT models. The color coding corresponds to the human relatedness scores.

6. Conclusions and Future Works

Recently, many contrastive learning methods have been proposed to fine-tune BERT in the STS downstream task, which learns from the contrast between similar and dissimilar sentences. However, the effect of the order of difficulty level of triplets during training has not been investigated in them. In this work, our aim was to study the effect of using both contrastive learning and curriculum learning, simultaneously. For this purpose, we proposed a curriculum contrastive learning model by transferring self-taught knowledge for fine-tuning BERT, which first sorts the triplets based on curriculum order by transferring BERT self-taught knowledge, and then learns based on a contrastive objective. We developed SelfCCL-BERT and SelfCCL-SEBERT models based on BERT and SBERT architectures. We evaluated our models on standard STS and transfer learning tasks.

Our models, SelfCCL-BERT and SelfCCL-SBERT, surpasses baseline model SBERT, SBERT-base-nli-v2, and SOTA SimCSE model in terms of average Spearman's rank correlation on STS tasks. Furthermore, our models, SelfCCL-BERT and SelfCCL-SBERT, outperforms SOTA SimCSE on transfer learning tasks. Although our models fell behind SOTA SupMPN model on the STS tasks, it still has good performance compared to other baselines and requires low computational power compared to SOTA SupMPN, according to Table 3.

Moreover, as explained in Section 5.1, SelfCCL-SBERT requires less computational power than SelfCCL-BERT, while it achieves competitive performance compared to SelfCCL-BERT on both STS and transfer learning tasks.

Besides, the analysis of correlation between human-annotated scores and cosine similarity of sentence embeddings generated by our models on SICK dataset shows that our SelfCCL models learn better representation space compared to the baseline models.

In summary, the empirical results show that the use of curriculum learning together with contrastive learning leads to a relatively small improvement in the efficiency of the model for measuring semantic similarity between texts.

The challenge that can be discussed here is the conflict between contradiction and similarity concepts. As explained in the third experiment, two sentences can be semantically contradictory but conceptually completely similar. As a future research direction, this challenge can be investigated for the STS task.

For future studies on curriculum learning, we plan to investigate curriculum data augmentation. Curriculum data augmentation involves gradually increasing the noise in the data to generate new data. For example, the synonym replacement method involves gradually increasing the number of words that are replaced with similar words using WordNet or contextual word embeddings, e.g., BERT.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, S.D.; supervision, M.F.A. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Grant No: 120E100.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Our pre-trained models and training data are publicly available at: https://github.com/SoDehghan/SelfCCL (accessed on 13 December 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long-Short Term Memory
CL	Curriculum Learning
ELMo	Embeddings from Language Model
MNRL	Multiple Negatives Ranking Loss
NLI	Natural Language Inference
NLP	Natural Language Processing
NT-Xent	Normalized Temperature-scale Cross-Entropy
SBERT	Sentence BERT
SOTA	State-Of-The-Art
STS	Semantic Textual Similarity
USE	Universal Sentence Encoder

References

1. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 3–5 June 2019; Volume 1, pp. 4171–4186. [CrossRef]
2. Sina J.S.; Sadagopan, K.R. BERT-A: Fine-Tuning BERT with Adapters and Data Augmentation. 2019. Available online: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15848417.pdf (accessed on 30 November 2022).
3. Flender, S. What Exactly Happens When We Fine-Tune BERT? 2022. Available online: https://towardsdatascience.com/what-exactly-happens-when-we-fine-tune-bert-f5dc32885d76 (accessed on 30 November 2022).
4. Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; Xu, W. 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, Virtual Event, 1–6 August 2021; Volume 1. [CrossRef]
5. Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; Li, L. On the Sentence Embeddings from Pre-trained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 16–20 November 2020; pp. 9119–9130. [CrossRef]
6. Zhang, Y.; He, R.; Liu, Z.; Lim, K.H.; Bing, L. An unsupervised sentence embedding method by mutual information maximization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020 ; pp. 1601–1610. [CrossRef]
7. Wu, Z.; Sinong, S.; Gu, J.; Khabsa, M.; Sun, F.; Ma, H. CLEAR: Contrastive Learning for Sentence Representation. *arXiv* **2020**, arXiv:2012.15466.
8. Kim, T.; Yoo, K.M.; Lee, S. 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, Virtual Event, 1–6 August 2021; Volume 1. [CrossRef]
9. Giorgi, J.; Nitski, O.; Wang, B.; Bader, G. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual Event, 1–6 August 2021; Volume 1. [CrossRef]
10. Liu, F.; Vulić, I.; Korhonen, A.; Collier, N. 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, Punta Cana, Dominican Republic, 7–11 November 2021. [CrossRef]
11. Carlsson, F.; Gyllensten, A.C.; Gogoulou, E.; Hellqvist, E.Y.; Sahlgren, M. Semantic Re-Tuning with Contrastive Tension. International Conference on Learning Representations (ICLR). 2021. Available online: https://openreview.net/pdf?id=Ov_sMNau-PF (accessed on 30 November 2022).
12. Gao, T.; Yao, X.; Chen, D. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021. [CrossRef]
13. Chuang, Y.-S.; Dangovski, R.; Luo, H.; Zhang, Y.; Chang, S.; Soljačić, M.; Li, S.-W.; Yih, W.-T.; Kim, Y.; Glass, J. Diffcse: Difference-based contrastive learning for sentence embeddings. *arXiv* **2022**, arXiv:2204.10298.
14. Klein, T.; Nabi, M. miCSE: Mutual Information Contrastive Learning for Low-shot Sentence Embeddings. *arXiv* **2022**, arXiv:2211.04928.
15. Dehghan, S.; Amasyali, M.F. SupMPN: Supervised Multiple Positives and Negatives Contrastive Learning Model for Semantic Textual Similarity. *Appl. Sci.* **2022**, *12*, 9659. [CrossRef]
16. Kamath, U.; Liu, J.; Whitaker, J. Transfer Learning: Scenarios, Self-Taught Learning, and Multitask Learning. In *Deep Learning for NLP and Speech Recognition*; Springer: Cham, Switzerland, 2019. [CrossRef]
17. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.

18. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014. [CrossRef]
19. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with sub word information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. ._a_00051. [CrossRef]
20. Porner, N.M. Combining Contextualized and Non-Contextualized Embeddings for Domain Adaptation and Beyond. Available online: https://edoc.ub.uni-muenchen.de/27663/1/Poerner_Nina_Mareike.pdf (accessed on 30 November 2022).
21. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML 2014), Beijing, China, 21–26 June 2014; pp. 1188–1198. [CrossRef]
22. Kiros, R.; Zhu, Y.; Salakhutdinov, R.; Zemel, R.; Torralba, A.; Urtasun, R.; Fidler, S. Skip-thought vectors. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, USA, 7–12 December 2015; pp. 3294–3302. [CrossRef]
23. Hill, F.; Cho, K.; Korhonen, A. Learning Distributed Representations of Sentences from Unlabelled Data. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1367–1377. [CrossRef]
24. Pagliardini, M.; Gupta, P.; Jaggi, M. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 1, pp. 528–540. [CrossRef]
25. Logeswaran, L.; Lee, H. An efficient framework for learning sentence representations. *arXiv* **2018**, arXiv:1803.02893.
26. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018. [CrossRef]
27. Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; Bordes, A. Supervised learning of universal sentence representations from natural language inference data. *arXiv* **2017**, arXiv:1705.02364.
28. Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. Universal Sentence Encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, 31 October–4 November 2018; pp. 169–174. [CrossRef]
29. Prottasha, N.J.; Sami, A.A.; Kowsher, M.; Murad, S.A.; Bairagi, A.K.; Masud, M.; Baz, M. Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning. *Sensors* **2022**, *22*, 4157. [CrossRef] [PubMed]
30. Kim, M.G.; Kim, M.; Kim, J.H.; Kim, K. Fine-Tuning BERT Models to Classify Misinformation on Garlic and COVID-19 on Twitter. *Int. J. Environ. Res. Public Health* **2022**, *19*, 5126. [CrossRef] [PubMed]
31. Agrawal, A.; Tripathi, S.; Vardhan, M.; Sihag, V.; Choudhary, G.; Dragoni, N. BERT-Based Transfer-Learning Approach for Nested Named-Entity Recognition Using Joint Labeling. *Appl. Sci.* **2022**, *12*, 976. [CrossRef]
32. Fernández-Martínez, F.; Luna-Jiménez, C.; Kleinlein, R.; Griol, D.; Callejas, Z.; Montero, J.M. Fine-Tuning BERT Models for Intent Recognition Using a Frequency Cut-Off Strategy for Domain-Specific Vocabulary Extension. *Appl. Sci.* **2022**, *12*, 1610. [CrossRef]
33. Reimers, N.; Gurevych, I. 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), Hong Kong, China, 3–7 November 2019. [CrossRef]
34. Bowman, S.R.; Angeli, G.; Potts, C.; Manning, C.D. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015. [CrossRef]
35. Williams, A.; Nangia, N.; Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 1. [CrossRef]
36. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006. [CrossRef]
37. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005. [CrossRef]
38. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. *arXiv* **2015**, arXiv:1503.03832.
39. Xuan, H.; Stylianou, A.; Liu, X.; Pless, R. Hard negative examples are hard, but useful. In *ECCV 2020: Computer Vision—ECCV 2020*; Springer: Cham, Switzerland, 2020. [CrossRef]
40. Sikaroudi, M.; Ghojogh, B.; Safarpoor, A.; Karray, F.; Crowley, M.;Tizhoosh, H.R. Offline versus Online Triplet Mining based on Extreme Distances of Histopathology Patches. *arXiv* **2020**, arXiv:2007.02200.
41. Gao, L.; Zhang, Y.; Han, J.; Callan, J. Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup. *arXiv* **2021**, arXiv:2101.06983.
42. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv* **2020**, arXiv:2002.05709.

43. Sohn, K. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016. Available online: https://proceedings.neurips.cc/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf (accessed on 30 November 2022).
44. Elman, J.L. Learning and development in neural networks: The importance of starting small. *Cognition* **1993**, *48*, 71–99. [CrossRef]
45. Bengio, Y.; Louradour, J.O.; Collobert, R.; Weston, J. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 41–48. [CrossRef]
46. Soviany, P.; Ionescu, R.T.; Rota, P. Curriculum Learning: A Survey. *Int. J. Comput. Vis.* **2022**, *130*, 1526–1565. [CrossRef]
47. Wang, X.; Chen, Y.; Zhu, W. A Survey on Curriculum Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 4555–4576. [CrossRef]
48. Wu, X.; Dyer, E.; Neyshabur, B. When do curricula work? *arXiv* **2021**, arXiv:2012.03107.
49. Hacohen, G.; Weinshall, D. On The Power of Curriculum Learning in Training Deep Networks. *arXiv* **2019**, arXiv.1904.03626.
50. Yegin, M.N.; Kurttekin, O.; Bahsi, S.K.; Amasyali, M.F. Training with growing sets: A comparative study. *Expert Syst.* **2022**, *39*, e12961. [CrossRef]
51. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *IEEE* **2021**, *109*, 43–76. [CrossRef]
52. Gholizade, M.; Soltanizadeh, H.; Rahmanimanesh, M. A Survey of Transfer Learning and Categories. *Model. Simul. Electr. Electron. Eng. J.* **2021**, *1*, 17–25. [CrossRef]
53. Raina, R.; Battle, A.; Lee, H.; Packer, B.; Ng, A.Y. Self-taught learning: Transfer learning from unlabeled data. In Proceedings of the 24th Annual International Conference on Machine Learning held in conjunction with the 2007 International Conference on Inductive Logic Programming, Corvalis, OR, USA, 20–24 June 2007; pp. 759–766. [CrossRef]
54. Henderson, M.; Al-Rfou, R.; Strope, B.; Sung, Y.; Lukacs, L.; Guo, R.; Kumar, S.; Miklos, B.; Kurzweil, R. Efficient Natural Language Response Suggestion for Smart Reply. *arXiv* **2017**, arXiv:1705.00652.
55. Su, J.; Cao, J.; Liu, W.; Ou, Y. Whitening sentence representations for better semantics and faster retrieval. *arXiv* **2021**, arXiv:2103.15316.
56. Wang, K.; Reimers, N.; Gurevych, I. TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP, Punta Cana, Dominican Republic, 16–20 November 2021. [CrossRef]
57. Muennighoff, N. SGPT: GPT Sentence Embeddings for Semantic Search. *arXiv* **2022**, arXiv:2202.08904.
58. Agirre, E.; Cer, D.; Diab, M.; Gonzalez-Agirre, A. 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, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*; Association for Computational Linguistics: Atlanta, GA, USA, 2012; pp. 385–393. Available online: https://aclanthology.org/S12-1051 (accessed on 30 November 2022).
59. Agirre, E.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W. *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*; Association for Computational Linguistics: Atlanta, GA, USA, 2013; pp. 32–43. Available online: https://aclanthology.org/S13-1004 (accessed on 30 November 2022).
60. Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Mihalcea, R.; Rigau, G.; Wiebe, J. SemEval-2014 task 10: Multilingual semantic textual similarity. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 81–91. Available online: https://aclanthology.org/S14-2010 (accessed on 30 November 2022).
61. Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Lopez-Gazpio, I.; Maritxalar, M.; Mihalcea, R.; et al. 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), Denver, CO, USA, 4–5 June 2015; pp. 252–263. [CrossRef]
62. Agirre, E.; Banea, C.; Cer, D.; Diab, M.; Gonzalez Agirre, A.; Mihalcea, R.; Rigau Claramunt, G.; Wiebe, J. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) Association for Computational Linguistics, San Diego, CA, USA, 16–17 June 2016; pp. 497–511. [CrossRef]
63. Cer, D.; Diab, M.; Agirre, E.; LopezGazpio, I.; Specia, L. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 1–14. [CrossRef]
64. Marelli, M.; Menini, S.; Baroni, M.; Entivogli, L.; Bernardi, R.; Zamparelli, R. A SICK cure for the evaluation of compositional distributional semantic models. In Proceedings of the International Conference on Language Resources and Evaluation (LREC), Reykjavik, Iceland, 26–31 May 2014; pp. 216–223. Available online: https://aclanthology.org/L14-1314/ (accessed on 30 November 2022).
65. Conneau, A.; Kiela, D. SentEval: An evaluation toolkit for universal sentence representations. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC), Miyazaki, Japan, 7–12 May 2018. Available online: https://aclanthology.org/L18-1269 (accessed on 30 November 2022).

66. Pang, B.; Lee, L. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, MI, USA, 25–30 June 2005; pp. 115–124. [CrossRef]
67. Hu M.; Liu, B. Mining and Summarizing Customer Reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 168–177. Available online: https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf (accessed on 30 November 2022).
68. Pang, B.; Lee, L. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume, Barcelona, Spain, 21–26 July 2004; pp. 271–278. Available online: https://aclanthology.org/P04-1035 (accessed on 30 November 2022).
69. Wiebe, J.; Wilson, T.; Cardie, C. Annotating Expressions of Opinions and Emotions in Language. *Lang. Resour. Eval.* **2005**, *39*, 165–210. [CrossRef]
70. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.; Potts, C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642. Available online: https://aclanthology.org/D13-1170/ (accessed on 30 November 2022).
71. Li, X.; Roth, D. Learning Question Classifiers. In Proceedings of the 19th International Conference on Computational Linguistics—Volume 1, COLING, Taipei, Taiwan, 26–30 August 2002; pp. 1–7. Available online: https://aclanthology.org/C02-1150/ (accessed on 30 November 2022).
72. Dolan, B.; Quirk, C.; Brockett, C. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004, Geneva, Switzerland, 23–27 August 2004; Available online: https://aclanthology.org/C04-1051 (accessed on 30 November 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.