applied sciences

MDPI

Article

SupMPN: Supervised Multiple Positives and Negatives Contrastive Learning Model for Semantic Textual Similarity

Somaiyeh Dehghan * and Mehmet Fatih Amasyali

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

Abstract: Semantic Textual Similarity (STS) is an important task in the area of Natural Language Processing (NLP) that measures the similarity of the underlying semantics of two texts. Although pre-trained contextual embedding models such as Bidirectional Encoder Representations from Transformers (BERT) have achieved state-of-the-art performance on several NLP tasks, BERT-derived sentence embeddings have been proven to collapse in some way, i.e., sentence embeddings generated by BERT depend on the frequency of words. Therefore, almost all BERT-derived sentence embeddings are mapped into a small area and have a high cosine similarity. Hence, sentence embeddings generated by BERT are not so robust in the STS task as they cannot capture the full semantic meaning of the sentences. In this paper, we propose SupMPN: A Supervised Multiple Positives and Negatives Contrastive Learning Model, which accepts multiple hard-positive sentences and multiple hard-negative sentences simultaneously and then tries to bring hard-positive sentences closer, while pushing hard-negative sentences away from them. In other words, SupMPN brings similar sentences closer together in the representation space by discrimination among multiple similar and dissimilar sentences. In this way, SupMPN can learn the semantic meanings of sentences by contrasting among multiple similar and dissimilar sentences and can generate sentence embeddings based on the semantic meaning instead of the frequency of the words. We evaluate our model on standard STS and transfer-learning tasks. The results reveal that SupMPN outperforms state-of-the-art SimCSE and all other previous supervised and unsupervised models.

Keywords: Natural Language Processing; sentence embedding; Semantic Textual Similarity; BERT; contrastive learning; deep learning

Citation: Dehghan, S.; Amasyali, M.F. SupMPN: Supervised Multiple Positives and Negatives Contrastive Learning Model for Semantic Textual Similarity. *Appl. Sci.* **2022**, *12*, 9659. https://doi.org/10.3390/app12199659

Academic Editor: Valentino Santucci

Received: 18 August 2022
Accepted: 20 September 2022
Published: 26 September 2022

1. Introduction

Semantic Textual Similarity (STS) is one of the fundamental tasks in the Natural Language Processing (NLP) that measures similarity between two texts regardless of having or not having common words. In fact, STS deals with computing the similarity between conceptually similar, but not necessarily lexically similar texts. STS plays a significant role in many NLP applications including information retrieval [1,2], text summarization [3,4], text classification [5,6], sentiment analysis [7,8], question answering [9,10], machine translation [11,12], entity recognition [13], etc.

Although recent sentence-encoding models such as BERT [14] have been very successful in some NLP tasks, without fine-tuning, the sentence embeddings generated with them fail to capture all the semantic meaning of sentences in the STS task and have low quality [15], i.e., they are mapped into a small area and so the cosine similarity scores between almost all sentence pairs are in the range of 0.6 and 1.0 [16]. This problem, which is referred to as the collapse issue of BERT in [16,17], stems from the anisotropic space of representation, as sentence embeddings generated by BERT depend on the frequency of words [15,16]. Therefore, if two sentences have some words in common, the cosine similarity between them is high regardless of whether they have completely different semantic meanings. For example, consider the following three sentences in BERT representation

space: "A woman is walking across the street eating a banana, while a man is following with his briefcase", "A woman eats ice cream walking down the sidewalk, and there is another woman in front of her with a briefcase", "A person eating crosses the street." The cosine similarity between the first and the second sentences is more than the cosine similarity between the first and the third sentences, while the first and the second sentences have completely different semantic meanings.

One of the general strategies to prevent these problems and achieve the desired performance on the STS task is to use contrastive learning for fine-tuning BERT [17]. Contrastive learning is a deep metric learning approach, the main idea of which is to learn such a representation space where similar sample pairs (anchor-positive) are brought closer together, while dissimilar sample pairs (anchor-negative) are pushed further away. Contrastive learning has shown outstanding performance on computer-vision tasks such as human-activity recognition [18–23], person re-identification [24,25], object detection [26,27], image classification [28,29], image processing [30,31], etc. Contrastive learning can be used in two ways, self-supervised and supervised settings. The self-supervised contrastive learning contrasts a single anchor-positive pair in a batch against many negative samples (other classes' positive samples in a batch), while supervised contrastive learning contrasts a set of all positive samples from the same class against many negatives (other classes' positive samples in a batch) [32].

Recently, contrastive learning has been used for fine-tuning the BERT pre-trained language model [16,33–37]. The method in [16,33–36] employs self-supervised NT-Xent loss from SimCLR paper [38] which accepts positives sentence pairs in the form of anchor-positive (x_i, x_i^+) and uses other input positive pairs as negative examples for contrastive learning. After this, ref. [37] proposed a supervised $\text{SimCSE}_{\text{sup}}$ model by applying a hard negative to NT-Xent loss which accepts triplets in the form of anchor-positive-negative (x_i, x_i^+, x_i^-).

However, recent literature on contrastive learning has attempted to enhance its discrimination performance by including multiple hard positives [32,39] or multiple hard negatives [40,41]. Therefore, in this study, we aim to answer this question: how can we benefit from the advantages of both multiple hard positives and multiple hard negatives to boost the performance of contrastive learning in fine-tuning BERT?

To address this question, we propose SupMPN: A Supervised Multiple Positives and Negatives Contrastive Learning Model via the extension of an objective function: Supervised Multiple Positives and Negatives Ranking Loss. For convenience, we abbreviate the foregoing to SupMPNRL. Our objective function accepts triplets in the form of $(x_i, x_{i1}^+, \ldots, x_{iP}^+, x_{i1}^-, \ldots, x_{iQ}^-)$, in which $(x_{i1}^+, \ldots, x_{iP}^+)$ and $(x_{i1}^-, \ldots, x_{iQ}^-)$ act as hard positives and hard negatives for anchor sentence x_i, respectively.

As deep-learning models require a lot of labeled data for training [42], we use Natural Language Inference (NLI) datasets including SNLI [43] and Multi-genre NLI (MNLI) [44] for training our model. NLI is the task of determining whether a hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given a premise sentence. SNLI and MNLI are two large-scale collections of human-labeled English sentence pairs with the labels entailment, contradiction, and neutral. We use entailment and contradiction hypotheses of SNLI and MNLI as hard positives and hard negatives, respectively.

Using multiple hard positives generalizes simple triplet loss to a desired number of positives, enabling the model to bring similar sentences closely into the embedding space, while using multiple hard negatives generalizes simple triplet loss to improve distinction among positive and negative sentences, enabling the model to push dissimilar sentences away in the embedding space. In this way, SupMPN can learn semantic meanings of sentences by contrasting among multiple similar and dissimilar sentences and can generate sentence embeddings based on the semantic meaning instead of the frequency of the words.

For a comprehensive comparison, we conduct three experiments. In the first experiment, we evaluate SupMPN on seven standard STS tasks. In the second experiment, we evaluate SupMPN on seven standard transfer-learning tasks. In the third experiment, we

compare Semantic Textual Similarity using sentence embeddings from BERT, state-of-the-art SimCSE, and SupMPN for some sentences.

Our main contributions and findings are as follows, in brief.

- We propose SupMPN model to fine-tune BERT so that it can generate sentence embeddings based on the semantic meaning instead of the frequency of the words.
- We provide a new contrastive objective function that involves the multiple hard positives and multiple hard negatives in contrasting learning simultaneously.
- Adding multiple hard positives and multiple hard negatives to contrastive learning boosts its performance by discrimination among multiple similar and dissimilar sentences.
- By contrasting among multiple similar and dissimilar sentences, our model can learn the semantic meaning of sentences and can generate better sentence representation space.
- Our model outperforms state-of-the-art SimCSE and all other previous supervised and unsupervised models.

The rest of the paper is structured as follows: In Section 2, a basic overview of sentence-embedding models is provided under the "Related Works" heading. In Section 3, a brief background on deep metric learning and contrastive learning is performed. In Section 4, our proposed model (SupMPN) is presented. In Section 5, the experiments are given. Finally, conclusions are provided in Section 6.

2. Related Works

The main and certainly the most common challenge for all NLP tasks, is the way to represent textual data as input. The traditional embedding techniques such as word2vec [45] and GloVe [46] only work on the word level. After this, the authors in [47] introduced a sentence-embedding model, called Skip-thought, which is an extension of the word2vec skip-gram method to apply sentences instead of words, i.e., instead of predicting the target word using surrounding words, Skip-thought predicts the target sentence by surrounding sentences. Later, ref. [48] presented fastText, which had the same goal as word2vec with a small difference: fastText, unlike word2vec, which treats each word in corpus such as an atomic entity, uses character n-grams and so can process out-of-vocabulary words.

Although word2vec, GloVe, and fastText encode words into a vector representation, there is still a need to represent whole sentences so that a computer can easily understand their semantic meanings in the entire text. Another simple way to create a single fixed-size sentence vector is averaging word vectors (AWV). This solution considers neither the interaction between words in a sentence, nor the order of words, while a good sentence encoder is expected to be able to create deep contextualized word representations that can handle polysemy (words with multiple meanings in different contexts).

Recently, deep contextualized word-embedding models such as Facebook's InferSent [49], AllenAI's ELMo [50], Google's BERT [14], and USE [51] have been proposed and received significant attention. InferSent is a Siamese network that uses Bi-LSTM, a deep neural network with memory to remember the whole sentence to encode. It is a supervised model trained on the Natural Language Inference (NLI) dataset. ELMo uses the Bi-LSTM network trained on a huge corpus including billions of words. USE have two encoder variations, e.g., one is the transformers encoder and the other is the Deep Averaging Network (DAN) trained on supervised data (e.g., NLI), and unsupervised data which are drawn from a variety of web sources such as Wikipedia, web news, question–answer pages, and discussion forums.

Finally, BERT, the contextualized word-embedding model, was proposed. BERT achieves the state-of-the-art performance in some of NLP tasks. It is a giant deep neural network with millions of parameters and uses a cross-encoder architecture. The cross-encoder architecture of BERT requires that the two sentences be passed to the network simultaneously which lead to greater computational overheads [52,53]. Therefore, this feature makes training BERT from scratch very time-consuming for sentence-pair tasks

such as semantic similarity search. On the other hand, BERT uses Mask Language Model (MLM) and Next Sentence Prediction (NSP) objectives. The MLM objective enables BERT to learn word-level or phrase-level semantic relationships [54] and the NSP objective enables BERT to learn longer-term dependencies across sentences. With the NSP objective, BERT only answers to this question: given sentence A and B, is B the next sentence for A? Meanwhile, the STS task wants to answer this question: given sentence A and B, are A and B similar or not?

As an effective solution to solve these problems, Sentence-BERT (SBERT), a modification of the BERT using Siamese network, has been proposed by [52]. SBERT uses bi-encoder architecture through training a Siamese network on top of the BERT model and pays attention to the sentence-level semantic relation in its training objective. Although SBERT increases BERT performance on STS tasks through training with human-labeled NLI datasets, it still fails to produce good sentence embeddings.

Recently, several models have been proposed to enhance BERT and/or RoBERTa [55] on Semantic Textual Similarity (STS) using contrastive learning. In these models, contrastive learning is used to fine-tune BERT and improve the quality of sentence-embedding space [16,33–37,56,57]. Their main idea is to bring similar sentences closer to each other and push dissimilar sentences far away by a contrastive objective. We will briefly describe each of them in Section 5.4.

3. Background

3.1. Deep Metric Learning

Deep metric learning aims to automatically learn an embedding space model so that similar samples are placed into nearby space, while dissimilar samples are pushed away using the Euclidean or cosine distance. The common objective functions used in deep metric learning are contrastive loss and triplet loss which were first proposed in image processing and computer vision tasks [58–60].

In the contrastive learning, the simplest and the oldest method, proposed by [58,59], there is a pair of embedding vectors (x_i, x_j) and a label, either 1 or 0. If the embedding pair is from the same class, this label will be 1 and objective function tries to reduce the distance between them. Otherwise, the label will be 0 and objective function tries to increase the distance between them.

In triplet learning, which was first introduced for face recognition in [60], the loss is computed over triplets of an anchor, a positive and a negative sample (x_i, x_i^+, x_i^-), so that the distance between anchor and positive pairs must be less than the distance between anchor and negative pairs. The simple triplet loss is computed as:

$$L_{TL} = \max\left(D(x, x^+) - D(x, x^-) + m,\ 0\right) \tag{1}$$

where $D(\cdot)$ is a metric function for measuring distance, which can be Euclidean or cosine distance, and m is a margin, which is a hyper-parameter used to determine how far the dissimilar images (or dissimilar sentences) should be from the anchor image (or anchor sentence).

3.2. Triplet Selection

The main challenge when training with triplet loss is how to prepare input triplets (x, x^+, x^-), because as the amount of training data increases, the number of possible triplets increases cubically [39]. The effectiveness of the triplet loss relies strongly on the triplet selection. On the other hand, selecting useful triplets is important to ensure fast convergence [60,61]. This means that, given an anchor x, we want to select hard positive and hard negative such that:

Hard positive: $argmax(D(x, x^+))$
Hard negative: $argmin(D(x, x^-))$

3.3. Offline and Online Triplet Mining

Preparing triplets can be offline or online which is referred to as offline/online triplet mining. In the offline triplet mining method, triplets are prepared before the training begins, e.g., at the start of each epoch, by computing all the possible triplets on the training data, and then the hardest of them are selected. However, this approach is very inefficient since computing all possible triplets and choosing hard triplets at the start of each epoch is very time-consuming. In the online triplet mining, also known as batch-wise approach or technique of in-batch negative [62], the idea is to prepare triplets during the training step within a mini-batch of data [60,63], where for each anchor in a batch, other in-batch positives and negatives are taken as negatives. There are several contrastive loss functions based on online triplet mining in the literature. Here, we introduce five of them. The first two are based on margin and the rest are based on cross-entropy.

BatchAll and BatchHard: Ref. [39] proposed two online (batch-wise) margin-based triplet losses (BatchHard and BatchAll) which only deal with positive samples from each class. The way they work is to accept a batch with $P \times K$ samples (P classes with K positive instances per each class). Therefore, for each class in the batch, the other classes' positives act as negatives samples. BatchAll loss function is in the simple margin-based triplet loss format, is summed over all the possible triplets in the mini-batch, and was formulated as:

$$L_{BA} = \overbrace{\sum_{i=1}^{P}\sum_{a=1}^{K}}^{all\ anchors} \quad \overbrace{\sum_{\substack{p=1 \\ p\neq a}}^{K}}^{all\ positives} \quad \overbrace{\sum_{\substack{j=1 \\ j\neq i}}^{P}\sum_{n=1}^{K}}^{all\ negatives} \left[D\left(x_a^i, x_p^i\right) - D(x_a^i, x_n^j) + m\right]_+ \tag{2}$$

where m is the margin, $D(\cdot)$ is the metric distance function, and $[\cdot]_+ := max(\cdot, 0)$ is the standard hinge loss [64].

In the BatchHard loss function, for each anchor x_a^i the Hardest Positive and Hardest Negative (HPHN), which are the farthest positive and nearest negative in the mini-batch, are selected. Hence, its loss function is:

$$L_{BH} = \overbrace{\sum_{i=1}^{P}\sum_{a=1}^{K}}^{all\ anchors} \left(\overbrace{\max_{p=1\ldots K} D\left(x_a^i, x_p^i\right)}^{hardest\ positive} - \underbrace{\min_{\substack{j=1\ldots P \\ n=1\ldots K \\ j\neq i}} D(x_a^i, x_n^j)}_{hardest\ negative} + m \right)_+ \tag{3}$$

Normalized Temperature-Scaled Cross-Entropy loss (NT-Xent): Ref. [38] proposed a batch-wise self-supervised contrastive loss which is a modification of the multi-class N-pair loss [40] with addition of the temperature parameter τ to scale the cosine similarities. This loss function only accepts positive pairs in the form of (x_i, x_i^+). That is, each sample in the batch belongs to just one positive pair and all other possible pairs with x_i are negative pairs (denominator). Therefore, for a positive pair of (x_i, x_i^+) within a mini-batch of N pairs, the loss function is defined as:

$$L_{NT-Xent} = -\log \frac{e^{sim(x_i, x_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(x_i, x_j^+)/\tau}} \tag{4}$$

where $sim(\cdot)$ is the standard cosine similarity.

Supervised Contrastive loss (SupCon): Ref. [32] extended the self-supervised NT-Xent loss to the fully supervised setting by adding multiple positives. The self-supervised NT-Xent loss pulls the anchor and the positive sample closer together and pushes the anchor away from many negative samples (other positive samples in batch), while the SupCon uses

additional information by adding multiple hard positives for each class. Hence, SupCon generalizes the NT-Xent loss to use multiple positive samples from the same class for each anchor. Therefore, the SupCon loss simply extends (x_i, x_i^+) to $(x_i, x_{i1}^+, \ldots, x_{iP}^+)$ and is defined as:

$$L_{SupCon} = -\frac{1}{P}\sum_{k=1}^{P} \log \frac{e^{sim(x_i, x_{ik}^+)/\tau}}{\sum_{j=1}^{N} e^{sim(x_i, x_j^+)/\tau}} \tag{5}$$

Multiple Negatives Ranking Loss (MNRL): This loss function is an implementation of [41] in SBERT documentation web page (https://www.sbert.net, accessed on 1 August 2022). It is a cross-entropy-based and in-batch negative loss function. In MNRL, the anchor-positive pair of (x_i, x_i^+) has been extended to $(x_i, x_i^+, x_{i1}^- \ldots, x_{iQ}^-)$ by adding multiple hard negatives. Therefore, for a triplet in the form of $(x_i, x_i^+, x_{i1}^- \ldots, x_{iQ}^-)$ in a mini-batch with N triplets, the loss is computed as:

$$L_{MNRL} = -\log \frac{e^{sim(x_i, x_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(x_i, x_j^+)/\tau} + \sum_{j=1}^{N}\sum_{k=1}^{Q} e^{sim(x_i, x_{jk}^-)/\tau}} \tag{6}$$

4. SupMPN: Supervised Multiple Positives and Negatives Contrastive Learning Model

We propose a Supervised Multiple Positives and Negatives Contrastive Learning model (SupMPN). Our contribution is to incorporate both multiple hard positives and multiple hard negatives simultaneously in contrastive learning. In SupMPN, we consider multiple triplets in the form of $(x_i, x_{i1}^+, \ldots, x_{iP}^+, x_{i1}^-, \ldots, x_{iQ}^-)$ where $(x_{i1}^+, \ldots, x_{iP}^+)$ and $(x_{i1}^-, \ldots, x_{iQ}^-)$ act as hard positives and hard negatives for anchor sentence x_i, respectively.

Therefore, SupMPN accepts multiple hard-positive sentences and multiple hard-negative sentences simultaneously and then tries to bring positives (similar sentences) closer to the anchor sentence, while pushing negatives (dissimilar sentences) away from the anchor sentence. In this way, SupMPN can solve the collapse problem of BERT-based sentence representation. Moreover, it can create more semantic-based embeddings. All these enable SupMPN to create a better sentence representation space.

4.1. Training Objective

Our contrastive objective function is an extension of MNRL [41]. We name it Supervised Multiple Positives and Negatives Ranking Loss. For convenience, we abbreviate the foregoing to SupMPNRL. Formally, we extend $(x_i, x_i^+, x_{i1}^- \ldots, x_{iQ}^-)$ to $(x_i, x_{i1}^+, \ldots, x_{iP}^+, x_{i1}^- \ldots, x_{iQ}^-)$, where x_i is an anchor sentence, $(x_{i1}^+, \ldots, x_{iP}^+)$ are hard positives and $(x_{i1}^- \ldots, x_{iQ}^-)$ are hard negatives for x_i in the input triplets within mini-batch with size N. Therefore, for the input triplet in the form of $(x_i, x_{i1}^+, \ldots, x_{iP}^+, x_{i1}^- \ldots, x_{iQ}^-)$, the loss is defined as:

$$\begin{aligned} L_{SupMPNRL} &= \frac{1}{P}\sum_{k=1}^{P} -\log \frac{e^{sim(x_i, x_{ik}^+)/\tau}}{e^{sim(x_i, x_{ik}^+)/\tau} + S_{positives} + S_{negatives}}, \\ S_{positives} &= \sum_{j=1}^{N}\sum_{k=1}^{P} \mathbb{1}_{j\neq i} e^{sim(x_i, x_{jk}^+)/\tau}, \\ S_{negatives} &= \sum_{j=1}^{N}\sum_{k=1}^{Q} e^{sim(x_i, x_{jk}^-)/\tau} \end{aligned} \tag{7}$$

where $sim(\cdot)$ is the standard cosine similarity, and $\mathbb{1}_{[j\neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 iff $j \neq i$.

4.2. Desirable Properties of SupMPN Model

SupMPN extends self-supervised contrastive learning to supervised contrastive learning by adding multiple hard positives and multiple hard negatives. We summarize the desirable properties of SupMPN which boost performance of contrastive learning in fine-tuning BERT, and consequently lead to better sentence representation space.

Using multiple hard positives: Hard positives are true positives which are farthest from the anchor sentence. Hard positives play an important role in contrastive learning and can improve performance greatly. Using multiple hard positives generalizes simple triplet loss to an arbitrary number of positives which enables the model to align similar sentences closely in the representation space [32].

Using multiple hard negatives: Hard negatives are similar (nearest) to the correct answer and are hard to differentiate from the positives. Therefore, using multiple hard negatives generalizes the triplet loss by allowing joint comparison among more hard-negative examples and enables the model to improve discrimination among positive and negative sentences [32,40].

No need for hard-positive mining and hard-negative mining: Unlike the BatchHard loss [39], due to using multiple hard positives and multiple hard negatives for each anchor, hard positive and hard-negative mining (hard positive and hard-negative selection) are no longer required in SupMPN.

Using a batch-wise approach (Online triplet mining): Using other sentences' positives and negatives in a batch as negatives (technique of in-batch negative) for each anchor, increases contrastive power with more negatives [32].

5. Experiments

5.1. Training Data

We use Natural Language Inference (NLI) datasets, SNLI [43] and MNLI [44], to train our model. NLI, which is also known as Recognizing Textual Entailment (RTE), is a task of determining whether a hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given a premise. SNLI (570 K) and MNLI (433 K) are collections of human-written English sentence pairs with the labels entailment, contradiction, and neutral. We use entailment hypotheses as positive data and contradiction hypotheses as negative data.

We group SNLI and MNLI according to the premise sentences (anchor sentences). There are about (∼280 K) premise sentences in SNLI and MNLI. Then, we analyze them in terms of the number of entailment and contradiction hypotheses for each premise sentence (anchor sentence). We find out that some premises in both datasets (SNLI and MNLI) have more than one entailment or contradiction hypotheses. Table 1 shows the statistics of SNLI and MNLI based on the number of entailments and contradictions for each premise sentence.

One of the most notable findings was that 7956 premises (∼8 K) in the SNLI dataset have at least five entailments and five contradictions. In fact, it can be said that for each of the premise sentences that has much information, more entailment and contradiction hypotheses have been provided in the SNLI and MNLI. Figure 1 shows an example of the premise sentence from SNLI with five entailment and five contradiction hypotheses. We call these 7956 multiple triplets (∼8 K) sub-SNLI and use it in the form of $(x_i, x_{i1}^+, \ldots, x_{i5}^+, x_{i1}^- \ldots, x_{i5}^-)$ as part of our training data. Therefore, our training inputs have one anchor, five hard positives, and five hard negatives in the form of $(x_i, x_{i1}^+, \ldots, x_{i5}^+, x_{i1}^- \ldots, x_{i5}^-)$ and we need to generate multiple positives and multiple negatives for the rest of training data (∼272 K) which have less than five positives or five negatives. We do not use STS training sets in our experiments.

Table 1. SNLI [43] and MNLI [44] statistics based on the number of entailment and contradiction hypotheses for each premise sentence.

	Number of premises with exactly one entailment and one contradiction	Number of premises with two, three, or four entailments and contradictions	Number of premises with five or more entailments and contradictions
SNLI	139,299	1890	7956
	Number of premises with exactly one entailment and one contradiction	Number of premises with exactly two entailments and two contradictions	Number of premises with exactly three entailments and three contradictions
MNLI	125,860	1783	489

Premise (as the anchor)	A woman is walking across the street eating a banana, while a man is following with his briefcase.
Entailments (as hard positives)	1. A woman eats a banana and walks across a street, and there is a man trailing behind her. 2. A person eating. 3. A woman eating a banana crosses a street. 4. The woman is eating a banana. 5. The woman is outside.
Contradictions (as hard negatives)	1. Nobody has food. 2. The woman and man are playing baseball together. 3. A woman eats ice cream walking down the sidewalk, and there is another woman in front of her with a purse. 4. A woman sits for lunch. 5. The woman is having coffee at the cafe.

Figure 1. Example of a premise sentence from SNLI [43] with multiple entailment and contradiction hypotheses.

5.2. Preparing Multiple Positives and Multiple Negatives

Contrastive learning usually exploits data-augmentation techniques to construct positive pairs, while negative examples are sampled from other classes' positives in mini batches [65]. In computer vision, data augmentation has been widely done based on image rotation, image sharpening, image corruption, and object deletion [66]. However, similar strategies may not work well in NLP, as changing the order of words in a sentence or deleting words from a sentence may substantially affect its semantic [33,67].

According to previous unsupervised works for fine-tuning BERT using contrastive learning, text augmentation techniques including word deletion and word synonym replacement have yielded relatively better results [16,33]. Therefore, we studied them for generating positive pairs. Additionally, for the first time, we studied paraphrasing for generating positive pairs using the T5 text-to-text model [68] on Semantic Textual Similarity. However, similar to the results of SimCSE [37], word deletion, word synonym replacement and paraphrasing did not perform well in all seven STS tasks and so did not produce a good average on STS tasks. We have provided more details of these three text augmentation techniques in Appendix A.1. Therefore, our main strategies for generating positive and negative samples are as follows:

Preparing multiple positives: No data augmentation is used. We simply copy each anchor several times as its positive samples.

Preparing multiple negatives: We sample negatives for each anchor from the entailments or contradictions of other anchors in the training data (SNLI and MNLI).

5.3. Training Setups

We use $\text{SimCSE}_{\text{sup}}$ (https://github.com/princeton-nlp/SimCSE, accessed on 1 August 2022) setup and start from pre-trained BERT-base (uncased) and BERT-large (uncased) models [14] hosted on the Hugging Face Model Hub (https://huggingface.co, accessed on

1 August 2022). We take average embeddings of the first and last layers (avg-first-last) as the pooling mode. We run SupMPN on NVIDIA A100 GPUs with CUDA 11 and train it for 3 epochs. We use a batch size of 256 for BERT-base and a batch size of 200 for BERT-large. We could not test more than batch size of 200 for BERT-large due to the lack of computation power.

We also tested different batch sizes of 64, 128, 256, 512 for BERT-base and observed that our model is sensitive to the batch size as the [32,38] have pointed out, contrastive learning requires a large batch, so that larger batches lead to learn better representations. However, there was no noticeable change in performance from the batch size of 256 onward on STS tasks, and even with batch size of 512, the performance dropped slightly on transfer-learning tasks. The underlying reason can be stated in terms of other data points in the batch serving as the negatives for the given data point and also our model accepting multiple negatives for each anchor sentence (in our implementation, five negatives for each anchor sentence), meaning that increasing the batch size leads to lower performance, because more negative samples does not necessarily mean hard-negative samples [69]. On the other hand, enormous number of negatives can lead to worsening signal-to-noise ratio for the model gradients which could explain the decline in performance [70].

5.4. Baseline and Previous Supervised and Unsupervised Models for Comparison

In our experiments, we compare our proposed SupMPN model to the previous state-of-the-art SimCSE and other supervised and unsupervised sentence encoder models. We list them separately as supervised and unsupervised models in Table 2. It is worthwhile emphasizing that our model (SupMPN) and state-of-the-art SimCSE are trained on the same data (entailments and contradictions from SNLI and MNLI) and other models are trained on various datasets with different sizes. We compared them in terms of type and size of training data in Appendix A.2.

The non-BERT models are GloVe [46], InferSent [49], and USE [51] which we briefly explained in Section 2. We consider SBERT [52] as a baseline model. We introduce other BERT-based models briefly in the following:

BERT-flow: Ref. [15] claim that sentence embeddings generated by BERT depend on the frequency of words that lead to the anisotropy of sentence representations. Therefore, to solve this problem, they feed embedding vectors to a flow network [71] and try fitting them to a standard Gaussian distribution.

BERT-whitening: Ref. [72] proposed this model which, as with BERT-flow [15], tries to solve the problem of the anisotropy of the sentence representations of BERT using the whitening operation in machine learning. In addition, by applying the whitening operation, the dimensions of the sentence representation and the storage cost are reduced. Consequently, the model retrieval speed is accelerated.

CLEAR: Ref. [33] proposed this model for fine-tuning BERT using combined loss as a combination of MLM (mask language model) and CL (contrastive loss) objectives, which rely on text augmentation techniques.

CT-BERT: Ref. [57] proposed this model which uses two independent encoders and then tries to maximize/minimize the dot product between two identical/different sentences.

ConSERT: Ref. [16] proposed this model to solve the collapse problem of BERT by generating different form of input samples using data augmentation and a contrastive objective on top of BERT-encoder.

DeCLUTR: Ref. [35] proposed this model which learns via different spans from the same document as positive samples using contrastive learning.

IS-BERT: Ref. [56] proposed this model which uses a contrastive objective based on mutual information maximization mechanism between the global and local sentence representation.

Mirror-BERT: Ref. [36] proposed this model which uses self-supervised contrastive learning based on random span masking as data augmentation for the input space.

SBERT-base-nli-v2: This checkpoint is a pre-trained model from the Sentence-Transformer packages (https://huggingface.co/kwang2049/SBERT-base-nli-v2, accessed on 1 August 2022). SBERT-base-nli-v2 was trained on SNLI and MNLI data using the Multiple-Negative Ranking Loss (MNRL) in [73].

SG-BERT: Ref. [34] proposed this model which is a self-supervised contrastive learning method using the redesign of NT-Xent objective with self-guidance. They exploit similarities between different sentence embeddings made by BERT itself.

SimCSE: Ref. [37] proposed $\text{SimCSE}_{\text{unsup}}$ and $\text{SimCSE}_{\text{sup}}$ models. $\text{SimCSE}_{\text{unsup}}$ is a self-supervised contrastive learning that takes an input sentence and predicts itself using the dropout noise. $\text{SimCSE}_{\text{sup}}$ uses entailment and contradiction pairs from NLI datasets and extends self-supervised to supervised contrastive learning. Additionally, they apply an auxiliary Masked Language Modeling (MLM) objective to its models and stated that adding MLM boosts performance on transfer-learning tasks (not on STS tasks).

TSDAE: Ref. [73] proposed this model, which is an unsupervised method based on pre-trained transformers, and sequential denoising autoencoder. In the training phase, TSDAE uses an autoencoder that encodes corrupted sentences into fixed-sized vectors (encoder) and then reconstructs the original sentences from this sentence embedding (decoder). Later, at the inference phase, TSDAE only uses the encoder for creating sentence embeddings.

5.5. First Experiment: Evaluation on STS Tasks

We evaluate SupMPN on seven standard Semantic Textual Similarity (STS) tasks: STS 2012–2016 [74–78], STS Benchmark [79] and SICK-Relatedness [80]. These datasets provide gold labels between 0 and 5 on the semantic relatedness of sentence pairs. The Spearman's rank correlation between the cosine-similarity of the sentence embeddings and the gold labels are computed.

For evaluation on the STS tasks, we use a modified version of the SentEval toolkit by [37]. SentEval [81] is a popular library for evaluating the quality of sentence embeddings (https://github.com/facebookresearch/SentEval, accessed on 1 August 2022). It covers various tasks including binary and multi-class classification, Natural Language Inference and sentence similarity. Ref. [37] reported three differences in STS evaluation settings done by the SentEval toolkit. These differences are using additional regressors for training frozen sentence embedding on the STS-B [79] and SICK-R datasets [80], reported metrics that can be Spearman's or Pearson's correlation coefficient, and method of result aggregations. As has been argued in [37,52], the Spearman's correlation, which measures the ranking instead of the actual scores, it is appropriate for evaluation of sentence embeddings. We use Spearman's rank correlation in our experiments. For several STS subsets each year, there are three different options for collecting results: all, mean, and wmean. In the "all" settings, all the topics are concatenated and then the overall Spearman's correlation is reported. In "mean" setting, results for the different subsets are calculated separately and then their simple average is reported. In the "wmean" setting, which is similar to the "mean" setting, the weighted average with subset sizes is reported.

Since most papers do not state the method they take, ref. [37] reproduced the results of SBERT [52], BERT-flow [15] and BERT-whitening [72] and compared them to their original results. They reported two different settings, "all" and "wmean". Ref. [52] took the "all" setting, and [15,72] took the "wmean" settings. Since SBERT usually is considered as the baseline in the literature, ref. [37] took the "all" setting and made two changes on the original SentEval: (1) added the "all" setting to all STS tasks, and (2) changed STS-B and SICK-R to not use an additional regressor. Ref. [37] requested researchers unify the so-called settings in evaluating sentence embeddings in their future works and released modification of SentEval in their GitHub repository (https://github.com/princeton-nlp/SimCSE, accessed on 1 August 2022). Hence, we take their modified SentEval and use it in our first evaluation.

Results: As shown in Table 2, the SupMPN model outperforms the state-of-the-art SimCSE model and all other previous unsupervised and supervised models on STS tasks. SupMPN achieves averages of 82.07% and 83.15% Spearman's correlation on STS tasks using BERT-base and BERT-large, respectively. The corresponding improvements are 7.18 points for $\text{SupMPN}_{\text{base}}$ and 8.26 points for $\text{SupMPN}_{\text{large}}$ on STS tasks, compared to baseline SBERT.

Table 2. Experimental results on STS tasks. We report Spearman's rank correlation as $\rho \times 100$. The best result in each column is in bold. †: [52], ‡: [35], ♠: [56], ♣: [36], ★: [73], ♦: [16], ♡: [33], ♢: [34], and all the other results are from [37].

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
$\text{BERT}_{\text{base}}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$\text{BERT}_{\text{base}}$-flow-NLI	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$\text{BERT}_{\text{base}}$-whitening-NLI	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-$\text{BERT}_{\text{base}}$ ♠	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-$\text{BERT}_{\text{base}}$	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
SG-$\text{BERT}_{\text{base}}$ ♢	66.84	80.13	71.23	81.56	77.17	77.23	68.16	74.62
Mirror-$\text{BERT}_{\text{base}}$ ♣	69.10	81.10	73.00	81.90	75.70	78.00	69.10	75.40
$\text{SimCSE}_{\text{unsup}}$-$\text{BERT}_{\text{base}}$	68.40	82.41	**80.91**	78.56	78.56	76.85	72.23	76.25
TSDAE-$\text{BERT}_{\text{base}}$ ★	55.02	67.40	62.40	74.30	73.00	66.00	62.30	65.80
ConSERT-$\text{BERT}_{\text{base}}$ ♦	70.53	79.96	74.85	81.45	76.72	78.82	77.53	77.12
ConSERT-$\text{BERT}_{\text{large}}$ ♦	73.26	82.37	77.73	83.84	78.75	81.54	78.64	79.44
$\text{RoBERTa}_{\text{base}}$ (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
CLEAR-$\text{RoBERTa}_{\text{base}}$ ♡	49.00	48.90	57.40	63.60	65.60	72.50	75.60	61.08
DeCLUTR-$\text{RoBERTa}_{\text{base}}$ ‡	52.41	75.19	65.52	77.12	78.63	72.41	68.62	69.99
Supervised models								
InferSent-GloVe †	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder †	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
$\text{SBERT}_{\text{base}}$ †	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
$\text{SBERT}_{\text{base}}$-nli-v2 ★	72.50	84.80	80.20	84.80	80.00	83.90	78.00	80.60
$\text{SBERT}_{\text{base}}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
$\text{SBERT}_{\text{base}}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
CT-$\text{SBERT}_{\text{base}}$	74.84	83.20	78.07	83.84	77.93	81.46	76.42	79.39
SG-$\text{BERT}_{\text{base}}$ ♢	75.16	81.27	76.31	84.71	80.33	81.46	76.64	79.41
$\text{SimCSE}_{\text{sup}}$-$\text{BERT}_{\text{base}}$	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
SupMPN-$\text{BERT}_{\text{base}}$	**75.96**	**84.96**	80.61	**85.63**	**81.69**	**84.90**	**80.72**	**82.07**
$\text{SimCSE}_{\text{sup}}$-$\text{BERT}_{\text{large}}$	75.78	86.33	80.44	**86.06**	80.86	84.87	81.14	82.21
SupMPN-$\text{BERT}_{\text{large}}$	**77.53**	**86.50**	**81.68**	85.99	**82.87**	**86.09**	**81.38**	**83.15**

5.6. Second Experiment: Evaluation on Transfer-Learning Tasks

We evaluate the SupMPN model on the seven SentEval transfer-learning tasks using the default configuration of the SentEval toolkit [81]. Sentence embeddings are used as features for a logistic regression classifier. The logistic regression classifier is trained on various tasks in a 10-fold cross-validation setting and the prediction accuracy is computed for the test fold. We evaluate SupMPN on the following seven SentEval transfer tasks: MR [82], CR [83], SUBJ [84], MPQA [85] , SST-2 [86], TREC [87], and MRPC [88]. We compare SupMPN to some of the previous models mentioned in Section 5.4, for which the evaluation on the transfer-learning tasks were reported in the literature. The results are given in Table 3.

Results: As shown in Table 3, SupMPN outperforms state-of-the-art SimCSE and all other previous supervised and unsupervised models on transfer-learning tasks. SupMPN achieves averages of 86.96% and 87.75% accuracy on transfer-learning tasks using BERT-base and BERT-large, respectively. The corresponding improvements are 0.34 for $\text{SupMPN}_{\text{large}}$ on transfer-learning tasks, compared to baseline SBERT.

Table 3. Experimental results on the transfer-learning tasks. We report prediction accuracy. The best result in each column is in bold. †: [52], ♠: [56], ♢: [34], ∞: [57], and all the other results from [37].

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
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
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
SimCSE$_{\text{sup}}$-BERT$_{\text{base}}$-MLM	82.68	88.88	94.52	89.82	88.41	87.60	76.12	86.86
SupMPN-BERT$_{\text{base}}$	82.93	89.26	94.76	90.21	86.99	88.20	**76.35**	*86.96*
SupMPN-BERT$_{\text{large}}$	**84.06**	**90.25**	94.59	90.26	88.58	91.00	75.48	**87.75**

5.7. Third Experiment: Textual Semantic Similarity in Representation Space

In this experiment, we compare sentence similarity scores using embeddings from our model SupMPN, BERT [14], and SimCSE$_{\text{sup}}$ [37]. For BERT and SimCSE$_{\text{sup}}$, we use "bert-base-uncased" (https://huggingface.co/bert-base-uncased, accessed on 1 August 2022) and "princeton-nlp/sup-simcse-bert-base-uncased" (https://huggingface.co/princeton-nlp/sup-simcse-bert-base-uncased, accessed on 1 August 2022) models, respectively.

In this experiment, we consider 12 sentences in four different topics (three sentences in each topic). The topics are weather, smartphone, sport, and food. Since we want to compare semantic scoring of these models, we select the sentences of each topic such that two of them have positive meanings and one of them has a negative meaning (anchor-positive-negative). In addition, in the first two topics (weather and sport), the positive or negative meanings of the sentences are completely clear. But in the second and third topics (smartphone and food), positive sentences are selected in such a way that they are difficult to infer by the models compared to negative sentences. We use cosine similarity for scoring. The results are depicted in Figure 2.

Results: As shown in Figure 2a, the BERT model could not produce proper vector space for sentences with different topics and almost all pairs of sentences have similarity scores in the range of 0.6 to 1.0. SimCSE$_{\text{sup}}$ model, Figure 2b, despite making a good distinction among different topics, could not embed properly positive and negative sentences in two topics, smartphone and food. For example, in SimCSE$_{\text{sup}}$, for triplet sentences as (anchor: "The women are eating seafood", positive: "The women are having lunch", negative: "The women are not eating seafood"), the similarity between the anchor-negative pair is more than the similarity between the anchor-positive pair. In accordance with Figure 2c, SupMPN can embed sentences properly in all four topics and, unlike SimCSE$_{\text{sup}}$, it can correctly infer positive and negative sentences in smartphone and food topics and distinguish between them, even if they have common words or not. Therefore, SupMPN can create better semantic-based sentence representation space and can solve the collapse problem of BERT.

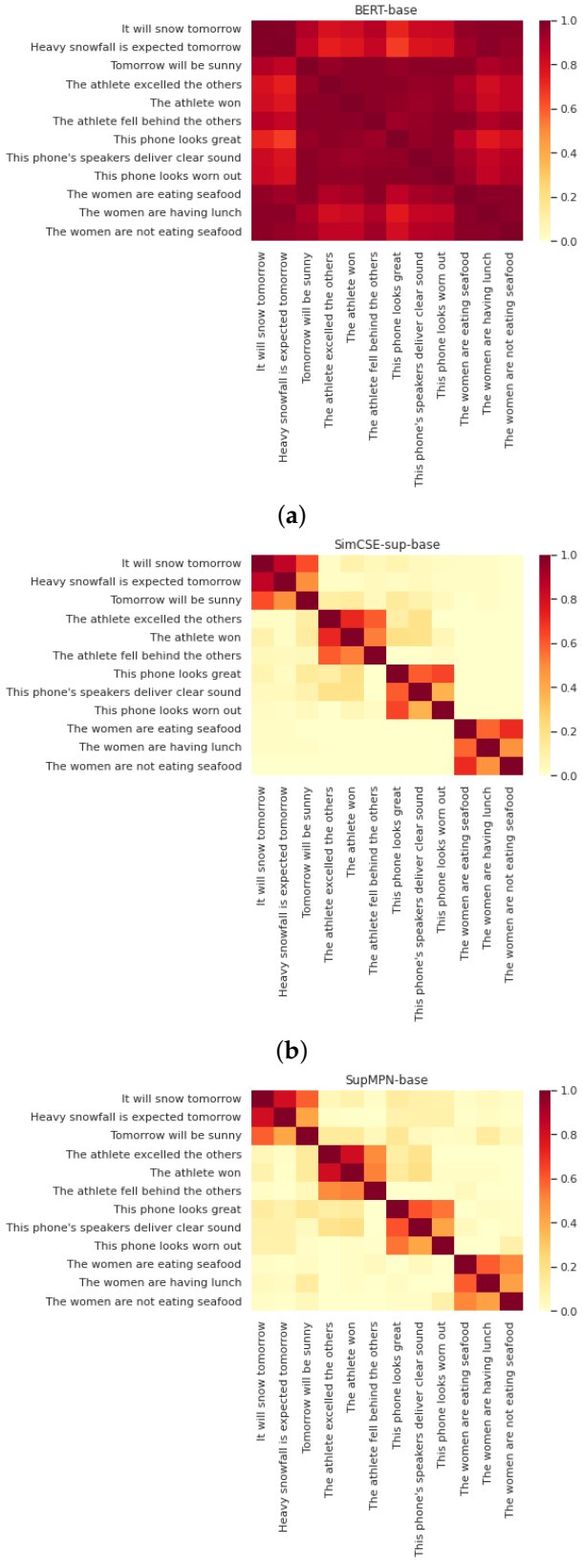

Figure 2. Semantic textual similarity using sentence embeddings from (**a**) BERT, (**b**) $\text{SimCSE}_{\text{sup}}$, and (**c**) SupMPN, starting from BERT-base checkpoint, on four topics, weather, sport, smartphone, and food. We used cosine similarity for scoring.

6. Conclusions

Since the BERT pre-trained language model has a collapse problem, the BERT-generated sentence embeddings mapped into a small area and almost all sentence pairs have a cosine similarity in the range of 0.6 and 1.0. This paper proposed an effective contrastive learning model, called SupMPN, to deal with this problem. We developed a new contrastive objective function to fine-tune BERT that involves the multiple hard positives and multiple hard negatives in contrasting learning simultaneously. Using multiple hard positives, SupMPN can bring an arbitrary number of similar sentences closer in the representation space, while, using multiple hard negatives, it is able to improve discrimination among positives and negatives sentences. SupMPN can solve the collapse problem of BERT. Accordingly, the sentence embeddings generated by SupMPN can capture the underlying semantics of sentences. As a result, SupMPN can generate sentence embeddings based on the semantic meaning instead of the frequency of the words. The experiments on STS and transfer-learning tasks demonstrate that SupMPN significantly outperforms baseline, state-of-the-art SimCSE and all other previous unsupervised and supervised models. Our model obtains 7.18 and 8.26 performance improvements in terms of average Spearman's rank correlation on the seven STS tasks compared to the baseline model starting from BERT-base and BERT-large, respectively. Our model obtains 0.34 performance improvement in terms of average accuracy on the seven standard transfer-learning tasks compared to the baseline model start from BERT-large. In summary, our approach significantly improves the sentence representation space. The only shortcoming and limitation of our model is that it requires high computational power.

In the future, we will investigate the curriculum-learning strategy for Semantic Textual Similarity whose mechanism is based on training a machine-learning model from easier to harder samples, inspired by human meaningful learning order from easiest concepts to most complex ones.

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 research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Our code, pre-trained models and training data are publicly available at: https://github.com/SoDehghan/SupMPN (accessed on 18 August 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
ELMo	Embeddings from Language Model
MLM	Mask Language Model
MNRL	Multiple Negatives Ranking Loss
NLI	Natural Language Inference
NLP	Natural Language Processing
NSP	Next Sentence Prediction
NT-Xent	Normalized Temperature-scale Cross-Entropy
RoBERTa	Robustly Optimized BERT Pretraining Approach
SBERT	Sentence-BERT

STS	Semantic Textual Similarity
SupCon	Supervised Contrastive
USE	Universal Sentence Encoder

Appendix A

Appendix A.1. Result of Using Text Augmentation to Prepare Positive Samples

We study text augmentation techniques for generating positive samples for each anchor sentence. As mentioned in Section 5.2, we use word deletion, word synonym replacement, and paraphrasing. We consider four options including none (No data augmentation is used), word deletion, synonym replacement using WordNet, and paraphrasing using Text-to-Text Transfer Transformer (T5) generative language model [68]. We briefly explain these four option settings in the following:

- **None:** No data augmentation is used. We simply copy each anchor sentence several times as its positive samples (our implementation).
- **Random Word Deletion (RD):** We randomly delete 10% of each entailment's words.
- **Synonym Replacement (SR):** We randomly substitute 20% of each entailment's words with their synonyms using WordNet [89].
- **Paraphrasing (PP):** We paraphrase almost 50% of entailments.

We use NLPAug (https://github.com/makcedward/nlpaug, accessed on 1 August 2022) library [90] for word deletion and synonym replacement. NLPAug is a straightforward data-augmentation library which implements 15 methods for text data augmentation based on word, sentence, and character. Additionally, we use Parrot (https://github.com/PrithivirajDamodaran/Parrot_Paraphraser, accessed on 1 August 2022) paraphraser library [91] for paraphrasing. Parrot is a T5 model fine-tuned on some of paraphrase datasets such as MRPC Paraphrase (https://www.microsoft.com/en-us/download/details.aspx?id=52398, accessed on 1 August 2022) [88], Google PAWS (https://github.com/google-research-datasets/paws, accessed on 1 August 2022) [92], ParaNMT (https://github.com/jwieting/para-nmt-50m#readme, accessed on 1 August 2022) [93], Quora Question Pairs (https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs, accessed on 1 August 2022), SNIPS Commands (https://github.com/sonos/nlu-benchmark/tree/master/2017-06-custom-intent-engines, accessed on 1 August 2022) [94].

As shown in Table A1, none of these augmentation produces acceptable average on all STS tasks and so, cannot outperform the None option (SupMPN-BERT$_{base}$-None). SupMPN-BERT$_{base}$-SR only works well on STS12 and STS15, SupMPN-BERT$_{base}$-RD only works well on STS16, and SupMPN-BERT$_{base}$-PP only works well on SICK-R. We think the underlying reason depends on the nature of STS as each task includes several sub-tasks from various sources with different properties.

Table A1. Result of applying various text augmentation techniques for creating positive pairs on STS tasks. In each column, values greater than our implementation (None option) is in bold.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
SupMPN-BERT$_{base}$-None	75.96	**84.96**	**80.61**	85.63	81.69	**84.90**	80.72	**82.07**
SupMPN-BERT$_{base}$-RD	75.82	84.80	80.37	85.76	**81.98**	84.36	80.08	81.88
SupMPN-BERT$_{base}$-SR	**76.75**	83.82	80.34	**86.04**	80.67	83.88	80.09	81.66
SupMPN-BERT$_{base}$-PP	76.10	84.66	79.81	84.53	81.65	84.02	**81.18**	81.71

STS is related to both textual entailment (TE) and paraphrasing, but it differs in several ways. In STS, graded equivalence between a pair of texts is bidirectional but in TE the equivalence is directional, e.g., a car is a vehicle, but a vehicle is not necessarily a car. Additionally, both TE and paraphrasing are binary yes/no decision (e.g., a vehicle is not a car), but STS is defined as a graded-similarity notion (e.g., a vehicle and a car are more similar than a wave and a car [74–78]. Therefore, the sub-tasks in each STS tasks (STS12, STS13, STS14, STS15, STS16) are affected by augmentation techniques.

For example, STS12 contains sentences from previously existing paraphrase datasets so, synonym replacement (SR) works well in it. STS16 includes plagiarism detection sub-task, therefore, randomly word deletion (RD) works well in it as part of the original sentence can be copied in augmented sentence. SICK-R is constructed for both entailment and relatedness. Therefore, paraphrasing (pp) works well in SICK-R as paraphrasing (PP) through T5 model can preserve the overall semantic meaning and entailment inferencing of the original sentence.

Appendix A.2. Statistics

Training data and their sizes in SupMPN model and other supervised and unsupervised models are compared in Table A2.

Table A2. Training data and their sizes in SupMPN and other models.

Model	Training Data	Size
BERT	Book Corpus + English Wikipedia	Not Specified
BERT-flow	SNLI + MNLI	570 K + 433 K
BERT-mirror	Training set of the STS Benchmark (for STS tasks)	10 K
BERT-whitening	SNLI + MNLI	570 K + 433 K
CLEAR	Book Corpus + English Wikipedia	Not Specified
CT-BERT	English Wikipedia	Not Specified
ConSERT	SNLI + MNLI	570 K + 433 K
DeCLUTR	Open Web Text corpus	497 K
InferSent	SNLI + MNLI	570 K + 433 K
IS-BERT	SNLI + MNLI	570 K + 433 K
SBERT	SNLI + MNLI	570 K + 433 K
SBERT-base-nli-v2	Part of (SNLI + MNLI)	Not specified
SG-BERT	Part of (SNLI + MNLI)	Not Specified
$SimCSE_{sup}$	Part of (SNLI + MNLI)	628 K
SupMPN	Part of (SNLI + MNLI)	628 K
TSDAE	English Wikipedia	Not Specified
USE	Web sources + Question answering + SNLI	Not Specified

References

1. Hliaoutakis, A.; Varelas, G.; Voutsakis, E.; Petrakis, E.G.M.; Milios, E. Information Retrieval by Semantic Similarity, 2006. Available online: https://www.researchgate.net/publication/283921249_Information_retrieval_by_semantic_similarity (accessed on 1 August 2022).
2. Kim, S.; Fiorini, N.; Wilbur, W.J.; Lu, Z. Bridging the gap: Incorporating a semantic similarity measure for effectively mapping PubMed queries to documents. *J. Biomed. Inform.* **2017**, *75*, 122–127. [CrossRef] [PubMed]
3. Mohamed, M.; Oussalah, M. SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Inf. Process. Manag.* **2019**, *56*, 1356–1372. [CrossRef]
4. Hou, Y.-B. A Text Summarization Method Based on Semantic Similarity among Sentences. *DEStech Trans. Social Sci. Educ. Human Sci.* **2020**. [CrossRef]
5. Mukherjee, I.; Mahanti, P.K.; Bhattacharya, V.; Banerjee, S. Text classification using document-document semantic similarity. *Int. J. Web Sci.* **2013**, *2*, 1–26. [CrossRef]
6. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751. [CrossRef]
7. Malandrakis, N.; Falcone, M.; Vaz, C.; Bisogni, J.; Potamianos, A.; Narayanan, S. SAIL: Sentiment Analysis using Semantic Similarity and Contrast Features. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 512–516. [CrossRef]
8. Janda, H.K.; Pawar, A.; Du, S.; Mago, V. Syntactic, Semantic and Sentiment Analysis: The Joint Effect on Automated Essay Evaluation. *IEEE Access* **2019**, *7*, 108486–108503. [CrossRef]
9. Bordes, A.; Chopra, S.; Weston, J. Question Answering with Subgraph Embeddings. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 615–620. [CrossRef]
10. Lopez-Gazpio, I.; Maritxalar, M.; Gonzalez-Agirre, A.; Rigau, G.; Uria, L.; Agirre, E. Interpretable semantic textual similarity: Finding and explaining differences between sentences. *Knowl.-Based Syst.* **2017**, *119*, 186–199. [CrossRef]
11. Castillo, J.; Estrella, P. Semantic Textual Similarity for MT evaluation. In Proceedings of the Seventh Workshop on Statistical Machine Translation, Montreal, QC, Canada, 7–8 June 2012. Available online: https://aclanthology.org/W12-3103 (accessed on 1 August 2022).

12. Zou, W.Y.; Socher, R.; Cer, D.; Manning, C.D. Bilingual word embeddings for phrasebased machine translation. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1393–1398. Available online: https://aclanthology.org/D13-1141 (accessed on 1 August 2022).
13. Liu, S.; He, T.; Li, J.; Li, Y.; Kumar, A. An Effective Learning Evaluation Method Based on Text Data with Real-time Attribution—A Case Study for Mathematical Class with Students of Junior Middle School in China. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**. [CrossRef]
14. 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]
15. 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]
16. 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 (Volume 1), Virtual Event, 1–6 August 2021. [CrossRef]
17. Chen, X.; He, K. Exploring Simple Siamese Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021. [CrossRef]
18. Jalal, A.; Kim, Y.-H.; Kim, Y.-J.; Kamal, S.; Kim, D. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recognit.* **2017**, *61*, 295–308. [CrossRef]
19. Jalal, A.; Quaid, M.A.K.; Sidduqi, M.A. A Triaxial Acceleration-based Human Motion Detection for Ambient Smart Home System. In Proceedings of the 2019 IEEE 16th International Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 8–12 January 2019; pp. 353–358. [CrossRef]
20. Wu, H.; Pan, W.; Xiong, X.; Xu, S. Human activity recognition based on the combined SVM&HMM. In Proceedings of the 2014 IEEE International Conference on Information and Automation (ICIA), Hailar, China, 28–30 July 2014; pp. 219–224. [CrossRef]
21. Piyathilaka, L.; Kodagoda, S. Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features. In Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), Melbourne, Australia, 19–21 June 2013. [CrossRef]
22. Uddin, M.T.; Uddin, A. Human activity recognition from wearable sensors using extremely randomized trees. In Proceedings of the International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Savar, Bangladesh, 21–23 May 2015. [CrossRef]
23. Tang, C.I.; Perez-Pozuelo, I.; Spathis, D.; Mascolo, C. Exploring Contrastive Learning in Human Activity Recognition for Healthcare. Presented at the Machine Learning for Mobile Health Workshop at NeurIPS 2020, Vancouver, BC, Canada, 2020. *arXiv* **2020**, arXiv:2011.11542. [CrossRef]
24. Huang, Q.; Yang, J.; Qiao, Y. Person re-identification across multi-camera system based on local descriptors. In Proceedings of the IEEE Conference on Distributed Smart Cameras, Hong Kong, China, 30 October–2 November 2012; pp. 1–6. [CrossRef]
25. Khaldi, K.; Shah, S.K. CUPR: Contrastive Unsupervised Learning for Person Re-identification. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021)—Volume 5: VISAPP, Online, 8–10 February 2021; pp. 92–100. [CrossRef]
26. Chen, I.-K.; Chi, C.-Y.; Hsu, S.-L.; Chen L.-G. A real-time system for object detection and location reminding with RGB-D camera. In Proceedings of the 2014 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 10–13 January 2014. [CrossRef]
27. Xie, E.; Ding, J.; Wang, W.; Zhan, X.; Xu, H.; Sun, P.; Li, Z.; Luo, P. DetCo: Unsupervised Contrastive Learning for Object Detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 8372–8381. [CrossRef]
28. Ahad, M.A.R.; Kobashi, S.; Tavares, J.M.R.S. Advancements of image processing and vision in healthcare. *J. Healthc. Eng.* **2018**, *2018*, 8458024. [CrossRef] [PubMed]
29. Zhang, Z.; Jang, J.; Trabelsi, C.; Li, R.; Sanner, S.; Jeong, Y.; Shim, D. ExCon: Explanation-driven Supervised Contrastive Learning for Image Classification. *arXiv* **2021**, arXiv:2111.14271.
30. Rathore, M.M.U.; Ahmad, A.; Paul, A.; Wu, J. Real-time continuous feature extraction in large size satellite images. *J. Syst. Archit. EUROMICRO* **2016**, *64*, 122–132. [CrossRef]
31. Madhusudana, P.C.; Birkbeck, N.; Wang, Y.; Adsumilli, B.; Bovik, A.C. Image Quality Assessment using Contrastive Learning. *IEEE Trans. Image Process.* **2022**, *31*, 4149–4161. [CrossRef] [PubMed]
32. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised Contrastive Learning. *arXiv* **2021**, arXiv:2004.11362.
33. Wu, Z.; Sinong, S.; Gu, J.; Khabsa, M.; Sun, F.; Ma, H. CLEAR: Contrastive Learning for Sentence Representation. *arXiv* **2020**, arXiv:2012.15466.
34. 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 Processin, Virtual Event, 1–6 August 2021; Volume 1. [CrossRef]

35. 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]
36. 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]
37. 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]
38. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv* **2020**, arXiv:2002.05709.
39. Hermans A.; Beyer, L.; Leibe, B. In defence of triplet loss for person re-identification. *arXiv* **2017**, arXiv:1703.07737.
40. 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 1 August 2022).
41. 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.
42. Liu, S.; Xu, X.; Zhang, Y.; Muhammad, K.; Fu, W. A Reliable Sample Selection Strategy for Weakly-supervised Visual Tracking. *IEEE Trans. Reliab.* **2022**, 1–12. [CrossRef]
43. 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]
44. 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]
45. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
46. 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]
47. 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]
48. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with sub word information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
49. 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.
50. 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]
51. 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]
52. 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]
53. Thakur, N.; Reimers, N.; Daxenberger, J.; Gurevych, I. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021. [CrossRef]
54. Wang, B.; Kuo, C.-C.J. SBERT-WK: A Sentence Embedding Method by Dissecting BERT-based Word Models. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2146–2157. [CrossRef]
55. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
56. Zhang, Y.; He, R.; Liu, Z.; Lim, K.H.; Bing, L. An unsupervised sentence embedding method by mutual information maximization. *arXiv* **2020**, arXiv:2009.12061.
57. 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 1 August 2022).
58. 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]

59. 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]
60. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. *arXiv* **2015**, arXiv:1503.03832.
61. 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]
62. Gao, L.; Zhang, Y.; Han, J.; Callan, J. Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup. *arXiv* **2021**, arXiv:2101.06983.
63. 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.
64. Rosasco, L.; Vito, E.D.; Caponnetto, A.; Piana, M.; Verri, A. Are loss functions all the same? *Neural Comput.* **2004**, *16*, 1063–1076. [CrossRef] [PubMed]
65. Shorten, C.; Khoshgoftaar, M.T.; Furht, B. Text Data Augmentation for Deep Learning. *J. Big Data* **2021**, *8*, 101. [CrossRef] [PubMed]
66. Shorten, C.; Khoshgoftaar, M.T. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
67. Wu, X.; Gao, C.; Zang, L.; Han, J.; Wang, Z.; Hu, S. ESimCSE: Enhanced Sample Building Method for Contrastive Learning of Unsupervised Sentence Embedding. *arXiv* **2021**, arXiv:2109.04380.
68. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, A.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
69. Kalantidis, Y.; Sariyildiz, M.B.; Pion, N.; Weinzaepfel, P.; Larlus, D. Hard Negative Mixing for Contrastive Learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21798–21809.
70. Mitrovic, J.; McWilliams, B.; Rey, M. Less Can Be More in Contrastive Learning. In *ICBINB@NeurIPS*; 2020; pp. 70–75. Available online: https://openreview.net/pdf?id=U2exBrf_SJh (accessed on 1 August 2022).
71. Kingma D.P.; Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS, Montréal, QC, Canada, 3–8 December 2018; pp. 10236–10245. [CrossRef]
72. Su, J.; Cao, J.; Liu, W.; Ou, Y. Whitening sentence representations for better semantics and faster retrieval. *arXiv* **2021**, arXiv:2103.15316.
73. 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]
74. 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 1 August 2022).
75. 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 1 August 2022).
76. 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 1 August 2022).
77. 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]
78. 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]
79. 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]
80. 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 1 August 2022).

81. 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 1 August 2022).
82. 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]
83. 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; ACM: New York, NY, USA, 2004; pp. 168–177. Available online: https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf (accessed on 1 August 2022).
84. 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 1 August 2022).
85. Wiebe, J.; Wilson, T.; Cardie, C. Annotating Expressions of Opinions and Emotions in Language. *Lang. Resour. Eval.* **2005**, *39*, 165–210. [CrossRef]
86. 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 1 August 2022).
87. 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; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 1–7. Available online: https://aclanthology.org/C02-1150/ (accessed on 1 August 2022).
88. 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; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004. Available online: https://aclanthology.org/C04-1051 (accessed on 1 August 2022).
89. Miller, G.A.; Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, K.j. WordNet: An online lexical database. *Int. J. Lexicogr.* **1990**, *3*, 235–244. [CrossRef]
90. Ma, E. NLP Augmentation. 2019. Available online: https://github.com/makcedward/nlpaug (accessed on 1 August 2022).
91. Damodaran, P. Parrot: Paraphrase Generation for NLU. v1.0. 2021. Available online: https://github.com/PrithivirajDamodaran/Parrot_Paraphraser (accessed on 1 August 2022).
92. Zhang, Y.; Baldridge, J.; He, L. PAWS: Paraphrase Adversaries from Word Scrambling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1. [CrossRef]
93. Wieting, J.; Gimpel, K. ParaNMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1. [CrossRef]
94. Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; et al. Snips Voice Platform: An embedded Spoken Language Understanding system for private-by-design voice interfaces. *arXiv* **2018**, arXiv:1805.10190.