SDA: Simple Discrete Augmentation for Contrastive Sentence Representation Learning

Dongsheng Zhu[1,3,*]**, Zhenyu Mao**[2,*]**, Jinghui Lu**[2,*]**, Rui Zhao**[2]**, Fei Tan**[2,†]

[1] Baidu Inc. [2] SenseTime Research
[3] Fudan University

dszhu20@fudan.edu.cn
{maozhenyu, lujinghui1, zhaorui, tanfei}@sensetime.com

Abstract

Contrastive learning has recently achieved compelling performance in unsupervised sentence representation. As an essential element, data augmentation protocols, however, have not been well explored. The pioneering work SimCSE resorting to a simple dropout mechanism (viewed as **continuous** augmentation) surprisingly dominates discrete augmentations such as cropping, word deletion, and synonym replacement as reported. To understand the underlying rationales, we revisit existing approaches and attempt to hypothesize the desiderata of reasonable data augmentation methods: balance of semantic consistency and expression diversity. We then develop three simple yet effective **discrete** sentence augmentation schemes: **punctuation insertion**, **modal verbs**, and **double negation**. They act as minimal noises at lexical level to produce diverse forms of sentences. Furthermore, **standard negation** is capitalized on to generate negative samples for alleviating feature suppression involved in contrastive learning. We experimented extensively with semantic textual similarity on diverse datasets. The results support the superiority of the proposed methods consistently. Our key code is available at https://github.com/Zhudongsheng75/SDA

Keywords: Neural language representation models; Parsing, Grammar, Syntax, Treebank; Semantics; Semi-supervised, weakly-supervised and unsupervised learning

1. Introduction

Current state-of-the-art methods utilize contrastive learning algorithms to learn unsupervised sentence representations (Gao et al., 2021; Yan et al., 2021). They learn to bring similar sentences closer in the latent space while pushing away dissimilar ones (Hjelm et al., 2018). In the paradigm, multiple data augmentation methods have been proposed from different perspectives to curate different variants[1] (Oord et al., 2018; Zhu et al., 2020). The variants and the corresponding original samples are deemed positive pairs in the learning procedure (Chen et al., 2020). Previous studies have shown that the quality of variants largely shapes the learning of reasonable representations (Chen et al., 2020; Hassani and Khasahmadi, 2020).

Conventional methods directly employ operations such as cropping, word deletion, and synonym replacement in natural sentences (Wei and Zou, 2019; Wu et al., 2020; Meng et al., 2021). In addition, recent studies resort to network architectures for manipulating embedding vectors, such as dropout, feature cutoff, and token shuffling (Gao et al., 2021; Yan et al., 2021). It enables more subtle variants of training samples in the continuous latent space in a controllable way and thereby renders better representations, which is usually evidenced by more appealing performance in typical
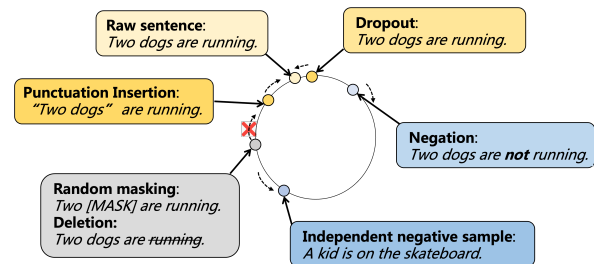

Figure 1: Normalized representation visualization of different augmentation methods and the way they should be optimized.

downstream NLP tasks (e.g., textual semantic similarity) (Zhang et al., 2021; Chen et al., 2021a).

If two sentences with large semantic gaps are paired positively, the representation alignment is likely to deteriorate (Wang and Isola, 2020). Discrete augmentation methods usually have less competitive results, because they can hardly keep the sentence semantically consistent when applied randomly at lexical level. As shown in Figure 1, sentence semantics become incomplete after random masking and word deletion (Wu et al., 2020) so that optimizing their embeddings in the direction of the original sentence (aka anchor) leads to misunderstanding.

To learn better representations, augmentation methods are supposed to generate samples that are not only diverse in representation (Tian et al.,

---

[*] Equal contribution.
[†] Corresponding author.
[1] It refers to a sentence sample in this work.

14459

2020) but also similar in semantics. There is, however, a trade-off between semantic consistency and expression diversity. Bigger differences between vanilla samples and augmented ones also convey less faithful semantics. Therefore, we hypothesize that a good augmentation method in contrastive learning should have desiderata to balance them.

Continuous methods can control semantic changes since they utilize designed network structures to process redundant features (Huang et al., 2021). SimCSE (Gao et al., 2021) utilizes dropout (Srivastava et al., 2014) to obtain different embeddings of the same sentence to construct positive pairs. But such continuous methods lack interpretability to inspire further exploration of sentence augmentation. To better prove our hypotheses and find the promising direction of lexical data augmentation methods, we propose three **S**imple **D**iscrete **A**ugmentation (SDA) methods to satisfy the desiderata to different extents: *Punctuation Insertion* (PI), *Modal Verbs* (MV), and *Double Negation* (DN). Their impacts on the expression diversity increase in a row but semantic consistency with the original sentence tends to diminish gradually. In linguistics, punctuation usually represents pause or tone (e.g., comma, exclamations) which has no specific meaning itself. Modal verbs are used as supplementary to the predicate verb of the sentence indicating attitudes such as permission, request, and so on, which helps to reduce uncertainty in semantics. DN helps two negatives cancel each other out and thereby produces a strong affirmation, whereas the improperly augmented sentence is at risk to be logically confusing.

Although the proposed augmentation methods keep the semantic meaning by carefully adding minor noises, the generated sentences are still literally similar to the original sentence. Recent research (Robinson et al., 2021) has pointed out the feature suppression problem of contrastive learning. This phenomenon could result in shortcut solutions that the model only learns the textual rather than semantic similarity. A focus on hard examples has been proven effective to change the scope of the captured features (Robinson et al., 2021). Thus, we further utilize standard negation to construct text contradicting all or part of the meaning of the original sentence as hard negative samples. By doing so, the model is encouraged to learn to differentiate sentences bearing similar lexical items yet reversed meanings.

To summarize, the contributions of this work are as follows:

- We propose SDA methods (including standard negation) for contrastive sentence representation learning, which leverages discrete sentence modifications to enhance the performance of representation learning (Section 4).

- Comprehensive experimental results demonstrate that SDA achieves significantly better performance, advancing the state-of-the-art performance to a new bar from 78.49 to 79.60 (Section 5).

- Extensive ablations and in-depth analysis are conducted to investigate the underlying rationale and clarify the hyper-parameters choices (Section 6).

2. Related Works

2.1. Sentence Representation Learning

BERT (Devlin et al., 2019) has steered the trajectory of sentence representation towards the technical orientation of Pre-trained Language Models (PLM). A multitude of endeavors (Tan et al., 2020, 2021; Li et al., 2020; Su et al., 2021; Lu et al., 2023a,b) has been dedicated to substantial improvements based on this paradigm, leading to significant advancements in diverse domains. Notably, there is a pronounced practical demand for sentence-level text representations (Conneau et al., 2017; Williams et al., 2018). Consequently, learning unsupervised sentence representations based on PLM has become a focal point in recent years (Reimers and Gurevych, 2019; Zhang et al., 2020). Current state-of-the-art methods utilize contrastive learning to learn sentence embeddings (Kim et al., 2021; Yan et al., 2021; Gao et al., 2021), which in experimental results, can even rival supervised methods. However, to advance unsupervised contrastive learning methods further, data augmentation emerges as a pivotal component.

2.2. Data Augmentation in Contrastive Learning

Early research on contrastive sentence representation learning (Zhang et al., 2020) didn't utilize explicit augmentation methods to generate positive pairs. Later, methods (Giorgi et al., 2021; Wu et al., 2020, 2021) which use text augmentation methods, such as word deletion, span deletion, reordering, synonym substitution, and word repetition, to generate different views for each sentence achieve better results. Compared to augmentation methods applied on text, several studies (Janson et al., 2021; Yan et al., 2021; Gao et al., 2021; Wang et al., 2022a) utilize neural networks, such as dual encoders, adversarial attack, token shuffling, cutoff and dropout, to obtain different embeddings for contrasting. A more recent study DiffCSE (Chuang et al., 2022) designed an extra MLM-based word replacement detection task as an equivalent augmentation. The purpose of data augmentation in
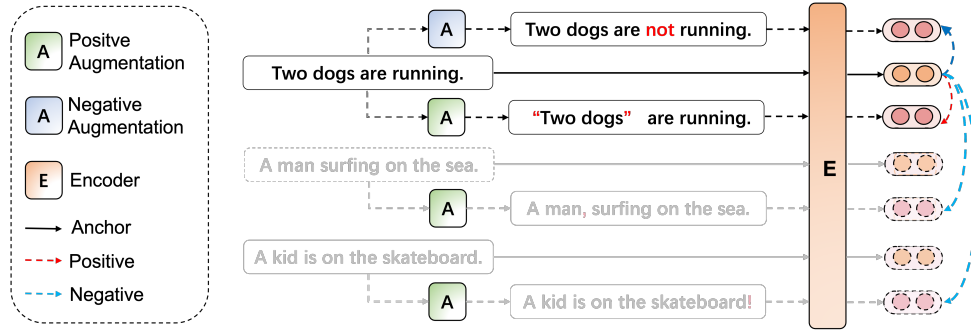
Figure 2: An overview of the framework. The figure can be embodied as a training batch. Each sentence is passed through the augmentation module to generate one positive and one negative for the anchor, the positives generated by other sentences in the batch are also deemed as negatives for the anchor.

this study is to generate both semantic similar and expression diverse samples so that models can learn to extend the semantic space of the input samples.

3. Preliminaries

3.1. Sentence-level Contrastive Learning

Given a set of sentence pairs $\mathcal{D} = \{(x_i, x_i^+)\}$, where sentence pairs (x_i, x_i^+) are semantically similar and deemed as positive pairs. Contrastive learning aims to learn a dense representation $\mathbf{h}_i$ of a sentence x_i by gathering positive samples together while pushing others apart in the latent space (Belghazi et al., 2018). In practice, the training proceeds within a mini-batch of N sentence pairs. The objective is formulated as:

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

where $\mathbf{h}_i$ and $\mathbf{h}_i^+$ respectively denote the representation of x_i and x_i^+, $\mathrm{sim}()$ is the cosine similarity function and τ is the temperature parameter. Under the unsupervised setting, the semantically related positive pairs are not explicitly given. Augmentation methods are used to generate x_i^+ for training sample x_i.

3.2. Unsupervised SimCSE

In transformer model $f(\cdot)$, there are dropout masks placed on fully-connected layers and attention probabilities. SimCSE[2] builds the positive pairs by feeding the same input x_i to the encoder twice, i.e., $x_i^+ = x_i$. With different dropout masks z_i and z_i^+, the two separate output sentence embeddings constitute a positive pair as follows:

$$\mathbf{h}_i = f_{z_i}(x_i), \mathbf{h}_i^+ = f_{z_i^+}(x_i) \quad (2)$$

---

[2]The SimCSE mentioned in this article are all under the unsupervised setting.

Method	Sentence
None	He travelled widely in Europe.
PI	He **,** travelled widely in Europe.
MV	He **must have** travelled widely in Europe.
DN	**It is not the fact that** he **didn't** travel widely in Europe.
Negation	he **didn't** travel widely in Europe.

Table 1: An example of different methods to generate the augmented sentence. The highlighted red texts denote changes after augmentation.

3.3. Dependency Parsing and Syntax Tree

Dependency parsing represents the relationships between words in a sentence in the form of dependencies. Each word in the sentence is connected to another word, indicating its grammatical role and the type of relationship it has with other words.

Syntax trees represent the hierarchical structure of a sentence's grammar. They consist of nodes, where each node represents a word or a grammatical unit, and edges represent syntactic relationships. The root node represents the main clause, and branches indicate phrases and sub-clauses.

4. Methodology

In this work, the augmentation module to generate positive samples for training data is denoted as $A(\cdot)$. As illustrated in Fig. 2, we utilize $A(\cdot)$ to subtly reword the original sentence in an attempt to change the representation of the sentence to a limited extent on the premise that the sentence roughly remains unchanged semantically. Afterword, Eq. 2 can be rewritten as follows:

$$\mathbf{h}_i = f_{z_i}(x_i), \mathbf{h}_i^+ = f_{z_i^+}(A(x_i)) \quad (3)$$

In practice, we utilize spaCy[3] for dependency
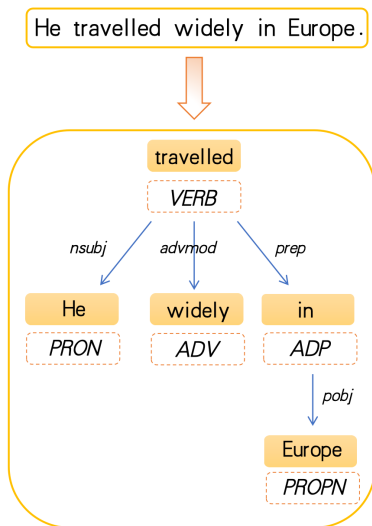
---

[3]https://spacy.io/

14461

Figure 3: The syntax tree constructed through dependency parsing and its representation.

parsing[4] and then implement punctuation insertion, modal verbs, and double negation. As shown in Figure 3, first, each sentence needs to be processed through dependency parsing to obtain a central word and the dependencies between the other words with the central word. Then, with the central word as the root node and other words as child nodes, the corresponding syntax tree can be constructed through dependencies. Based on this, we can design rules by traversing the syntax tree in level to implement the SDA augmentation for sentences as shown in Table 1 and Appendix A.

4.1. Punctuation Insertion (PI)

Punctuation usually doesn't carry specific information, whereas changing punctuation can have a slight effect on the sentence. We devise a few scalable rules and thereby generate semantically similar examples as positives in a relatively controllable way: (1) If a sentence has a subordinate clause, insert a comma <,> between them; (2) If there is a noun subject in the sentence, enclose it with double quotes <" "> or append a comma <,>; (3) Replacing the punctuation at the end of a sentence with an exclamation point <!> or appending it to the sentence.

These simple rules insert punctuation marks at different positions of a sentence to ensure the diversity of augmentations. Moreover, the rules are context-aware schemes. For example, <!> at the end of sentences expresses emphasis tone, <,> between clauses means to pause, and <" "> frames

the subject to highlight it. In this way, PI preserves sentence structure and meaning to the greatest extent and minimizes the magnitude of semantic changes.

4.2. Modal Verbs (MV)

Adding modal verbs to a sentence is also a positive generation method that slightly adjusts the meaning of the sentence. The semantic change brought by MV may be greater than punctuation, but it is still linguistically reasonable. We first collect a list of modal verbs such as: "must", "should", and "ought to", as well as phrases that have the same function such as "have to", "can't but", "can't help to", etc. Then we randomly pick one phrase to modify the predicate verb of the main clause. MV is more radical than PI because it supplements the speaker's attitudes (e.g., likelihood, permission, obligation, and so on) towards the predicate verbs of the sentence, uncertainty in semantics is thus reduced to some extent.

4.3. Double Negation (DN)

In English and Chinese, double negatives cancel one another and produce an affirmative. We leverage this linguistic rule to generate double negation as the positive of the raw sentence. But double negative is difficult to implement and multiple negative words in one sentence could make it confusing. Therefore, DN is considered to be the most radical among all three augmentation methods.

The implementation is to apply a negative transformation to a sentence twice. Either inserting a negative word or removing a negative is regarded as a negative transformation. The negative words are used to modify nouns or verbs in the sentence, and corresponding negative words will be selected, e.g., "no" is inserted before nouns and "not" before verbs. If the sentence is only able to apply negative transformation once, we simply add a negation phrase, such as "It is not true that" or "It can't be that", in front of the sentence as a second negative transformation.

4.4. Negation as Negative Sample

The aforementioned augmentation methods are to generate semantically consistent positive samples for the anchor sentence. To ensure the model can learn the semantic correlation rather than the textual shortcut solutions, we generate a negation sentence for every training sample serving as a negative sample. The negation sentence and the anchor are highly similar in the text but have opposite semantic meanings. Learning to distinguish a negation sentence from the anchor is helpful to alleviate feature suppression (Wang et al., 2022b). However,

---

[4]We empirically think that grammatically correct sentence transformation is more reasonable than the stochastic manipulation without complying with basic grammar rules explicitly.

Method	Augmented %
Punctuation Insertion	98.14%
Modal Verbs	88.32%
Double Negation	87.89%

Table 2: The ratio of three augmentation protocols.

Param	BERT$_{base}$	RoBERTa$_{base}$	CN-RoBERTa$_{base}$
learning rate	2e-5	1e-5	5e-5
batch size	128	64	128
τ	0.05	0.05	0.05
δ	0.5	0.5	0.5

Table 3: Hyper-parameter settings for English and Chinese (CN) pre-trained model.

since negative samples in contrastive learning are usually independent sentences that are completely irrelevant to the original sentence, it is inappropriate to treat negation sentences the same as those independent negative samples. Therefore, we relax the restriction of the negation sentence in the optimization with a hyper-parameter δ, and rewrite the contrastive loss function in Eq. 1 as follows:

$$l_i = -\frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau} + e^{[\text{sim}(\mathbf{h}_i, \mathbf{h}_i^-) - \delta]/\tau}} \quad (4)$$

where $\mathbf{h}_i^-$ is the representation of the negation sentence x_i^-. And δ can be considered as a margin which relaxed the distance between sentence x_i and x_i^- compared to other independent negative samples. The implementation of negation transformation has been introduced in Section 4.3, and the rules guarantee the transformation is applicable for arbitrary sentences.

5. Experiment Settings

5.1. Datasets

Semantic textual similarity (STS) task measures the semantic similarity of any two sentences, it uses Spearman's correlation (Myers et al., 2013) to measure the correlation between the ranks of predicted similarities and the ground-truth labels. We conduct our experiments on seven standard English and one Chinese STS datasets. For English STS, we follow SimCSE[5] (Gao et al., 2021) to use one million sentences randomly drawn from English Wikipedia for training. Datasets STS12-STS16 have neither training nor development sets, and thus we evaluate the models on the development set of STS-B to search for optimal settings of hyper-parameters. For Chinese STS, we choose the unlabeled corpus data in SimCLUE dataset[6] for training, which contains about 2.2 million sentences. In the test phase, we use the Chinese STS-B from CNSD[7] without the training set.

5.2. Baselines

For English STS, we compare our methods to competitive contrastive methods including 1) **IS-BERT**

maximizes agreement between sentence embeddings and token embeddings (Zhang et al., 2020), 2) **CT-BERT** uses different encoders to obtain augmented views (Janson et al., 2021), 3) **DeCLUTR** samples spans from the same document as positive pairs (Giorgi et al., 2021), 4) **ConSERT** designs four kinds of augmentation methods, i.e., adversarial attack, token shuffling, cutoff and dropout (Yan et al., 2021), 5) **SimCSE** utilizes standard dropout as augmentation (Gao et al., 2021) and 6) **ESim-CSE** (Wu et al., 2021) introduces word repetition and momentum contrast mechanisms (He et al., 2020a), 7) **ArcCSE** (Zhang et al., 2022) modifies contrastive loss with angular margin and models entailment relation of triplet sentences, 8) **DiffCSE** (Chuang et al., 2022) adds an extra MLM-based word replacement detection task as an equivalent augmentation.

5.3. Training Details

We use the same contrastive learning framework introduced by SimCSE, and train three models with Eq. 4. Each model uses one of the three augmentation methods to generate the positive sample and negation transformation to obtain the extra negative sample. Since not all sentences can be augmented with the rules as detailed in Table 2, we pair the sentence with itself.

Our implementation is based on `transformers` package. Following SimCSE, we use pre-trained checkpoints of English BERT[8] (uncased) (Devlin et al., 2019) or RoBERTa[9] (Liu et al., 2019). For the Chinese STS task, we use Chinese-RoBERTa-base[10] as our pre-trained model. Two MLP layers with batch normalization are added on top of the [CLS] representation in training but removed in evaluation. We train our model with Adam optimizer for one epoch and evaluate the model every 250 training steps on the development set and keep the best checkpoint for the final evaluation on test sets. The settings of important hyper-parameters are listed in Table 3.

---

[5]https://github.com/princeton-nlp/SimCSE

[6]https://github.com/CLUEbenchmark/SimCLUE

[7]https://github.com/pluto-junzeng/CNSD

[8]https://huggingface.co/bert-base-uncased

[9]https://huggingface.co/roberta-base

[10]https://huggingface.co/hfl/chinese-roberta-wwm-ext

Methods	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
			BERT Models					
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
ConSERT$_{base}$♠	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE-BERT$_{base}$◇	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
ESimCSE-BERT$_{base}$♣	69.79	83.43	75.65	82.44	79.43	79.44	71.86	77.43
ArcCSE-BERT$_{base}$♠	72.08	84.27	76.25	82.32	79.54	79.92	72.39	78.11
DiffCSE-BERT$_{base}$♡	**72.28**	84.43	76.47	83.90	80.54	**80.59**	71.23	78.49
Our Methods								
Punctuation Insertion	71.92	84.38	**76.84**	**83.92**	80.45	80.25	74.26	**78.86**
Modal Verbs	71.35	84.45	76.60	83.77	**80.57**	80.31	**74.85**	78.84
Double Negation	71.23	**84.49**	75.88	83.34	79.37	79.67	74.32	78.33
Ensemble	72.31±0.38	83.66±0.64	76.59±0.38	84.10±0.20	80.41±0.42	80.17±0.52	72.78±0.49	78.57±0.43
			RoBERTa Models					
DeCLUTR-RoBERTa$_{base}$◇	52.41	75.19	65.52	77.12	78.63	72.41	68.62	69.99
SimCSE-RoBERTa$_{base}$◇	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
ESimCSE-RoBERTa$_{base}$♣	69.90	82.50	74.68	83.19	80.30	80.99	70.54	77.44
DiffCSE-RoBERTa$_{base}$♡	70.05	83.43	75.49	82.81	82.12	82.38	71.19	78.21
Our Methods								
Punctuation Insertion	70.92	83.59	76.87	**83.73**	**82.42**	**83.02**	74.89	79.35
Modal Verbs	**72.37**	**83.80**	77.51	83.58	82.29	82.98	74.69	**79.60**
Double Negation	71.07	83.56	**77.60**	83.38	81.59	81.82	**75.44**	79.21
Ensemble	72.64±0.62	83.45±0.38	76.90±0.17	83.56±0.21	81.82±0.38	82.76±0.31	74.77±0.6	79.37±0.07

Table 4: Performance comparison with existing competitive methods on STS tasks. The bold numbers top the competitions with the same pre-trained encoder and datasets. ◇: results from (Gao et al., 2021); ♣: results from (Wu et al., 2021); ♡: results from (Zhang et al., 2020); ♠: results from (Chuang et al., 2022).

Method	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
SimCSE-RoBERTa$_{base}$◇	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
ArcCSE-BERT$_{base}$♠	79.91	85.25	**99.58**	**89.21**	84.90	89.20	74.78	86.12
DiffCSE-RoBERTa$_{base}$♡	82.82	88.61	94.32	87.71	**88.63**	90.40	76.81	87.04
Our Methods								
Punctuation Insertion	**83.59**	87.79	93.81	88.10	87.81	**91.60**	76.93	**87.09**
Modal Verbs	82.24	**88.88**	93.67	88.10	87.10	90.00	76.58	86.65
Double Negation	81.73	87.26	93.61	88.12	87.26	89.00	77.28	86.32
Ensemble	82.26±0.8	88.61±0.29	93.96±0.59	88.81±0.07	87.64±0.82	88.60±0.60	76.23±0.93	86.59±0.17

Table 5: SentEval transfer tasks results of different models. ◇: results from (Gao et al., 2021); ♡: results from (Zhang et al., 2020); ♠: results from (Chuang et al., 2022).

6. Results and Analysis

6.1. Main Results

The experimental results including the three SDA methods are shown in Table 4. Ensemble refers to the integration of three SDA methods. The implementation involves randomly selecting one augmentation method for each training sentence to generate positive samples. To eliminate the uncertainty introduced by the random seed, the experiment was conducted five times, and the ensemble results are presented as a range. In the case of BERT and RoBERTa, ensemble is represented as the average of three SDA methods. It should be noted that the results in the Table 4 have integrated standard negation as hard negative examples.

Overall, experimental results clearly demonstrate that the proposed SDA methods lead to significant improvement in unsuperivsed sentence representation learning, which is evidenced by pushing state-of-the-art performance to a new bar of 79.60 with an overall improvement of 1.11%. More specifically, when we look into those RoBERTa-based cases

(Table 4, the bottom half block), three proposed data augmentation methods (i.e., PI, MV, and DN) consistently surpass the previous state-of-the-art performance by a clear margin across all STS.

In BERT-based cases, all methods can slightly outperform to state-of-the-art methods across all datasets. We suspect the reason why our methods perform worse on BERT than on RoBERTa is caused by the introduction of next sentence prediction task (NSP) in pre-training BERT, which is not adopted in RoBERTa. The task predicts whether two sentences are semantically consecutive, making BERT more sensitive to the sentence-level semantic change as compared to RoBERTa. The learned representations from BERT are less likely to conform to the "semantic consistency" hypothesis. Previous works (Liu et al., 2019; He et al., 2020b) also express similar opinions that NSP is not helpful for sentence representation.

It is interesting to find out that the performance of PI and MV are generally better than that of DN in most cases. These observations can be explained by our "semantic-expression trade-off" hypothesis.

Methods	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
SimCSE-RoBERTa$_{base}$	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
Modal Verbs (MV)	**72.37**	**83.80**	**77.51**	**83.58**	**82.29**	**82.98**	**74.69**	**79.60**
w/o Negation as Negative	<u>71.80</u>	<u>82.91</u>	<u>76.19</u>	<u>83.50</u>	81.55	<u>82.37</u>	68.78	<u>78.16</u>
w/o Positive Augmentation	68.48	82.23	73.70	81.15	<u>81.56</u>	81.26	<u>72.17</u>	77.22

Table 6: The effect of the positive augmentation and the negation transformation. The winners and runners-up are marked in bold font and underlined, respectively.

Methods	Chinese STS-B
Roberta-base (last CLS)	68.25
SimCSE-RoBERTa-base	71.10
Punctuation Insertion	71.98
Modal Verbs	**72.12**
Double Negation	71.65

Table 7: Performance comparison on Chinese STS task. All results come from our experiments, where SimCSE is the result we reproduced on Chinese dataset.

Concretely, PI and MV introduce diversity in expression while retaining the semantic meaning of anchor sentences significantly. In contrast, the sophisticated correlation between linguistic expression (i.e., syntax, vocabulary) and the subtlety in semantics makes DN difficult to faithfully replicate the meaning of anchor sentences, deteriorating unsupervised learning performance. For example, "I'm hungry more" is not semantically equivalent to "I'm not hungry no more", and actually, they convey opposite meanings to some extent. Although we use rules based on syntactic analysis to limit the occurrence of this situation, it is difficult to completely eliminate it in practice. In the next section, we will conduct comprehensive ablation studies to elaborate on the importance of semantic consistency.

6.2. Transfer Tasks

Furthermore, to analyze the actual impact of semantic consistency on sentence representation, we evaluate the SentEval transfer tasks in Table 4. The effect of the RoBERTa series models are better than that of BERT. So we selected the RoBERTa models to compare them with the previous state-of-the-art unsupervised contrastive learning models in transfer tasks using the SentEval toolkit[11]. For each task, SentEval trains the average embedding of the last two layers as the sentence embedding and evaluates the performance on the downstream task. To ensure a fair comparison, we did not introduce any additional training tasks, such as masked language modeling. As shown in Table 5, our SDA methods are able to match or surpass state-of-the-art ap-

proaches in certain tasks while falling behind other methods in some tasks. This suggests that different sentence representation learning methods excel at different downstream tasks, which may be related to their training paradigms. Nevertheless, our SDA method exhibits strong competitiveness in tasks other than the SUBJ. It's important to note that the results in Table 4 include the standard negation.

6.3. Ablation Study

To analyze the improvement made by the generated positive sample and the extra negative sample respectively, we compare four settings: using no discrete augmentation (SimCSE), using both positive augmentation (MV) and negation as negative, only using positive augmentation (w/o Negation as Negative) and only using negation as negative (w/o Positive Augmentation). We present the result of MV here in Table 6 since its general good performance. We can observe that either using positive augmentation or using negation as negative achieves better performance compared to the method without discrete augmentation, and combining the two methods can make further improvements.

Notice that although using negation as negative performs seems worse on most STS series (STS12-16, STS-B) datasets than using positive augmentation, it obtains 3.39% lift on SICK-R dataset. By measuring the average Jaccard similarity (Jaccard, 1912) between the test sentence pairs, we find that sentence pairs in SICK-R (57.24%) are much more similar on text than the STS series of datasets (47.39%), which supports our argument that textual similar negative sample can help alleviate feature suppression problem.

6.4. Results on Chinese Datasets

Results on Chinese STS datasets are presented in Table 7. As compared to state-of-the-art method, PI, MV and DN gain an improvement of 0.78%, 1.02% and 0.55% respectively, demonstrating that the proposed SDA methods can be seamlessly generalized to other languages like Chinese. This is consistent with our expectation since some linguistic phenomena are common in most of languages,

---

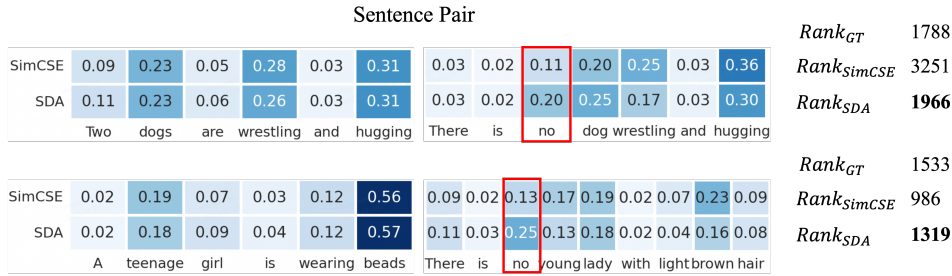[11]https://github.com/facebookresearch/SentEval

Figure 4: Cases from SICK-R test set. The heatmap visualizes the spectrum of the weight of words in the sentence representation. We rank sentence pairs based on sorted similarity scores in ascending order. A better ranking should be closer to the ground truth (GT).

Augmentations			STS-B
None (Unsup-SimCSE)			82.45
Cropping	10%	20%	30%
	77.81	71.38	63.62
Word Deletion	10%	20%	30%
	75.89	72.20	68.24
Synonym Replacement			77.45
Mask 15%			62.21
Word Repetition			84.09
Punctuation Insertion			84.55
Modal Verbs			**84.99**
Double Negation			84.12

Table 8: Performance comparison with discrete augmentation methods. Dropout is set to default in training as adopted in SimCSE and its variants.

Augmentations	avg. STS
None (Unsup-SimCSE)	76.57
Random Insertion	76.22
Punctuation Insertion	77.90

Table 9: Performance comparison between random augmentation and rule-based augmentation.

e.g., minimal changes on punctuation or tone will not affect semantics too much.

6.5. Random manipulation vs. rule-based augmentation

We further compare different discrete augmentation methods such as cropping, word deletion, synonym replacement, masking, and word repetition as well as no discrete augmentation[12] under the same contrastive learning framework. The results are shown in Table 8. A comparison of the development set of STS-B shows that all the proposed augmentation methods outperform previous methods. Cropping, word deletion, synonym replacement, and masking (Wei and Zou, 2019; Wu et al., 2020; Meng et al., 2021) even perform worse than applying no discrete augmentation. We assume the worse performance is in part due to the randomness inserted by these manipulations, dramatically changing the sentence structure. Word repetition duplicates words in a sentence and rarely changes the semantics, thus it gets a competitive result,

which again demonstrates the importance of keeping semantic consistency.

We then try different manipulations where the randomness is reduced and thus semantic shift is smaller. More specifically, in Table 9, we test three settings, no manipulation, inserting punctuation based on rules (i.e., PI), as well as randomly inserting punctuation (Random Insertion). Notice that we don't use negation as a negative sample in this experiment to control the variable. It turns out that applying augmentation according to the rules stands out among them, while Random Insertion performs worst. This result again verifies the importance of retaining semantic consistency since randomly inserted punctuation is more likely to disturb sentence semantics while rule-based PI generates syntax-correct sentences, boosting representation learning.

6.6. Case Study

We calculate the weight of each word in the sentence representation by gathering the attention weights of the output token. We find that the learned representations of our method pay more attention to negative words. Since negative words have great influences on sentence semantics, SDA has largely excelled in those cases where negative words occurred, as exemplified in Fig. 4.

6.7. Parameter Analysis

We first compare the results with different augmentation proportions. When we set the augmentation proportion to $x\%$, only $x\%$ samples in the dataset

---

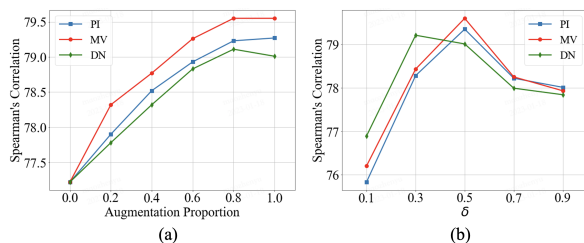[12]When no discrete augmentation is applied, the setting falls back to SimCSE.

Figure 5: Parameter sensitivity for (a) proportion of augmented sentence pairs in training and (b) margin δ.

are paired with augmentation, and $(1-x)\%$ samples are paired with themselves, which means $(1-x)\%$ samples use dropout as augmentation method. We can see from Fig. 5 (a), the performance increases with a higher augmentation proportion.

Then, we study how parameter δ influences our methods. By tuning δ rate from 0.1 to 0.9, we collect the results in Fig. 5 (b). We find the best settings are 0.5 for both PI and MV, and 0.3 for DN. Since the augmented sentence constructed by DN is more similar to the negation sentence on text than the other two augmentation methods, thus a tighter restriction is needed to help differentiate the positive and negative.

7. Conclusion

In this work, we study the role of data augmentation in contrastive sentence representation learning and argue that the desiderata of reasonable data augmentation methods are to balance semantic consistency and expression diversity. Three simple yet effective discrete augmentation methods, i.e., punctuation insertion, modal verbs, and double negation are developed to substantiate the hypothesis. We further alleviate feature suppression with a negated sentence as a negative sample. Armed with well-executed discrete data augmentation, we achieve better results compared to more complicated state-of-the-art methods. Our work offers a promising direction of textual data augmentation and can be readily extended to multiple languages and beyond along this direction.

8. Limitations

Since our augmentation methods are implemented by rules, there could be some cases that the rules are not applicable or cause syntax mistakes. We have tried our best to alleviate such phenomenon and make the rule cover as many samples as possible as shown in Table 2. We believe that it is absolutely feasible to use advanced LLMs like GPT4 to improve the quality of double negative generation,

especially in cases where some simple methods do not work well. However, the cost of API calls or LLMs deployment cannot be ignored, especially for academia. Using LLMs to combine simple methods like ours to accelerate inference time or reduce resource dependencies is a valuable new direction.

In this work, we explore discrete text augmentation methods and propose three of them which achieve better results on STS tasks than previous studies. However, we believe that effective discrete augmentation methods can be abundant. Our purpose is not to find the best augmentation method, but to reveal the underlying rationales of a good augmentation method. Hence, we make our code available for replication and extension by the community.

9. Bibliographical References

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Acl 2023 tutorial: Retrieval-based language models and applications. *ACL 2023*.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR.

Jiaao Chen, Dinghan Shen, Weizhu Chen, and Diyi Yang. 2021a. Hiddencut: Simple data augmentation for natural language understanding with better generalizability. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4380–4390.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ting Chen, Calvin Luo, and Lala Li. 2021b. Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems*, 34:11834–11845.

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023. Improving contrastive learning of sentence embeddings from AI feedback. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11122–11138, Toronto, Canada. Association for Computational Linguistics.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. *arXiv preprint arXiv:2204.10298*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. 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 (Volume 1: Long Papers)*, pages 879–895.

Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126. PMLR.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020a. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020b. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information

estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Zhiqi Huang, Lu Hou, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2021. Ghostbert: Generate more features with cheap operations for bert. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6512–6523.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.

Sverker Janson, Evangelina Gogoulou, Erik Ylipää, Amaru Cuba Gyllensten, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations, 2021*.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jinghui Lu, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023a. Punifiedner: A prompting-based

unified ner system for diverse datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13327–13335.

Jinghui Lu, Dongsheng Zhu, Weidong Han, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023b. What makes pre-trained language models better zero-shot learners? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2288–2303.

Yu Meng, Chenyan Xiong, Payal Bajaj, Paul Bennett, Jiawei Han, Xia Song, et al. 2021. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *Advances in Neural Information Processing Systems*, 34:23102–23114.

Jerome L Myers, Arnold D Well, and Robert F Lorch. 2013. *Research design and statistical analysis*. Routledge.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

N. Reimers and I. Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. 2021. Can contrastive learning avoid shortcut solutions? *Advances in neural information processing systems*, 34:4974–4986.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Fei Tan, Yifan Hu, Changwei Hu, Keqian Li, and Kevin Yen. 2020. Tnt: Text normalization based pre-training of transformers for content moderation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4735–4741.

Fei Tan, Yifan Hu, Kevin Yen, and Changwei Hu. 2021. Bert-beta: A proactive probabilistic approach to text moderation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8667–8675.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839.

Bing Wang, Liang Ding, Qihuang Zhong, Ximing Li, and Dacheng Tao. 2022a. A contrastive cross-channel data augmentation framework for aspect-based sentiment analysis. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6691–6704.

Hao Wang, Yangguang Li, Zhen Huang, Yong Dou, Lingpeng Kong, and Jing Shao. 2022b. Sncse: Contrastive learning for unsupervised sentence embedding with soft negative samples. *arXiv preprint arXiv:2201.05979*.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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, Volume 1 (Long Papers)*, pages 1112–1122.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2021.

Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. *arXiv preprint arXiv:2109.04380*.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

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

Dejiao Zhang, Shang-Wen Li, Wei Xiao, Henghui Zhu, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2021. Pairwise supervised contrastive learning of sentence representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5786–5798.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610.

Yuhao Zhang, Hongji Zhu, Yongliang Wang, Nan Xu, Xiaobo Li, and Binqiang Zhao. 2022. A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4892–4903.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.

14470

A. Pseudo-Codes of Algorithms

Algorithm 1, 2, 3 are the pseudo-codes of the three proposed augmentation methods. The input sentence is first parsed by a spaCy sentence parser to obtain each token's part-of-speech (POS) and dependency tag. Then we apply the rules described in the algorithms on the parsed sentence to construct the augmented sentences.

Algorithm 1 Punctuation Insertion

Input: one sentence S
Output: the augmented sentence
1: **if** subordinate clause in $\mathcal{S}$ **then**
2:     $pos \leftarrow$ Start position of subordinate clause
3:     **return** $S.\text{Insert}(',', pos)$
4: **end if**
5: **if** noun subject in S **then**
6:     $pos_e \leftarrow$ End position of the noun subject
7:     $S \leftarrow S.\text{Insert}(',', pos_e)$
8:     **return** S
9: **end if**
10: **if** S end with punctuation $punc$ **then**
11:     $S \leftarrow S.\text{Replace}(punc, '!')$
12: **else**
13:     $S \leftarrow S.\text{Append}('!')$
14: **end if**
15: **return** S

Algorithm 2 Modal Verbs

Input: one sentence S, set of modal verbs $\mathcal{V}_{mv}$
Output: the augmented sentence
1: $mv \leftarrow \text{Random.Choice}(\mathcal{V}_{mv})$
2: **for** word w in sentence **do**
3:     **if** w is be-verb **then**
4:         **return** $S.\text{Replace}(w, mv + ' be')$
5:     **end if**
6:     **if** $w.dep == ROOT$ and $w.pos == verb$
        then
7:         **return** $S.\text{Replace}(w, mv + w.lemma)$
8:     **end if**
9: **end for**
10: **return** S

Algorithm 3 Double Negation

Input: one sentence S
Output: the augmented sentence
1: $S' \leftarrow S, count \leftarrow 0$
2: **for** word w in S **do**
3:     **if** $count == 2$ **then**
4:         **return** S
5:     **end if**
6:     **if** w is negative word **then**
7:         $S \leftarrow S.\text{Delete}(w)$
8:         $count + +$
9:     **end if**
10:     **if** $w.dep == aux$ **then**
11:         $S \leftarrow S.\text{Replace}(w, w + ' not')$
12:         $count + +$
13:     **end if**
14:     **if** $w.dep == ROOT$ and w is verb **then**
15:         $S \leftarrow S.\text{Replace}(w, 'do not ' + w)$
16:         $count + +$
17:     **end if**
18: **end for**
19: **if** $count == 1$ **then**
20:     **return** $'Not that ' + S$
21: **else**
22:     **return** S'
23: **end if**