Unsupervised hard Negative Augmentation for contrastive learning

Anonymous ACL submission

Abstract

We present Unsupervised hard Negative Augmentation (UNA), a method that generates synthetic negative instances based on the term frequency-inverse document frequency (TF-IDF) retrieval model. UNA uses TF-IDF scores to ascertain the perceived importance of terms in a sentence and then produces negative samples by replacing terms with respect to that. Our experiments demonstrate that models trained with UNA improve the overall performance in semantic textual similarity tasks. Additional performance gains are obtained when combining UNA with the paraphrasing augmentation. Further results show that our method is compatible with different backbone models. Ablation studies also support the choice of having a TF-IDF-driven control on negative augmentation.

1 Introduction

Self-supervised contrastive learning (SSCL) has emerged as a potent method for a variety of linguistic tasks such as sentiment analysis, textual entailment, and similarity retrieval (Gao et al., 2021; Yan et al., 2021; Giorgi et al., 2021; Wu et al., 2020; Fang and Xie, 2020). Data augmentation in SSCL is oftentimes used to generate positive instances, based on an anchor instance, with representations that are trained to be more similar to each other or to the anchor itself (He et al., 2020; Grill et al., 2020). While various data augmentations are effective in supervised NLP tasks (Wei and Zou, 2019; Karimi et al., 2021), they appear to lose efficacy when applied within the context of contrastive learning. For example, a dropout mask outperformed established augmentations such as cropping, word deletion, and synonym replacement (Gao et al., 2021). This could indicate that the properties of the chosen text augmentations are at odds with the SSCL objective, which seeks model invariance to these changes, while it should not.

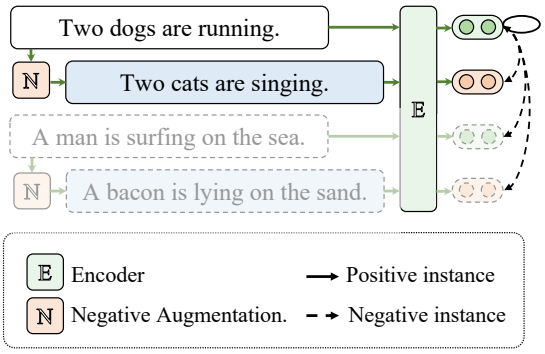

Figure 1: Illustration of UNA. Negative augmentation, driven by TF-IDF, is applied to anchor instances to generate sentences with potential semantic differences.

By contrast, negative data augmentation has garnered much less attention. Sinha et al. (2021) introduced negative data augmentation for computer vision tasks. In addition, Tang et al. (2022) examined as many as 16 augmentation strategies for NLP tasks with SSCL, discovering that synthetic sentences augmented with certain methods (*e.g.* frequently introducing grammatical errors) perform better when treated as negative samples. Interestingly, their main performance gain originated from applying augmented instances as positive samples whilst including negatively augmented instances tended to degrade the overall performance, especially in Semantic Textual Similarity (STS) tasks.

In this paper, we argue that the potential contribution of synthetic negatives in SSCL has been undervalued. We therefore propose Unsupervised hard Negative Augmentation (UNA), an augmentation strategy for generating negative samples in SSCL. Figure 1 depicts how UNA is applied during the model pre-training stage (details in section 3). UNA's core premise is the deployment of a TF-IDF-driven methodology to generate negative pairs. This improves model performance in downstream STS tasks, with additional gains when applied in conjunction with paraphrasing.

2 Preliminaries

2.1 Contrastive learning

Contrastive learning is a training method that uses a Siamese neural network structure to learn transferable representations (Das et al., 2016; He et al., 2020; Chen et al., 2020). The key idea is to bring similar data samples closer to each other while pushing dissimilar ones apart. Given a batch of B training sentences, let x_i, x_j, and x_k be the anchor, positive, and negative sentences, respectively. Positive instances are defined as different views of the anchor (usually generated from different augmentations), while negative instances are other data points in the training batch. We denote their corresponding embeddings with $\mathbf{h}_i$, $\mathbf{h}_j$, and $\mathbf{h}_k$, respectively. The InfoNCE loss is used for self-supervised instance discrimination:

$$\mathcal{L} = \mathbb{E}\left[-\log \frac{e^{f_{\text{sim}}(\mathbf{h}_i,\mathbf{h}_j)/\tau}}{\sum_{k=1}^{B} \mathbb{1}_{[k\neq i,j]} e^{f_{\text{sim}}(\mathbf{h}_i,\mathbf{h}_k)/\tau}}\right], \quad (1)$$

where τ is a temperature hyperparameter, and the f_{sim} function is a similarity metric (*e.g.* cosine similarity). The objective of the loss function is to maximise the similarity between positive pairs while minimising it between negative pairs. This encourages the neural network to capture discriminative and potentially informative features for downstream tasks.

2.2 Hard negative sampling

Hard negative samples, which are negative samples that are challenging for the model to differentiate from the anchor instance, have proven to be effective in contrastive learning (Kalantidis et al., 2020). Instead of random sampling, there has been a focus on obtaining hard negatives using samples that are close to the anchor instance in the embedding space (Robinson et al., 2021). This may encourage some separation (disambiguation) between instances that may be close in the embedding space but have contrastive semantic interpretations. In addition, synthetic instances, obtained via transformations such as MixUp (Zhang et al., 2018) or Cutmix (Yun et al., 2019), have been used (Sinha et al., 2021; Shu et al., 2022). This has showcased the utility of transformation-based generative samples that are not present in an observed dataset.

3 Unsupervised hard Negative Augmentation (UNA)

Terms with a higher degree of substance (*i.e.* words that are less common) are on average more important in determining the meaning of a sentence (Ramos et al., 2003), thus swapping such words may generate semantically distant negatives for the model to train upon (see Table B2). Motivated by Xie et al. (2020), who used TF-IDF (Luhn, 1958; Sparck Jones, 1972) to generate positive instances by swapping the words in the sentence that are less important, we now reverse this paradigm. Our proposed method (UNA) introduces a TF-IDF-driven generation of hard negative pairs, whereby words with more substance have a greater probability of being swapped, and more common words do not. UNA consists of 3 steps that we describe next (see also Appendix, Algorithm 1).

Step 1 — Derive a TF-IDF representation. A TF-IDF score for a term t in a document d from a corpus $\mathcal{D}$ with N documents is given by:

$$\text{tf-idf}(t, d, \mathcal{D}) = \text{tf}(t, d) \times \text{idf}(t, \mathcal{D}), \quad (2)$$

where $\text{tf}(t, d) = \log(1 + n_t/n)$ and $\text{idf}(t, \mathcal{D}) = -\log(N_t/N)$. n_t and n respectively denote the count of term t and the total count of terms in document d. N_t denotes the number of documents from $\mathcal{D}$ that contain term t. The TF-IDF representation is held in a matrix $\mathbf{Z} \in \mathbb{R}^{N\times m}$, where m is the number of terms in the vocabulary.

Step 2 — Determine which terms will be replaced at the sentence level. For a sentence in the corpus represented by the TF-IDF vector $\mathbf{z} \in \mathbf{Z}$, let p_i denote the probability of replacing term i. p_i is determined as follows:

$$p_i = \min\left(\beta\left(z_i - \min(\mathbf{z})\right)/C, 1\right), \quad (3)$$

$$C = \frac{1}{n_z}\sum_{i=1}^{n_z}\left(z_i - \min(\mathbf{z})\right), \quad (4)$$

where z_i is the i-th element of $\mathbf{z}$, n_z is the number of terms that are present in the sentence, and β is a hyperparameter that tunes the augmentation magnitude. In our experiments, we set $\beta = 0.5$ (see Table C5), and thus $p_i \in [0, 1]$ with a mean of 0.5 (See Equation 5). Note that we always set $p_i = 1$ for the term with the greatest TF-IDF score in a sentence. The intuition behind this is to have at least one term replaced in every sentence, which guarantees that all augmented negative samples are different from their paired original sentences.

Step 3 — Sample replacement terms based on the level of information. We aim to choose replacement terms that have a similar level of importance to the terms that are replacing (based on Step 2) with respect to their TF-IDF representation in our corpus. To perform this, we rely on the maximum TF-IDF score of terms in the corpus. For a term j, this is denoted by s_j, which is the maximum value of the j-th column of $\mathbf{Z}$. To replace term i we then sample from a set of terms located in the maximum TF-IDF score vicinity of term i. We determine a radius of r terms (below and above the reference maximum TF-IDF score of term i) and sample a replacement term with respect to their maximum TF-IDF scores. By encouraging replacements of terms with alternatives that have a comparable level of importance in a sentence, we generate new sentences that are more likely to retain the original structure but convey something semantically distant. Hence, these augmented sentences, guided by the TF-IDF representation $\mathbf{Z}$, could be considered as hard negative samples.

4 Experiments and results

Augmentation implementation. UNA is applied during the self-supervised pre-training stage.[1] For B sentences in a training batch, we generate B negative sentences with UNA, doubling the batch size. In our experiment setup, this process is carried out once every 5 training batches. The TF-IDF representation has a vocabulary size of $451{,}691$ terms, which is obtained from the training dataset (see Appendix A). The only text preprocessing operations performed are basic tokenisation and lowercasing.

In addition, we incorporate UNA along with paraphrasing, where positive instances are generated by utilising a paraphrasing model, namely PEGASUS (Zhang et al., 2020) (see Appendix C). Paraphrasing rephrases the anchor sentence while maintaining its original meaning. All sentences in the training corpus are paraphrased to be considered as the corresponding positive instances to the original sentences. As a result, this process doubles the amount of sentences for TF-IDF, increasing the vocabulary size to $474{,}653$ terms.

Baseline model. We first reproduce SimCSE following the self-supervised pre-training and evaluation setup in Gao et al. (2021). We note that there is a difference in performance in the reproduced result compared to the original one (while maintaining the same seed), specifically from 0.825 (reported) to 0.816 (reproduced) with BERT and from 0.8376 (reported) to 0.8471 (reproduced) with RoBERTa on the STS-B development dataset (Cer et al., 2017). For a more consistent evaluation, we present all augmentation results in comparison to the reproduced version of the unsupervised SimCSE-BERT$_{\text{base}}$ and SimCSE-RoBERTa$_{\text{base}}$ model. We also run experiments with different seeds to establish a more robust difference in performance (see Table D7).

Self-supervised pre-training. For the self-supervised pre-training stage, we train on the same dataset as in the paper that introduced SimCSE (Gao et al., 2021). We use BERT and RoBERTa as our base models and start training with pre-trained checkpoints. To be specific, the uncased BERT model[2] and the cased pre-trained RoBERTa model[3]. The training batch size is set to 64 and the temperature, τ, is equal to 0.05. Dropout is applied to all models, value set to 0.1. The model is trained on the dataset for 1 epoch using the AdamW optimiser (Loshchilov and Hutter, 2019). The initial learning rate of BERT$_{\text{base}}$ models is 3×10^{-5} and decays to 0 with a step-wise linear decay scheduler (applied after each batch). The learning rate for RoBERTa$_{\text{base}}$ models is set constantly to 5×10^{-6} (this improves performance compared to the one reported by Gao et al. (2021)). The radius r (for replacement term sampling based on the maximum TF-IDF score) is set to $4{,}000$ (Table C6 justifies this choice), which is around 1% of the vocabulary size. During the self-supervised pre-training stage, we evaluate the model on the STS-B development dataset every 100 batches. The checkpoint with the best validation result is used for downstream task evaluation.

Downstream task evaluation. The evaluation is based on 7 STS tasks, each containing pairs of sentences with corresponding labelled similarity scores. Details of the datasets can be found in Appendix A. The results are based on the unified setting provided by Gao et al. (2021). First, we freeze the pre-trained network to extract sentence embeddings from the evaluation datasets. These embeddings are obtained from the first layer of the last

[1] The source code will be made available upon acceptance.

[2] Pre-trained BERT model, https://huggingface.co/bert-base-uncased

[3] Pre-trained RoBERTa model, https://huggingface.co/roberta-base

	Augmentation		SentEval - STS tasks							
	paraphrasing	UNA	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS Bench.**	**SICK Rel.**	**Avg.**
BERT	✗	✗	.6904	.7952	.7303	.8019	.7815	.7616	.7116	.7532
	✗	✓	.6802	**.8226**	.7306	.8200	.7908	.7760	.7093	.7614
	✓	✗	**.7003**	.7978	.7531	.8253	.7908	.7856	.7165	.7671
	✓	✓	.6939	.8158	**.7616**	**.8304**	**.7910**	**.8030**	**.7412**	**.7767**
RoBERTa	✗	✗	.6907	.8175	.7336	.8193	.7974	.7996	.6961	.7649
	✗	✓	.7075	.8155	.7343	.8261	.8053	.8021	.6808	.7674
	✓	✗	.6962	.8162	.7442	.8321	.8000	.8128	**.7362**	.7768
	✓	✓	**.7095**	**.8188**	**.7493**	**.8334**	**.8092**	**.8189**	.7352	**.7820**

Table 1: Performance on the SemEval STS tasks (Spearman correlation) when UNA and/or paraphrasing are added with BERT and RoBERTa as the backbone model. Please note that STS Bench. and SICK Rel. abbreviate STS Benchmark and SICK Relatedness, respectively.

hidden state. Next, we compute cosine similarity scores between the embeddings of sentence pairs in the evaluation datasets. Spearman's rank correlation is used to measure the relationship between the predicted similarity scores and the ground truth.

Results. We compare adding UNA to two baselines, SimCSE with $\text{BERT}_{\text{base}}$ and $\text{RoBERTa}_{\text{base}}$. As shown in Table 1, the application of UNA improves the average correlation of the unsupervised SimCSE-$\text{BERT}_{\text{base}}$ from .7532 to .7614, and from .7649 to .7674 on the unsupervised SimCSE-$\text{RoBERTa}_{\text{base}}$. Further, when both paraphrasing and UNA are deployed, correlation increases to .7767 with BERT backbone (an outcome which is robust across different seeds; see Table D7) and further to .7820 with RoBERTa backbone. This is also an improvement over the sole deployment of paraphrasing ($\rho = .7671$ with BERT and $\rho = .7768$ with RoBERTa). The performance on different backbone models points to the same result pattern, demonstrating the effectiveness of UNA with a different backbone language model, and, to some extent, supporting further generalisation claims. We argue that while paraphrasing and UNA both add synthetic data points to the training data, they introduce complimentary invariances to the model. Paraphrasing contributes to recognising sentences with similar semantics, while UNA provides challenging negative signals for separating sentences, and both work in tandem. Interestingly, we reach SOTA performance on the SICK Relatedness task (unsupervised, comparison with other models in Table D8), showcasing that the combination of paraphrasing with UNA can be a straightforward method to improve performance on downstream tasks.

Step 2 Random	Step 3 Random	STS-B dev.	STS Avg.
✓	✓	.7798	.7258
✗	✓	.8086	.7384
✗	✗	**.8235**	**.7614**

Table 2: Ablation study on negative sampling. The first column indicates whether sentence terms are randomly selected to be replaced and the second column whether the replacement term selection is random. The last row is UNA. Performance on the STS-B development set and the average correlation on the 7 STS tasks are shown.

Ablation. The effectiveness of UNA relies on two vital steps: selecting terms with high TF-IDF scores and replacing them with terms that have similar maximum TF-IDF scores in the corpus. Randomly swapping the terms in a sentence does not contribute to generating informative negative samples. To demonstrate this, we conducted an ablation study with $\text{BERT}_{\text{base}}$ model, validating performance on the STS-B development set as well as the average 7 STS tasks. Results are enumerated in Table 2. This highlights that guiding the term selection and replacement process with TF-IDF provides significantly better outcomes than doing either or both steps randomly.

5 Conclusion

In this paper, we propose a novel and efficient negative augmentation strategy, UNA, guided by the TF-IDF retrieval model. Results show that UNA is compatible with different backbone models and could bring great performance improvements to downstream STS tasks, especially when combined with paraphrasing.

6 Limitations

Our experiments focused on English language datasets. Therefore, the findings may not provide constructive directions for other languages with different characteristics compared to English. In addition, the application of TF-IDF on single sentences (as opposed to lengthier text constructs) might provide quite dataset-specific results which may or may not work in favour of the proposed augmentation method (UNA). This is something that follow-up work should investigate.

References

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

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

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.

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

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1597–1607.

Tsz-Him Cheung and Dit-Yan Yeung. 2021. MODALS: Modality-agnostic Automated Data Augmentation in the Latent Space. In *International Conference on Learning Representations*.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218.

Arpita Das, Harish Yenala, Manoj Chinnakotla, and Manish Shrivastava. 2016. Together we stand: Siamese Networks for Similar Question Retrieval. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 378–387.

Hongchao Fang and Pengtao Xie. 2020. CERT: Contrastive Self-supervised Learning for Language Understanding. *arXiv preprint 2005.12766*.

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.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. 2020. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard Negative Mixing for Contrastive Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21798–21809.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754.

Tassilo Klein and Moin Nabi. 2022. SCD: Self-contrastive decorrelation of sentence embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 394–400.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Hans Peter Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive Learning with Hard Negative Samples. In *International Conference on Learning Representations*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Yuxuan Shu, Xiao Gu, Guang-Zhong Yang, and Benny P L Lo. 2022. Revisiting Self-Supervised Contrastive Learning for Facial Expression Recognition. In *33rd British Machine Vision Conference 2022, BMVC 2022*.

Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. 2021. Negative Data Augmentation. In *International Conference on Learning Representations*.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Zilu Tang, Muhammed Yusuf Kocyigit, and Derry Tanti Wijaya. 2022. AugCSE: Contrastive sentence embedding with diverse augmentations. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 375–398.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480.

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.

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*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised Data Augmentation for Consistency Training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268.

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.

Seonghyeon Ye, Jiseon Kim, and Alice Oh. 2021. Efficient Contrastive Learning via Novel Data Augmentation and Curriculum Learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1832–1838.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 11328–11339.

Kun Zhou, Beichen Zhang, Xin Zhao, and Ji-Rong Wen. 2022. Debiased contrastive learning of unsupervised sentence representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6120–6130.

Appendix

A Dataset details

Self-supervised pre-training dataset. All models are pre-trained on the English Wikipedia corpus (1 million sentences) provided by Gao et al. (2021). We note that while we use this training dataset for a consistent comparison with the previous work, it is not entirely suitable for obtaining a TF-IDF representation due to its very short document length, and quite restrictive topic coverage.

STS tasks. The evaluation is based on 7 STS tasks, namely STS 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017), and SICK Relatedness (Marelli et al., 2014), each containing pairs of sentences with their corresponding labelled similarity score ranging from 0 to 5. The size of each evaluation (test) set is enumerated in Table A1.

B Methodology details

We further demonstrate how β controls the augmentation magnitude in UNA. According to Equation 3 and 4, if we set $a_i = z_i - \min(\mathbf{z})$, then

Task	Test samples	Task	Test samples
STS12	3,108	STS16	1,186
STS13	1,500	STS Bench.	1,379
STS14	3,750	SICK Rel.	4,927
STS15	3,000		

Table A1: Size of STS test datasets.

Sentence
Another factor was caffeine. another **instrumental** was **x-45c**.
Greetings from the real universe. **acylations** from the real **government-owned**.
We should play with legos at camp. we **kulish** play with **minimum** at **yakiudon**.

Table B2: UNA-generated sentences. The original sentences are in green while the sentences augmented with UNA are in black. Words that have been replaced by another word are bolded.

$p_i = \min(\beta a_i / C, 1)$. We temporarily ignore the maximum boundary of 1 to simplify the analysis. The mean of p, denoted as $\bar{p}$, can be depicted as:

$$\bar{p} = \frac{1}{n_z} \sum_{i=1}^{n_z} \left(\frac{\beta a_i}{\frac{1}{n_z} \sum_{i=1}^{n_z} a_i} \right) . \quad (5)$$

Given that $\frac{1}{n_z} \sum_{i=1}^{n_z} a_i$ is a constant value for every sentence, we have $\bar{p} = \beta$. When factoring in the upper boundary, this relationship is modified to $\bar{p} \leq \beta$, which suggests that the mean probability of replacing term i in the sentence closely aligns with the augmentation magnitude β.

Examples of sentences generated with UNA are tabulated in Table B2 using the vocabulary of the training dataset described in section A.

C Augmentation details

C.1 Paraphrasing

In our experiments, paraphrasing is applied as an augmentation strategy to create positive samples of the anchor instances. To paraphrase each sentence in the training dataset, we use a paraphrasing model from the Huggingface hub,[4] which is fine-tuned based on the PEGASUS model (Zhang et al., 2020). During pre-training, we consider the original sentences and their corresponding paraphrased

[4]Fine-tuned PEGASUS model for paraphrasing, https://huggingface.co/tuner007/pegasus_paraphrase

Augmentations	SentEval - STS tasks							
	STS12	STS13	STS14	STS15	STS16	STS Bench.	SICK Rel.	Avg.
SimCSE-BERT$_{\text{base}}$ * (reproduced)	.6904	.7952	.7303	.8019	.7815	.7616	.7116	.7532
EDA$^{\spadesuit}$ synonym replacement	.6118	.7106	.6590	.7179	.7424	.6623	.6218	.6751
EDA$^{\spadesuit}$ random insert	.6594	.7402	.7039	.7698	.7543	.7287	**.7200**	.7252
EDA$^{\spadesuit}$ random swap	.6850	.7808	.7212	.7830	.7767	.7643	.6984	.7442
EDA$^{\spadesuit}$ random delete	.6719	.7842	.7147	.7868	.7872	.7641	.6916	.7429
back-translation $_{\texttt{Helsinki}}$ $^{\heartsuit}$	.6876	.7510	.7072	.7933	.7802	.7618	.7034	.7406
paraphrasing $_{\texttt{PEGASUS}}$ $^{\clubsuit}$	.7003	.7978	**.7531**	**.8253**	**.7908**	**.7856**	.7165	**.7671**
MODALS$^{\diamondsuit}$ interpolation	**.7055**	.8137	.7472	.8104	.7852	.7725	.7064	.7630
MODALS$^{\diamondsuit}$ extrapolation	.6817	.7913	.7471	.8136	.7905	.7824	.7124	.7599
MODALS$^{\diamondsuit}$ linear sampling	.6923	.8114	.7381	.8090	.7837	.7745	.6981	.7582
MODALS$^{\diamondsuit}$ Gaussian noise	.6913	**.8143**	.7422	.8095	.7788	.7760	.7180	.7614
EfficientCL$^{\blacklozenge}$ PCA jittering	.6848	.7893	.7426	.8141	.7755	.7854	**.7200**	.7588

Table C3: Performance on STS tasks for different augmentations with BERT as the backbone model. * is the reproduced result of unsupervised SimCSE (Gao et al., 2021). The augmentations presented at the top part of the table are applied directly to sentences while the ones at the bottom part are added in the embedding space. $^{\spadesuit}$ denotes the results after adding the augmentation strategies proposed by Wei and Zou (2019). $^{\heartsuit}$ denotes the results after training with positive sentences generated via back-translation (Sennrich et al., 2016); we first translate the original text to French and then back to English. $^{\clubsuit}$ denotes the result produced by training with positive pairs generated by using a paraphrasing model based on PEGASUS (Zhang et al., 2020). $^{\diamondsuit}$ denotes the results after applying augmentation strategies proposed by Cheung and Yeung (2021). $^{\blacklozenge}$ denotes the result produced by adding perturbation on the embedding layer with PCA jittering proposed by Ye et al. (2021).

ones as positive pairs. Regarding the choice of paraphrasing model, we also assessed T5 by Raffel et al. (2020),[5] but it only reached a correlation of .7985 on the STS-B development dataset (compared to .8234 for PEGASUS).

C.2 Other augmentations

We conducted experiments to assess the performance of various augmentation strategies (explained below) along with contrastive learning. The results are enumerated in Table C3. The strategies listed in the top part of the table are input-level augmentations, *i.e.* they change the input sentence by applying transformations such as deleting or inserting words. The methods at the bottom are embedding-level augmentations. Paraphrasing provides the best results on average.

EDA. Easy Data Augmentation (EDA) (Wei and Zou, 2019) introduced 4 data augmentation strategies: synonym replacement using WordNet (Miller, 1995), random word insertion, random word position swapping, and random word deletion.

Back-translation. Similar to paraphrasing, back-translation is a method that rephrases the sentence using pre-trained language models. It involves translating the sentence to another language and back. In our experiments, we use the pre-trained translation model between English and French by Tiedemann and Thottingal (2020).

MODALS. Cheung and Yeung (2021) proposed the following augmentation strategies at the embedding level: hard example interpolation (interpolate a sample with its closest hard example), hard example extrapolation (use the centre of a set of samples), linear sampling (perturb an embedding along the direction of two random samples), and randomly add Gaussian noise to the embeddings.

PCA jittering. Ye et al. (2021) introduced this augmentation strategy that operates at the embedding level. It adds noise generated based on the application of PCA.

C.3 Results and analysis

Table C3 shows how established augmentation methods perform within a contrastive learning setup. We hypothesise that paraphrasing outperforms all other methods (on average) because it introduces more diverse but reliable positive pairs that assist in capturing useful representations. A drawback of employing pre-trained models for sentence rephrasing lies in the strong dependence on the specific pre-trained weights used. This seems

[5] Fine-tuned T5 model for paraphrasing, https://huggingface.co/Vamsi/T5_Paraphrase_Paws

Algorithm 1 UNA for self-supervised pre-training

Input: A set of N documents, $\mathcal{D}$
Derive the TF-IDF representation of $\mathcal{D}$ and store it in matrix $\mathbf{Z}$ ▷ *as described in section 3*
Store the max TF-IDF score (relevance) of each term in vector $\mathbf{s}$
for every α batches **do** ▷ *self-supervised pre-training*
  for document $j = 1$ to B **do** ▷ *each batch has B documents*
    The TF-IDF vector of document j is given by $\mathbf{z} \subseteq$ row of $\mathbf{Z}$ ▷ *ignore terms not present in document j*
    for $i = 1$ to n_z **do** ▷ *n_z is the number of terms in document j*
      $p_i = \min\left(\beta\left(z_i - \min\left(\mathbf{z}\right)\right)/C, 1\right)$, where $C = \left(1/n_z\right)\sum_{i=1}^{n_z}\left(z_i - \min\left(\mathbf{z}\right)\right)$ for word replacement
      Determine the adjacent set of terms $(\mathcal{A}_i)$ to i within radius r based on the scores in $\mathbf{s}$
      With probability p_i replace term i with a term from $\mathcal{A}_i$; sample the replacement term w.r.t. the scores in $\mathbf{s}$
    end for
    Generate document j' from j after the TF-IDF-driven stochastic term replacement
    Add document j' to the set of UNA's hard negative samples for this batch, $\mathcal{H}_b$
  end for
end for
Output: $\mathcal{H}_b$ every α batches

α	UNA	Paraphrasing & UNA
9	.8213	.8289
7	.8180	.8295
5	**.8235**	**.8345**
3	.7973	.8254
1	.7929	.8230
0	.8161	.8235

Table C4: Performance on the STS-B development dataset (Spearman correlation) for different negative sample injection frequencies (every α training batches) during the training process. $\alpha = 0$ denotes the absence of any form of augmentation.

β	UNA	Paraphrasing & UNA
0.8	.8061	.8328
0.7	.8122	.8354
0.6	.8176	**.8355**
0.5	**.8235**	.8345
0.4	.7968	.8320
0.3	.7798	.8313

Table C5: Performance on the STS-B development dataset (Spearman correlation) for different magnitudes of UNA controlled by hyperparameter β.

r terms	UNA	Paraphrasing & UNA
6,000	.8101	.8319
5,000	.8158	**.8363**
4,000	<u>.8235</u>	<u>.8345</u>
3,000	.8167	.8332
2,000	**.8264**	.8322
1,000	.8073	.8331

Table C6: Performance on the STS-B development dataset (Spearman correlation) for different replacement term radius (r) settings. The best results are bolded and the second-to-best results are underlined.

to be significantly affecting some of the methods, such as back-translation. The embedding-level augmentations might facilitate a better generalisation (as they are not dependent on specific terms), but given that UNA operates on terms (as opposed to embeddings) using them in tandem with UNA did not outperform the paraphrasing and UNA combination (performance not shown).

C.4 Implementation details of UNA

We implement UNA on our reproduced version of SimCSE (Gao et al., 2021) with PyTorch (Paszke et al., 2019). All experiments are conducted on 4 NVIDIA GeForce RTX 2080 Ti GPUs. UNA is applied during the self-supervised pre-training stage to create negative samples that contain similar words or sentence constructions to the anchor sentence. For clarity, we have also included UNA's algorithmic routine in Algorithm 1 (this is also described in section 3).

We apply UNA once every α training batches. The downstream performance is influenced by the frequency (α) of adding generated negative samples. If UNA is applied too frequently, the model may struggle to capture discriminative patterns between random negative samples. To find the optimal frequency, we conduct a grid search with $\alpha = \{1, 3, 5, 7, 9\}$ and evaluate performance on the STS-B development dataset (Table C4). The best performance is obtained by setting $\alpha = 5$, and hence, this is what we use in our experiments.

To determine the augmentation magnitude β, we validate values $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ on the STS-B development set (Table C5). We set $\beta = 0.5$ as this setting yields the greatest average correlation score (across UNA and paraphrasing & UNA).

Another hyperparameter in UNA is the radius r for determining the set of terms in **Step 3**. Per-

Seed	SimCSE	Paraphrasing & UNA
42	.7532	.7767
0	.7526	.7691
1	.7452	.7680
11	.7457	.7655
15	.7488	.7668
48	.7478	.7646
111	.7651	.7726
421	.7395	.7692
456	.7562	.7739
3407	.7623	.7707
Mean (std.)	.7516 (.0079)	.7698 (.0038)

Table D7: Comparison on using different random seeds between SimCSE and SimCSE with the paraphrasing and UNA augmentations across the 7 STS tasks (average Spearman correlation) with $\text{BERT}_{\text{base}}$ model.

formance on the STS-B development set for different values of r is shown in Table C6. The model reaches its peak performance for $r = 5{,}000$ and $2{,}000$ terms with and without paraphrasing, respectively. One potential explanation for this variation could be that the expanded paraphrasing vocabulary has an effect on the radius. We have decided to set $r = 4{,}000$ (with and without paraphrasing) which yields the second-best performance for both augmentation approaches.

D Complementary results

D.1 Random seeds

To examine the robustness of UNA to random seeds, we conducted experiments by pre-training with 10 random seeds on both SimCSE and UNA with paraphrasing. As shown in Table D7, our approach is slightly more robust and yields consistently superior performance.

D.2 Additional comparisons

We present a further comparison with additional pre-trained self-supervised contrastive learning models as tabulated in Table D8. Our method reached competitive results compared to SOTA methods with both backbone models, especially with RoBERTa.

Augmentations	SentEval - STS tasks							
	STS12	**STS13**	**STS14**	**STS15**	**STS16**	**STS Bench.**	**SICK Rel.**	**Avg.**
BERTbase (first-last avg.)	.3970	.5938	.4967	.6603	.6619	.5387	.6206	.5670
BERTbase-whitening	.5783	.6690	.6090	.7508	.7131	.6824	.6373	.6628
SCD-BERT$_{\text{base}}$ (Klein and Nabi, 2022)	.6694	.7803	.6989	.7873	.7623	.7630	.7318	.7419
Mirror-BERT$_{\text{base}}$ (Liu et al., 2021)	.691	.811	.730	.819	.757	.780	.691	.754
DCLR-BERT$_{\text{base}}$ (Zhou et al., 2022)	.7081	.8373	.7511	.8256	.7844	.7831	.7159	.7722
AugCSE-BERT$_{\text{base}}$ (Tang et al., 2022)	.7140	.8393	.7559	.8359	.7961	.7961	.7219	.7798
DiffCSE-BERT$_{\text{base}}$ (Chuang et al., 2022)	**.7228**	**.8443**	**.7647**	**.8390**	**.8054**	**.8059**	.7123	**.7849**
Ours-BERT$_{\text{base}}$	.6939	.8158	.7616	.8304	.7910	.8030	**.7412**	.7767
RoBERTabase (first-last avg.)	.4088	.5874	.4907	.6563	.6148	.5855	.6163	.5657
RoBERTabase-whitening	.4699	.6324	.5723	.7136	.6899	.6136	.6291	.6173
SCD-RoBERTa$_{\text{base}}$ (Klein and Nabi, 2022)	.6353	.7779	.6979	.8021	.7729	.7655	.7210	.7389
Mirror-RoBERTa$_{\text{base}}$ (Liu et al., 2021)	.666	.827	.740	.824	.797	.796	.697	.764
DCLR-RoBERTa$_{\text{base}}$ (Zhou et al., 2022)	.7001	.8308	.7509	**.8366**	.8106	.8186	.7033	.7787
AugCSE-RoBERTa$_{\text{base}}$ (Tang et al., 2022)	.6930	.8217	.7349	.8182	.8140	.8086	.6877	.7683
DiffCSE-RoBERTa$_{\text{base}}$ (Chuang et al., 2022)	.7005	**.8343**	**.7549**	.8281	**.8212**	**.8238**	.7119	**.7821**
Ours-RoBERTa$_{\text{base}}$	**.7095**	.8188	.7493	.8334	.8092	.8189	**.7352**	.7820

Table D8: Comparison with other pre-trained models on the SemEval STS tasks (Spearman correlation) with BERT and RoBERTa as the backbone model. Ours is presented with both UNA and paraphrasing added. The best results are bolded and the second-to-best results are underlined.