VAEGPT-Sim: Improving Sentence Representation with Limited Corpus Using Gradually-Denoising VAE

Zhenyi Wang, Haiyan Ning, Qing Ling, Dan Wang
Ant Group
{wangzhenyi.wzy, ninghaiyan, zhanwen.lq, wd165820}@antgroup.com

Abstract

Text embedding requires a highly efficient method for training domain-specific models on limited data, as general models trained on large corpora lack universal applicability in highly specific fields. Therefore, we have introduced VAEGPT-Sim, an innovative model for generating synonyms that combines a denoising variational autoencoder with a target-specific discriminator to generate synonymous sentences that closely resemble human language. Even when trained with completely unsupervised settings, it maintains a harmonious balance between semantic similarity and lexical diversity, as shown by a comprehensive evaluation metric system with the highest average scores compared to other generative models. When VAEGPT-Sim is utilized as a module for contrastive learning in text representation, it delivers state-of-the-art results in small-dataset training on STS benchmarks, surpassing ConSERT by 2.8 points. This approach optimizes the effectiveness of text representation despite a limited corpus, signifying an advancement in domain-specific embedding technology.

1 Introduction

Text representation is vital for NLP tasks such as clustering, classification, and similarity comparison (Babic et al., 2020). The resource of corpus labeling and the inadequacy of public datasets for certain domains make unsupervised learning with unlabeled data a preferred method. Text embeddings are generated either through word vector methods like Word2Vec, GloVe, and FastText (Mikolov et al., 2013; Pennington et al., 2014; Joulin et al., 2016), or via pre-trained language models such as BERT, InferSent, and Sentence-BERT (Conneau et al., 2017; Reimers and Gurevych, 2019) that capture sentence context. The latter, though, often lack semantic precision due to vector space anisotropy (Ethayarajh, 2019; Li et al., 2020a). Techniques like Bert-flow, whitening, ConSERT, SimCSE, DiffCSE, and InfoCSE have been developed to enhance embeddings by clustering similar sentences and separating dissimilar ones (Li et al., 2020a; Su et al., 2021; Yan et al., 2021; Gao et al., 2021; Chuang et al., 2022; Wu et al., 2022a). However, those top-performing models in STS benchmarks typically require retraining on datasets larger than 1,000,000 sentences.

However, field-specific corpora contain unique specialized words, professional phrases, common expressions, and even distinct text styles. In such real scenarios, it's questioned if a general model trained on a vast corpus can deliver satisfactory results. To address this, we construct a dataset of similar sentences from the biomedical environment domain, using PubMed professional sentences. We then evaluate the text embedding quality of different unsupervised trained BERT_{base} (Table 1).

Model	PubMed similar sentence pairs
SimCSE-BERT$_{base}$ (with 1M wikipedia sentences)	48.37
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (with 70k STS sentences)	45.87
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (with 55k PubMed sentences)	**56.08**

Table 1: Sentence embedding performance of different BERT_{base} on sentence pairs in a special domain (related to biomedicine and environment), evaluated based on "wmean" Spearman's correlation.

As shown above, the vertical model trained with a domain-consistent corpus outperforms the general model trained on a large corpus. While a general model may perform well on public datasets with everyday words and common phrases like STS, it is not a universal solution. The significance of the "special domain" becomes evident in

Findings of the Association for Computational Linguistics ACL 2024, pages 8666–8681

language fields with strong specialization, such as chemistry and biology. Therefore, it is crucial to explore methods for developing highly adapted semantic models for specific domains, with limited resources and minimal specialized data.

As a solution, we propose a novel generation model VAEGPT-Sim for synonymous data augmentation to train a better text representation model with small dataset and limited resource. It combines gradually-denoising VAE and self-judgment loss with the powerful generative model GPT. Therefore, it produces synonym sentences that closely resemble human writing during contrastive learning, keeping semantic similarity and lexical diversity in generation. When we insert VAEGPT-Sim into the framework of comparative learning (generate-CSE) as a core module, it is sufficient to generate rich and high-quality synonyms for the same input sentence among multiple training batches. Compared with BERT2BERT, BART, T5, and GPT2 (Chen et al., 2022; Lewis et al., 2020; Ni et al., 2022; Radford et al., 2019), our VAEGPT-Sim achieves the best result: With training on around 70,000 unlabeled sentences, $\text{BERT}_{\text{base}}$ trained by generate-CSE-VAEGPT-Sim achieves an impressive average wmean Spearman's correlation of 77.29 on 6 STS tasks. It outperforms all previous models trained on similar-scale in-domain datasets, as well as even approaches the performance level of SimCSE, trained on a dataset 14 times larger.

The solution we recommend is aimed at providing an effective solution for real-world industrial applications. First, in practical scenarios, the text corpora for each task vary significantly in domain, specialty, and style, making general models unsuitable. Thus the practical approach is to construct a specialized text representation model for each task based on the limited available data (only the data to be dealt with or historical data from the same source). As a result, by innovatively creating a universal synonym generation model and integrating it into a contrastive learning framework, we enables the rapid construction of specific text representation models with limited in-domain data and resources. Besides, our proposed lightweight synonym generation solution is more suitable for various industrial scenarios, overcoming limitations related to data security and privacy concerns associated with the use of public LLMs, and the high cost of training and running self-made LLMs in business-related fields.

Consequently, VAEGPT-Sim significantly improves BERT's text embedding. This solution is valuable for achieving text representation in resource-limited, data-limited industry settings in numerous specific domains, with a comparable results to models trained on much larger datasets.

2 Related Work

2.1 Contrastive Learning for Text Embedding

Contrastive learning, specifically NT-Xent, enhances semantic diversity in language models by aligning similar sentences and distinguishing unrelated ones. Inspired by computer vision research and exemplified by SimCLR, this learning framework has achieved remarkable results in image classification (Chen et al., 2020). For instance, CERT generates positive pairs through back translation (Fang and Xie, 2020), ConSERT uses token shuffling, cutoff, and dropout for synonymous sentence creation (Yan et al., 2021), and DeCLUTR combines a span definition and MLM loss in contrastive learning (Giorgi et al., 2021). After SimCSE achieved impressive performance with a simple dropout framework (Gao et al., 2021), ESimCSE enhanced SimCSE with the momentum contrast method to excel in STS tasks (Wu et al., 2022b). Furthermore, DiffCSE improved embedding effectiveness with a difference prediction objective (Chuang et al., 2022), and InfoCSE enhanced SimCSE by incorporating an additional masked language model task (Wu et al., 2022a). Those "CSE-series" models consistently surpass existing benchmarks in STS tasks, yet they all rely on extensive amounts of data (about 1M).

2.2 Adversarial and Denoising Techniques in Text Generation Models

Adversarial models like GAN, Seq-GAN, TextGAN, LeakGAN, and RelGAN improve generator performance through discriminator-generator interplay (Goodfellow et al., 2020; Yu et al., 2017; Zhang et al., 2017; Guo et al., 2018; Nie et al., 2019). They overcome non-differentiability challenges with techniques such as Gumble-softmax and policy gradients (Kusner and Hernández-Lobato, 2016; Yu et al., 2017). These models enhance diversity by modifying architectures and evaluation approaches (Zhang et al., 2017; Guo et al., 2018; Nie et al., 2019).

Variational Autoencoders (VAE) and diffusion models offer gradual noise recovery from a stan-

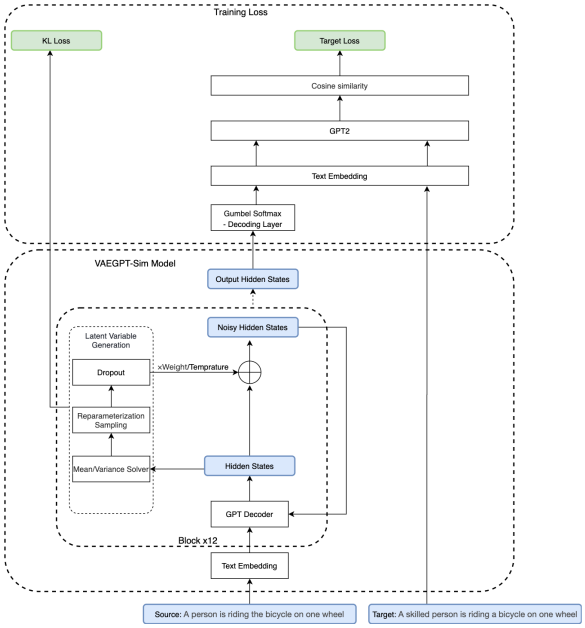

Figure 1: Illustration of VAEGPT-Sim model.

dard distribution (Kingma and Welling, 2014; Ho et al., 2020; Dhariwal and Nichol, 2021). VAE generates desired outputs by removing noise sampled from latent features, while diffusion models break down generated targets into noise gradually. VAE has been widely used, including models like DEM-VAE and OPTIMUS for improved control and integration with pre-trained models (Shi et al., 2020; Li et al., 2020b). Diffusion models, though challenging in generating discrete text sequences, is effective to enhance diversity by sampling and recovering noise (Reid et al., 2022).

These advancements have greatly enhanced text generation, enabling creative and diverse language production.

3 Approach

3.1 Synonym Generation Model VAEGPT-Sim

We enhance pretrained GPT2 by introducing a structured approach, resulting in improved synonym generation. The VAEGPT-Sim consists of two parts (Figure 1): the generation model itself and a discriminator for loss calculation during training. In VAEGPT-Sim, we preserve the main structure of GPT2, which includes 12 decoders with the same structure but different parameters. Then we introduce a structure for latent variable generation within each decoder block. This structure includes a solver for mean/variance calculation, a reparameterization sampler, and an additional dropout layer for distribution prediction and Gaussian noise sampling. This generates a set of random noises following an exclusive normal distribution for the current hidden states. These noises are then added to the original hidden state vector of the layer with varying ratios, calculated using the following formula:

$$h_i = h_i + l_i \times \frac{W}{e^{i+1}}$$

In the formula, h_i represents the hidden state vector of the i^{th} decoder, l_i represents the generated latent variable from this layer. A weight constant W (W = 0.1) and a temperature coefficient of e^{i+1} are also used. The noisy hidden state vector, obtained by adding the random noise, serves as the input for the next decoder block. For each block, the random noise is sampled from the core feature of the input sentence analyzed in the current block. This ensures that random components are added based on the key information of the current decoder. As the output approaches, the noise gradually diminishes due to the temperature coefficient, as the noise should be eliminated gradually.

After 12 decoding layers, the final output hidden state vector is decoded using the Gumbel Softmax method (Kusner and Hernández-Lobato, 2016) to obtain a predicted sequence of vocabulary IDs. This predicted sequence is evaluated using a weight-frozen GPT2 (utilizing the original pretrained GPT2 checkpoint without any fine-tuning), which provides embedding vectors for each sentence. The cosine similarity between the embedding vectors ($V_{predict}$: vector of the sentences generated by the model, and V_{target}: vector of the target synonymous sentence) is then calculated to determine the target loss. The final training loss is the sum of the KL divergence loss in 12 rounds of latent variable generation (weighted by W_{KL}, set to 0.0001 in this article) and the target loss:

$$\mathcal{L} = \mathcal{L}_{target} + \mathcal{L}_{KL} \times W_{KL}$$

$$= 1 - \frac{\sum_{i=1}^{N} sim(V_{predict}, V_{target})_{norm}}{N}$$

$$+ \frac{\sum_{i=1}^{N} \sum_{j=1}^{12} KL(N(\mu_j, \sigma_j^2) || N(0, I))}{N} \times W_{KL}$$

Here, N represents the batch size during the training process. Thus, this loss ensures both the randomness of noise sampling through core features

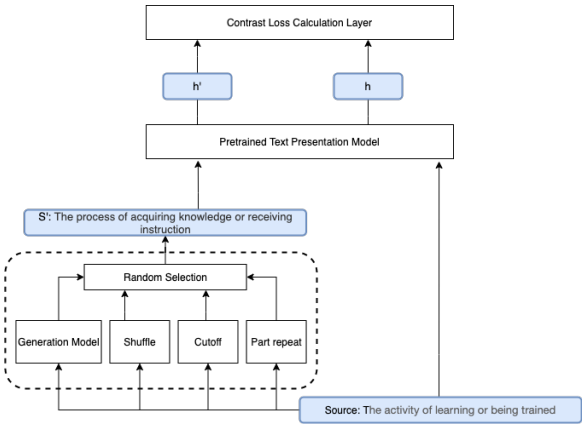

Figure 2: Illustration of Generate-CSE training frame.

and the semantic similarity of the generated sentences. Additionally, batch normalization is applied to μ and σ to prevent KL vanishing (Zhu et al., 2020).

During training, the discriminator structure guides the model to keep the semantic similarity instead of lexical similarity. However, during the prediction process, the discriminator can be removed, and the hidden state vector of the generation model itself is used as the output.

3.2 Contrastive Learning Frame Generate-CSE

We introduce a contrastive learning framework, Generate-CSE, to enhance the text representation model through generation models. In Figure 2, this approach involves preparing two inputs - the source sentence and the generated synonymous sentence - for the text representation model. Their final embeddings are compared using cosine similarity. However, in the process of generating synonymous sentences, we incorporate a Random Selection module. For each input sentence, this module generates a random number to choose a generation method (shuffle, cutoff, partial repetition, and generation model) to obtain the synonymous partner. The generation model here can be replaced with any synonymous sentence generation model, while employing VAEGPT-Sim as this module is the recommended approach in this paper. Detailed introduction to other modules refer to Appendix A.1. As a result, each batch contains sentence pairs with varying degrees of similarity, and in different epochs, every sentence has different pairs with diverse similarity levels. This ensures that the model learns a wide range of synonymous writing styles.

The contrastive loss used in this framework is adopted from previous studies (Yan et al., 2021; Gao et al., 2021). In a batch of N sentences, the generated sentences and source sentences form a set of 2N. This loss encourages a larger cosine similarity between synonymous sentence pairs and gradually reduces the semantic similarity among other combinations of sentences. A temperature hyperparameter τ (set as 0.05) controls the strength of this effect.

$$\mathcal{L}_{contrast} = -log \frac{e^{(sim(h_i,h_i'))/\tau}}{\sum_{j=1}^{2N} e^{(sim(h_i,h_j))/\tau}(j \neq i)}$$

4 Experiments

4.1 Setup for Generation Model

4.1.1 Fine-tune Setup

We conduct our experiments with 5 generation models: BERT2BERT, BART, T5, GPT2, and VAEGPT-Sim. Based on the official checkpoint downloaded from Huggingface (VAEGPT-Sim loads the GPT2 checkpoint), we continue to fine-tune those model: Two versions of the training dataset are constructed by collecting sentences with a label greater than 3 from SICK (9.4k pairs) (Marelli et al., 2014) and a label equal to 1 from MRPC (5.8k pairs) (Dolan and Brockett, 2005). The first version retains 11.4k pairs of sentences, with one sentence serving as the source and the other in the same pair as the target. Besides, we also create the **fully unsupervised version** by completely shuffling sentences: During training, we take one sentence as the source and use itself as the target. The datasets are randomly split into training and validation sets (in a 12:1 ratio). Please note that in our recommended approach, the generation model is always **fine-tuned only once** using the above dataset. Therefore, the resource consumption is completely fixed and limited, with no need for in-domain data related to downstream tasks.

4.1.2 Evaluation Method

For testing, we use the SICK-trail dataset with a label greater than 3 (0.5k pairs) to check the synonym generation effects (see Appendix for details of decoding method). The evaluation metrics used in this paper can be classified into two types. **Similarity to the target sentence**, covers both lexical and semantic aspects. The Rouge metrics, based on Rouge 1.0.1, are used to assess lexical similarity (Lin, 2004). The final evaluation metric for

this aspect is the Avg-Rouge, which is the average of Rouge-1, Rouge-2, and Rouge-L. To evaluate semantic similarity, two Cosine Similarity scores are computed using Glove-based Vector Mean (Pennington et al., 2014) and all-MiniLM-L6-v2 recommended by Sentence-BERT (Reimers and Gurevych, 2019). **Diversity in generation**, encompasses lexical distinctiveness in multiple rounds of generation as well as within a single round. We compute the proportion of non-repeating words to measure Word Diversity within the generated sentences and original sentences, as well as among the sentences generated in different rounds. Similarly, the Type-Token Ratio (TTR) is used to assess lexical non-repetition in single generation quality (Johnson, 1944). For a **comprehensive evaluation**, we calculate the overall average score of all the individual metrics mentioned above, as well as the deficiency score, which is determined as the minimum value between the average scores of the 3 diversity metrics and the 3 similarity metrics.

4.2 Setup for Text Representation Model

4.2.1 Training Setup

We utilize the public checkpoints of BERT$_{base}$ and BERT$_{large}$ for our experiments, fine-tuning them with 4 datasets for various tasks (Appendix A10). Most experiments are conducted on six English STS datasets: STS2012-2016(Agirre et al., 2012, 2013, 2014, 2015, 2016) and STS Benchmark (Cer et al., 2017). To prevent potential target leakage, we exclude the SICK dataset as it is already used in the generation model training. We randomly shuffle 70.4k single sentences from the target dataset, removing their labels and pair relations, and use them for the training dataset. This method aligns with previous studies on small dataset training in this area (Yan et al., 2021), allowing for fair comparisons. Additionally, to underscore the significance of the "special domain" in the Introduction, we train another BERT$_{base}$ using 55k single sentences randomly extracted from PubMedQA. Furthermore, in the Ablation Study, we randomly extract 70k single sentences from NLI to create an unsupervised small corpus that is entirely unrelated to the STS benchmark to train a new BERT$_{base}$. Additionally, to ensure a fair comparison, we further fine-tune the SimCSE BERT$_{base}$ using the 11.4k sentence pairs mentioned in 4.1.1 directly as positive samples for contrastive learning.

At the same time, we find that when using our training method, adopting a cls vector with an added mlp layer during the training phase and utilizing the last-pooling vector as the final outcome for evaluations and applications after training yields the best result. Consequently, this approach is employed in the majority of experiments related to Generate-CSE-VAEGPT-Sim unless specifically stated otherwise.

4.2.2 Evaluation Method

We evaluate the text representation models trained using different methods by measuring Spearman's correlation on the previously mentioned STS tasks and report averages from 3 repeated iterations. In Ablation Studies, we compare the impact of different generation modes and output vectors while keeping other training and evaluation conditions consistent. Additionally, we perform an experiment for retrieval task, as mentioned in (Chuang et al., 2022), using all sentences in the STS-B-test dataset as the search scope. We select 97 positive pairs (with label = 5) and retrieve one sentence to measure the ranking of the other sentence, assessing whether it appears in the top-1/5/10 positions, calculating the recall rate @1/5/10. Furthermore, for the evaluation of PubMed BERT$_{base}$, we randomly extract 56 single sentences from PubMedQA and request ChatGPT to rewrite the sentences with varying degrees of similarity and provide a similarity score as the test dataset.

4.3 Results

4.3.1 Synonymous Generation Quality

In the task of synonym generation, our goal is for the generation model to exhibit both semantic and lexical similarity to the target sentence while upholding diversity in its output, as opposed to simply regurgitating the input. Thus, the ideal generative model should achieve a harmonious balance across the aforementioned evaluation metrics without displaying significant deficiencies in any area of similarity or diversity. This is reflected in the highest average and deficiency scores in Table 2. Following training, our VAEGPT-Sim yields the best comprehensive evaluation scores. Notably, in a completely unsupervised training mode, BERT2BERT and GPT2 struggle to maintain the coherence of the generated sentences, while BART and T5 have completely forfeited generative diversity, merely replicating the original sentence. Conversely, VAEGPT-Sim has the adversarial structure of sentence meaning evaluation and the diversity-

Model	Similarity to the Target Sentence			Diversity in Generation			Comprehensive Evaluation	
	Avg-Rouge	Cosine Similarity (Vector Mean)	Cosine Similarity (Sentence-BERT)	Word Diversity on Source and Generation	Word Diversity on Multiple Generations	TTR	Average Score	Deficiency Score
BERT2BERT (without fine-tune)	53.41	56.75	-0.07	99.61	96.90	91.67	66.38	36.70
BART (without fine-tune)	78.63	95.98	74.35	2.40	1.99	96.27	58.27	33.55
T5 (without fine-tune)	73.05	95.33	66.79	42.08	31.63	60.44	61.55	44.72
GPT2 (without fine-tune)	51.33	91.09	27.63	91.57	80.65	62.62	67.48	56.68
BERT2BERT	51.30	37.02	0.92	99.90	98.39	98.66	64.37	29.75
BART	80.57	96.27	77.36	9.39	2.79	97.17	60.59	36.45
T5	80.39	96.27	77.43	14.58	3.94	97.10	61.62	38.54
GPT2	65.37	94.51	44.31	81.62	64.25	31.59	63.61	59.15
VAEGPT-Sim	77.47	93.74	62.60	60.95	34.43	96.00	**70.86**	**63.79**
BERT2BERT (unsupervised fine-tune)	44.37	90.30	0.52	89.95	58.93	20.34	50.74	45.07
BART (unsupervised fine-tune)	79.28	96.08	75.86	0.02	0.01	96.91	58.03	32.31
T5 (unsupervised fine-tune)	79.28	96.08	75.88	0.09	0.01	96.93	58.05	32.34
GPT2 (unsupervised fine-tune)	61.40	93.53	44.32	78.27	67.15	31.28	62.66	58.90
VAEGPT-Sim (unsupervised fine-tune)	76.76	92.82	68.28	53.79	27.70	95.42	**69.13**	**58.97**

Table 2: Generation quality of models on synonymous generation task, evaluated by the positive pairs in SICK-trail dataset (with a label > 3). Every indicator is multiplied by 100.

promoting structure of VAE. Therefore, even when trained with the original sentence as the target, it can maintain the semantic similarity while ensuring a certain level of generation diversity.

Table A2 (see Appendix) provides practical examples that deepen our understanding of the evaluation metrics. BART, designed for text summarization, often generates sentences identical to the original when applied to our single-sentence data. Fine-tuning with a small dataset does little to change its style. T5 initially produces unsatisfactory results, with repetitive and incoherent phrases. However, after fine-tuning, T5 demonstrates limited synonym rewriting ability, with four out of six sentences showing minor changes. GPT2, a continuous writing model, excels in producing fluent and expressive text. However, its focus deviates too far from synonymous generation, and simple fine-tuning fails to effectively redirect its generation. This results in sentences that are not true synonyms, such as changing "man" to "man and woman." Therefore, achieving the highest Average Score for similarity and diversity, along with a Deficiency Score indicating no significant shortcomings, makes it the optimal model for synonym generation. It consistently generates diverse and human-aligned synonymous sentences.

4.3.2 Text Representation Quality

Above generation models are used to construct Generate-CSE for fine-tuning BERT and obtaining sentence embeddings (Table 3). The quality of synonym generation directly affects the performance of the corresponding text representation model. BERT2BERT performs the worst, while including Random Selection improves its effectiveness because its generation quality is quite poor. BART and T5 have similar effects, with Random Selection providing slight improvements. GPT2 performs better than BERT2BERT but is still inferior to BART and T5. Compared with them, our VAEGPT-Sim achieves the best generation quality.

When comparing the performance of our Generate-CSE-VAEGPT-Sim (combined with VAEGPT-Sim and Random Selection) with previous small-dataset training studies (Table 4), Generate-CSE-VAEGPT-Sim-BERT$_{base}$ trained on 70k sentences outperforms BERT$_{base}$-flow-target, BERT$_{base}$-whitening-target, and ConSERT-BERT$_{base}$ trained similar in-domain sentences (around 90k, including 6 STS and SICK). Our method excels in small dataset training for text representation, even surpassing general models trained on very large-scale dataset in highly specific domains (Table 1).

Model	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	Avg.
BERT$_{large}$ (only with BERT2BERT)	56.84	65.52	61.33	67.34	71.90	62.72	64.28
BERT$_{large}$ (with BERT2BERT and Random Selection)	64.29	75.45	70.79	75.18	72.99	71.61	71.72
BERT$_{large}$ (only with BART)	66.65	78.86	73.70	77.86	79.10	77.27	75.57
BERT$_{large}$ (with BART and Random Selection)	67.31	79.49	73.86	78.92	79.34	77.96	76.15
BERT$_{large}$ (only with T5)	65.53	77.63	72.69	77.39	78.30	75.84	74.56
BERT$_{large}$ (with T5 and Random Selection)	68.07	79.61	73.90	79.55	79.20	78.11	76.41
BERT$_{large}$ (only with GPT2)	65.36	76.28	71.92	76.26	77.42	76.32	73.93
BERT$_{large}$ (with GPT2 and Random Selection)	67.55	78.69	73.66	78.59	78.63	78.12	75.87
BERT$_{large}$ (only with VAEGPT-Sim)	67.76	79.11	74.95	78.73	76.44	77.18	75.70
BERT$_{large}$ (with VAEGPT-Sim and Random Selection)	**68.82**	**80.18**	**75.05**	**80.46**	**80.28**	**79.46**	**77.38**

Table 3: Sentence embedding performance of BERT$_{large}$ on 6 STS dataset, evaluated based on "wmean" Spearman's correlation. BERT$_{large}$ is fine-tuned with different generation model and "Random Selection" (using the Random Selection module to randomly select the synonymous sentence from Shuffle, Cutoff, Partial Repetition, and a certain generation model, as showed in Figure 2).

Model	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	Avg.
GloVe embeddings (Mean Vector) ♢	55.14	70.66	59.73	68.25	63.66	58.02	62.58
BERT$_{base}$ (first-last-avg) ♣	57.86	61.97	62.49	70.96	69.76	59.04	63.68
BERT$_{base}$-flow-target △	63.48	72.14	68.42	73.77	75.37	70.72	70.65
BERT$_{base}$-whitening-target ♣	63.62	73.02	69.23	74.52	72.15	71.34	70.65
ConSERT-BERT$_{base}$ ♠	64.64	**78.49**	69.07	**79.72**	75.95	73.97	73.64
Generate-CSE-VAEGPT-Sim-BERT$_{base}$	**68.54**	77.96	**74.52**	79.50	**79.87**	**78.15**	**76.42**

Table 4: Sentence embedding performance of BERT$_{base}$ on 6 STS dataset, evaluated based on "wmean" Spearman's correlation. This Generate-CSE-BERT$_{base}$ is trained with VAEGPT-Sim and Random Selection. The results of other models are from the original literatures: △: (Li et al., 2020a), ♣: (Su et al., 2021), ♠: (Yan et al., 2021), ♢: (Gao et al., 2021).

4.4 Ablation Studies and Analysis

4.4.1 Different Generation Methods for Positive Pairs

In contrastive learning, recent advancements in large datasets (1M or even larger) have led to state-of-the-art performance, such as SimCSE. Therefore, when introducing our method, which can be trained quickly with a very small dataset, we need to examine the impact of dataset reduction on the success of positive sample generation methods.

As shown in Table 5 and A3, SimCSE-BERT$_{base}$ achieves a correlation of 77.17 with 1M training sentences, but it drops to 73.14 when trained on 70k sentences (although target-related). The impact of dataset size is significant, especially in a benchmark with non-professional daily English. When working with the same limited in-domain training data, our VAEGPT-Sim achieves the best results.

Model	Avg. STS
SimCSE-BERT$_{base}$ (cls) (1M data) ♢	77.17
BERT$_{base}$-Cutoff+Shuffle (last-pool) (70k data)	71.55
BERT$_{base}$-dropout (SimCSE method) (last-pool) (70k data)	73.14
BERT$_{base}$-dropout (SimCSE method) (cls) (70k data)	71.69
BERT$_{base}$-random repetition (last-pool) (70k data)	74.73
BERT$_{base}$-random repetition (cls) (70k data)	73.53
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (last-pool) (70k data)	**76.42**

Table 5: Influence of different generation methods of positive pairs with the same 70k training dataset (mixed 6 STS datasets).

4.4.2 Contribution of Training Dataset

To ensure a fair comparison with previous models, we conduct an ablation experiment to assess the influence of training datasets, as shown in Table 6 and A4. Initially, we utilize sentence pairs from MRPC and SICK in the generation model training, but when we use the same sentence pairs as positive samples to fine-tune SimCSE for comparison, we find a significant decline in results. This finding demonstrates that those pairs cannot be seen as reliable positive samples with specific labels for the text representation task, but rather as loosely similar sentence pairs with no strict standard. As a result, their working mechanism on the generation model can be inferred: Combined with the adversarial structure based on the soft criterion of text similarity, they guide the generation model

Model	Avg. STS
SimCSE-BERT$_{base}$ (joint-trained with 11.4k MRPC/SICK pairs and 1M Wikipedia sentences) ♢	73.24
SimCSE-BERT$_{base}$ (with 1M Wikipedia sentences, then fine-tuned with 11.4k MRPC/SICK pairs)	74.44
ConSERT-BERT$_{base}$ (with 90k SICK/STS sentences) ♠	73.64
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (with 70k STS sentences, VAEGPT-Sim with 11.4k MRPC/SICK pairs)	**76.42**
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (with 70k STS sentences, VAEGPT-Sim with 11.8k MRPC/SICK sentences)	74.90
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (with 70k NLI sentences)	74.00
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (with 70k NLI sentences, VAEGPT-Sim with 11.8k MRPC/SICK sentences)	72.98

Table 6: Ablation studies of the influence of different training dataset at each stage, based on "wmean" Spearman's correlation of 6 STS dataset.

to change its output form into a rewriting style from abstract or continuation style. Meanwhile, VAEGPT-Sim benefits from pairs with differences between input and target in terms of generation diversity (Table 2). Thus, when using VAEGPT-Sim trained with fully unsupervised data, the training effectiveness slightly decreases after losing this diversity in generation, but it still exceeds methods trained with similar target-related data, such as ConSERT. Furthermore, training BERT with VAEGPT-Sim and 70k target-unrelated sentences still yields higher results than ConSERT, even competing with SimCSE (74.00 vs. 73.24 and 74.44), despite the training data differing by a magnitude of 14 times. This highlights the effectiveness of this method in small-dataset tasks.

Model (trained with 70k NLI sentences)	Avg. STS
SimCSE-BERT$_{base}$	67.21
DiffCSE-BERT$_{base}$	69.62
InfoCSE-BERT$_{base}$	71.65
ESimCSE-BERT$_{base}$	72.17
Generate-CSE-VAEGPT-Sim-BERT$_{base}$	**72.98**

Table 7: Ablation studies comparing the performance of various training methods for text representation models using the same dataset of 70k NLI sentences, measured by "wmean" Spearman's correlation of 6 STS dataset.

Another experiment compares the performance of various contrastive learning training methods that have been introduced in recent years, all trained on the same small sample dataset with their officially recommended settings (Table 7 and A5). Under completely fair conditions, using only 70,000 unsupervised sentences, our BERT with VAEGPT-Sim method outperforms all others, including InfoCSE, ESimCSE, etc., and continues to achieve the best results. This demonstrates that our method is particularly effective in small sample training and more cost-efficient at securing better outcomes.

4.4.3 Vector Choice for Text Representation

Model	Avg. STS
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (cls)	76.69
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (last2avg)	76.88
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (first-last-avg)	76.91
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (whitening)	71.92
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (last-pool)	**77.38**

Table 8: Sentence embedding performance of different vectors from BERT$_{large}$ on 6 STS dataset, evaluated based on "wmean" Spearman's correlation.

Furthermore, we conduct an ablation study on different methods for obtaining text representation vectors. We find that for Generate-CSE-VAEGPT-Sim, the "last-pool" vector performs the best compared to the cls vector, last2avg vector, first-last-avg, and first-last-avg with whitening (Table 8). More detailed analysis can be found in the Appendix (Table A6).

4.4.4 Distribution of Sentence Embeddings

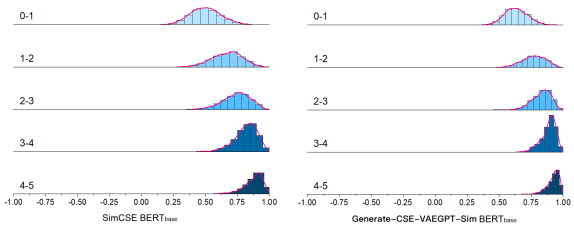

Figure 3: Distribution plots of cosine similarities between sentence pairs in STS-B. Pairs are divided into 5 groups by the original labels and x-axis shows its cosine similarity calculated from the text representation model.

The representation space of the Generate-CSE-VAEGPT-Sim is visualised by plotting the cosine similarities (Figure 3). Our BERT$_{base}$ exhibits a more concentrated cosine similarity distribution within each manually labeled segment, with a narrower range and higher kurtosis compared to SimCSE BERT$_{base}$. Particularly, Generate-CSE-VAEGPT-Sim BERT$_{base}$ outperforms in evaluating segments 3-4 and 4-5. Therefore, our method help to make better judgments on synonymous sentences by effectively learning a diverse range of them.

4.4.5 Retrieval Task

The performance of Generate-CSE-VAEGPT-Sim BERT$_{base}$ in the retrieval task is being evaluated using STS-B-test. Table 9 and A7 demonstrate

Model/Recall	@1	@5	@10
SimCSE BERT$_{base}$	76.17	94.97	**98.40**
Generate-CSE-VAEGPT-Sim-BERT$_{base}$	**78.15**	**95.64**	97.99

Table 9: The retrieval results for SimCSE and Generate-CSE-VAEGPT-Sim.

that BERT$_{base}$ trained with VAEGPT-Sim effectively identifies the most suitable proximal sentence. Compared to SimCSE BERT$_{base}$, there is a significantly higher likelihood of the most semantically relevant sentences being recalled in the first position, as well as in the first five positions. This further validates the effectiveness of VAEGPT-Sim in creating synonyms with limited data.

5 Conclusion

To address the industry's need for small-scale data training in specific domain text embedding models, we propose an innovative generation model VAEGPT-Sim to obtain diverse positive synonymous sentence pairs in contrastive learning process for text representation model. BERT trained with VAEGPT-Sim surpasses models trained on similar-scale data and achieves remarkable performance in sentence embedding on STS tasks. It can even rival larger-scale models like SimCSE.

6 Limitations

Focusing on demonstrating effectiveness in small dataset training environments similar to practical industrial applications, we have not conducted experiments in other scenarios. For example, we have only used basic contrastive learning loss, without further exploration after combining with MLM or supervised settings. Additionally, as shown in 4.4.4, the cosine similarities of segment 0-1 from our BERT are slightly higher, indicating a limited ability of our method to distinguish negative pairs due to the constrained size of the training dataset. This shortcoming may be magnified when trained with rich and diverse positive samples.

Acknowledgements

This research was supported by Ant Group. We extend our sincere gratitude to the anonymous reviewers for their thoughtful suggestions and comments, which have greatly improved the quality of this paper.

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, Denver, Colorado. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

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, Atlanta, Georgia, USA. Association for Computational Linguistics.

Eneko Agirre, Aitor Gonzalez-Agirre, Iñigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2016. SemEval-2016 task 2: Interpretable semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 512–524, San Diego, California. Association for Computational Linguistics.

Karlo Babic, Sanda Martincic-Ipsic, and Ana Mestrovic. 2020. Survey of neural text representation models. *Inf.*, 11(11):511.

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, Vancouver, Canada. Association for Computational Linguistics.

Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen,

Zhiyuan Liu, and Qun Liu. 2022. bert2BERT: Towards reusable pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2134–2148, Dublin, Ireland. Association for Computational Linguistics.

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*, ICML'20. JMLR.org.

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, Seattle, United States. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 670–680. Association for Computational Linguistics.

Prafulla Dhariwal and Alexander Quinn Nichol. 2021. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8780–8794.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 55–65. Association for Computational Linguistics.

Hongchao Fang and Pengtao Xie. 2020. CERT: contrastive self-supervised learning for language understanding. *CoRR*, abs/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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Wang Bo, and Gary Bader. 2021. Declutr: Deep contrastive learning for unsupervised textual representations. pages 879–895.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM*, 63(11):139–144.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Wendell Johnson. 1944. Studies in language behavior: A program of research. In *Psychological Monographs*, volume 56, page 1–15.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Matt J. Kusner and José Miguel Hernández-Lobato. 2016. GANS for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. 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, Online. Association for Computational Linguistics.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020b. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

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

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Weili Nie, Nina Narodytska, and Ankit Patel. 2019. Relgan: Relational generative adversarial networks for text generation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Machel Reid, Vincent J. Hellendoorn, and Graham Neubig. 2022. Diffuser: Discrete diffusion via edit-based reconstruction.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Wenxian Shi, Hao Zhou, Ning Miao, and Lei Li. 2020. Dispersed exponential family mixture VAEs for interpretable text generation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8840–8851. PMLR.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316.

Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022a. InfoCSE: Information-aggregated contrastive learning of sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3060–3070, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022b. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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, Online. Association for Computational Linguistics.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2852–2858. AAAI Press.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *International Conference on Machine Learning*.

Qile Zhu, Wei Bi, Xiaojiang Liu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. 2020. A batch normalized inference network keeps the KL vanishing away. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2636–2649. Association for Computational Linguistics.

A Appendix

A.1 Environments and Training Details

We use the NVIDIA A100 GPU for each experiment. For the training process of generation model, we use the grid-search of batch size $\in \{32, 64\}$,

learning rate $\in \{2e-3, 5e-6, 5e-7\}$, hyperparameter W $\in \{0.05, 0.1, 0.5, 1.0\}$, $W_{KL} \in \{0.0001, 0.001, 0.01\}$. While saving the checkpoint for the best average Rouge scores with every combination of hyperparameters, we choose the suitable setting (according to the training process of VAEGPT-Sim). Beside, for the training process of the constrastive learning, we use the grid-search of batch size $\in \{32, 64, 96\}$, learning rate $\in \{1e-6, 2e-6, 5e-7\}$, and the temperature $\tau \in \{0.05, 0.1\}$.

hyperparameter	Generation Model	Text Representation Model
learning rate	5E-07	2E-06
batch size	32	96
W	0.1	/
W_{KL}	0.0001	/
temperature	/	0.05

Table A1. The main hyperparemeters for the expriments.

Finally, we choose the best setting according to the highest average Spearman's score on 6 STS tasks (based on the training process of BERT$_{large}$). The best settings are listed in Table A1.

Meanwhile, in this paper, the decoding settings for all generation models are consistent. We have uniformly employed a conventional Beam Search with Sampling method, generating a set number of candidates (num-beams = 3), implementing n-gram blocking (no-repeat-ngram = 4), and using a sampling method to generate the candidates. As a result, all generation model comparisons in this article solely reflect the performance of the models themselves, without any influence from variations in decoding settings.

Additionally, the details of four generation modules within Generate-CSE, excluding the generation model itself, are as follows: The shuffle method introduces a word-level random permutation to a sentence; cutoff is applied to sentences with more than three words (those with three or fewer words remain unchanged), where a random number N is drawn from the range [1, min(6, Words(sentence)/3)], indicating the total number of words to be deleted, and then N positions are randomly selected within the sentence for word-level deletions. Dropout randomly masks tokens at a rate of 0.1. The random repeat method randomly selects a repeat count N from the range [1, max(1, int((Words(sentence)-1)*0.3))] and then randomly selects N positions from the sentence for word-level replication (copy the word from the original position and insert it back into the original spot).

A.2 Manual Evaluation for Generation Examples of Models

Table A2 displays the generation results of 7 models on 6 randomly selected sentences from the test dataset (see the last page), providing a glimpse into the quality of the generation process. For a comprehensive evaluation of synonymous generation, please refer to the detailed assessment in the manuscript.

A.3 Different Generation Methods of Positive Pairs

We conduct ablation studies to evaluate the effects of different generation methods for positive pairs, all using the same 70k training dataset. In addition to our VAEGPT-Sim, we compare three other approaches: "Cutoff and Shuffle" proposed by ConSERT, "dropout" proposed by SimCSE, and random repetition proposed by ESimCSE. The result is shown in Table A3, on the small dataset, the average Spearman's correlations of the three alternative methods all drop below 75, when our VAEGPT-Sim help to achieve an average score of 76.42. Clearly, when working with the same limited in-domain training data, our Generate-CSE-VAEGPT-Sim achieves the best results.

A.4 Influence of training dataset

In Table A3, we demonstrate that when the 1M training dataset is reduced to a 70k dataset, the dropout method cannot produce ideal results even when using the target sentences. In Table A4, we continue to show the influence of different training datasets on the final result. In an attempt to compare the different models in a fair condition, based on the checkpoint of SimCSE BERT$_{base}$ uploaded by the authors, we continue to fine-tune it with the SICK and MRPC pairs as the positive pairs, which we use to train our generation model. However, the results show that, whether we directly fine-tune it with those "supervised" pairs or joint-fine-tune it with the mixture of SICK and MRPC pairs and Wikipedia sentences (also the original Wikipedia dataset published by (Gao et al., 2021)), although we all use the recommended settings (learning rate = 3e-5, temperature hyperparameter = 0.05), it still leads to a decline in the effect. The results show that the pairs we use to train our generation model cannot be equated to the "positive pairs" of text similarity evaluation. When we compare our

Model	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	Avg.
SimCSE-BERT$_{base}$ (cls) (1M data) ♢	70.14	79.56	75.91	81.46	79.07	76.85	77.17
BERT$_{base}$-Cutoff+Shuffle (last-pool) (70k data)	63.81	74.15	70.75	74.75	73.12	72.72	71.55
BERT$_{base}$-dropout (SimCSE method) (last-pool) (70k data)	63.68	75.94	70.90	77.30	77.04	73.98	73.14
BERT$_{base}$-dropout (SimCSE method) (cls) (70k data)	62.06	74.65	69.64	75.88	75.64	72.24	71.69
BERT$_{base}$-random repetition (last-pool) (70k data)	65.50	76.96	73.32	78.31	78.27	76.05	74.73
BERT$_{base}$-random repetition (cls) (70k data)	64.01	75.96	72.48	77.14	77.09	74.53	73.53
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (last-pool) (70k data)	**68.54**	**77.96**	**74.52**	**79.50**	**79.87**	**78.15**	**76.42**

Table A3. Ablation studies of different generation methods of positive pairs with the same 70k training dataset.

Model	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	Avg.
SimCSE-BERT$_{base}$ (trained with 1M Wikipedia sentences) ♢	70.14	79.56	75.91	81.46	79.07	76.85	77.17
SimCSE-BERT$_{base}$ (joint-trained with 11.4k MRPC/SICK pairs and 1M Wikipedia sentences)	66.82	72.52	72.39	77.22	75.51	74.98	73.24
SimCSE-BERT$_{base}$ (trained with 1M Wikipedia sentences, then fine-tuned with 11.4k MRPC/SICK pairs)	67.12	75.74	73.08	75.85	77.36	77.47	74.44
ConSERT$_{base}$ (trained with 90k SICK and STS sentences) ♠	64.64	78.49	69.07	79.72	75.95	73.97	73.64
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (trained with 70k STS sentences, VAEGPT-Sim trained with 11.4k MRPC/SICK pairs)	68.54	77.96	74.52	79.50	79.87	78.15	76.42
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (trained with 70k STS sentences, VAEGPT-Sim trained with 11.8k MRPC/SICK sentences)	67.12	77.01	72.55	78.04	78.01	76.65	74.90
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (trained with 70k NLI sentences)	66.06	76.47	72.60	77.23	77.75	73.87	74.00
Generate-CSE-VAEGPT-Sim-BERT$_{base}$ (trained with 70k NLI sentences, VAEGPT-Sim trained with 11.8k MRPC/SICK sentences)	65.37	73.80	70.93	77.85	76.88	73.03	72.98

Table A4. Ablation studies of the influence of different training dataset in each stage.

results with the sentence-pair-fine-tuned SimCSE BERT$_{base}$, the results show that our VAEGPT-Sim can produce similar or even better results than the model trained by a much larger dataset.

At the same time, we also try to train our VAEGPT-Sim with totally unsupervised sentences and train the text representation model with target-unrelated NLI sentences (with the same scale of 70k sentences). The results show that, because training the generation model with the original sentence as the target will destroy its generation diversity (as shown in Table 2), it will result in about a 1-1.5 point loss in the effects of the final text representation models. Using target-unrelated sentences also results in a loss in final effects, but it is still better than other models trained with target-related datasets.

On the other hand, we download the official codes of several of the most outstanding contrastive learning training methods introduced in recent years, including SimCSE, DiffCSE, InfoCSE, and ESimCSE. We use their officially recommended best unsupervised training configurations to train the BERT$_{base}$ model with a completely fair sample of 70k NLI sentences, selecting the best performance within four epochs. In this comparison, which excluded other variables (Table A5), it can be observed that our method remains the most outstanding in small sample training, achieving the best results on the STS-12, STS-14, STS-15, STS-16, and STS-B, with an average (Avg. STS) superior by 0.81 to 5.77 points compared to other methods.

A.5 Types of Text Embedding Vectors

We conduct an ablation study on the different methods to obtain the text representation vector. We check 5 types of vectors from the same Generate-CSE-VAEGPT-Sim-BERT$_{base}$ model: the cls vector (cls), the pooling vector of the last layer of hidden states (last-pool), the average pooling vector of the last two layers of hidden states (last2avg), the average pooling vector of the first and last layer of hidden states (first-last-avg), and the first-last-avg vector dealing with whitening (whitening). As shown in Table A6, for the text representation model of Generate-CSE with a small-scale dataset, the last-pool vector achieves the best performance (77.38), followed by the first-last-avg (76.91), last2avg (76.88), and cls (76.69). In contrast, whitening has a negative effect on the text performances of Generate-CSE, although it used to make an obvious improvement to pretrained BERT. This may be because when the original basic model

Model (trained with 70k NLI sentences)	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	Avg.
SimCSE-BERT$_{base}$	59.02	67.91	65.58	72.30	73.94	64.51	67.21
DiffCSE-BERT$_{base}$	60.70	70.48	68.25	73.95	75.91	68.43	69.62
InfoCSE-BERT$_{base}$	64.17	75.73	67.03	77.70	75.19	70.11	71.65
ESimCSE-BERT$_{base}$	63.75	**77.99**	69.14	77.08	74.87	70.16	72.17
Generate-CSE-VAEGPT-Sim-BERT$_{base}$	**65.37**	73.80	**70.93**	**77.85**	**76.88**	**73.03**	**72.98**

Table A5. Performance comparison of different training methods for text representation models, utilizing a uniform dataset of 70k sentences.

Model	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	Avg.
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (cls)	67.78	79.24	74.88	79.94	79.18	79.14	76.69
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (last2avg)	68.66	79.74	74.92	80.28	79.02	78.64	76.88
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (first-last-avg)	68.59	79.16	74.82	80.45	79.83	78.57	76.91
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (whitening)	65.51	74.48	66.85	76.88	72.61	75.21	71.92
Generate-CSE-VAEGPT-Sim-BERT$_{large}$ (last-pool)	**68.82**	**80.18**	**75.05**	**80.46**	**80.28**	**79.45**	**77.38**

Table A6. Sentence embedding performance of different vectors from the Generate-CSE-VAEGPT-Sim-BERT$_{large}$ on 6 STS dataset, evaluated based on "wmean" Spearman's correlation.

SimCSE BERT$_{base}$	Generate-CSE-VAEGPT-Sim-BERT$_{base}$
query: a dog jogs through the grass	
a dog runs on brown grass	a dog trots through the grass
a dog runs through the long grass	a dog runs through the long grass
the black dog is running through the grass	the white and brown dog is running quickly through the grass
query: a deer jumps a fence	
a deer is jumping over a fence	a deer is jumping over a fence
a dog jumps over a hurdle	the dog leaps over the fence in the park
the dog leaps over the fence in the park	a dog jumps over a hurdle
query: there are people out on the street	
people are out on the street	people are out on the street
people are out sitting in front of a garden	a group of people standing in the street
a group of people standing in the street	people are sitting on benches

Table A7. Examples of retrieved top-3 examples by SimCSE and Generate-CSE-VAEGPT-Sim-BERT$_{base}$ from STS-B-test.

fails to provide adequate discrimination due to the underlying text representation vectors being too close, whitening is able to maximize the differences between the vectors, thereby achieving significant improvement. However, there is a limit to such improvement. When the text representation of the underlying vectors reaches a certain level of effectiveness, this method of amplifying differences may magnify erroneous judgments of text similarity and cause a decline in performance. As a result, we use the "last-pool" vector for text representation for other experiments in this article unless otherwise specified.

A.6 Experiments for Recall and Search

	Average value	Standard deviation
Rouge-1	88.19	0.08
Rouge-2	62.79	0.01
Rouge-l	81.44	0.04
Vector extreme similarity	91.56	0.09
Vector mean similarity	93.74	0.04
Word diversity - Source and Generation	60.95	0.19
Word diversity - Multiple Generation	34.43	0.67

Table A8. Statistics information for VAEGPT-Sim

	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B
Average value	0.6854	0.7796	0.7452	0.7950	0.7987	0.7815
Standard deviation	0.0120	0.0045	0.0046	0.0049	0.0028	0.0084

Table A9. Statistics information for Generate-CSE-VAEGPT-Sim BERT$_{base}$

From the Table 9 and the related analysis in the manuscript, we can find the Generate-CSE have the efficient retrieval ability on synonym sentence recall. There are some examples for the retrieval results of Generate-CSE BERT$_{base}$ and SimCSE BERT$_{base}$ shown in Table A7. BERT$_{base}$ trained with Generate-CSE obviously have more knowledge of synonyms (like 'trot' and jog), which helps it select sentences with nearly the same meaning in the first place. However, because of the lack of negative pairs in training process, the distance between sentences with different meanings evaluated by Generate-CSE BERT$_{base}$ has more space to improve.

A.7 Statistics information for key results

As explained in the experiment method, the results of each of our results are the average of many parallel experiments. Here we report the standard deviation of VAEGPT-Sim on various generative performance indicators and w-mean Spearman correlations of Generate-CSE BERT$_{base}$ on STS evaluation datasets (shown in Table A8 and Table A9).

A.8 Details for Datasets

Dataset	Scale	Usage
MRPC and SICK sentence pairs	11.4k pairs	Training VAEGPT-Sim and other generation model, further fine-tune SimCSE BERT for Table 6
MRPC and SICK sentences	11.8k sentences	Training VAEGPT-Sim and other generation model (unsupervised fine-tune)
SICK-trail sentence pairs	0.5k pairs	For generation model evaluation
PubMed sentences	55k sentences	Training text representation model for Table 1
STS sentences	70.4k sentences	Training text representation model for Table 3-7, Figure 3
NLI sentences	70.4k sentences	Training text representation model for Table 6
STS benchmark*	21.1k pairs	For text representation model evaluation
PubMed similar sentence pairs	56 pairs	For text representation model evaluation in Table 1

Table A10. Source, scale and usage of datasets used in this paper. *STS benchmark: In STS datasets, especially STS16, there are some sentence pairs that lack similarity labels. We follow the source code of previous researches and the official code of SentEval(Conneau and Kiela, 2018), and find that when performing text similarity evaluation, these pairs without labels are filtered out during the reading process and are not used in the evaluation. Therefore, we have only calculated the number of sentence pairs actually used for evaluation.

Many datasets are used in our article. For the generation model, we prepare two different datasets for training, and one test dataset for evaluation. For different experiments, we utilize 4 different training datasets to train the text representation model. Additionally, two different test datasets are prepared for varying purposes. To make it easier for readers to follow, we summarize the information and usage of them in Table A10.

Original sentence	BART (without fine-tune)	T5 (without fine-tune)	GPT2 (without fine-tune)	BART	T5	GPT2	VAEGPT-Sim
A man in a red uniform is making a jump in a dirt bike race	A man in a red uniform is making a jump in a dirt bike race	red uniform. a dirt bike race. A man in a red uniform is making a jump in the race.	at the end of a dirt road. The man, who asked not to be identified because he is not authorized to speak to the media, said he was riding his bike in the middle of the road...(too long to omit)	A man in a red uniform is making a jump in a dirt bike race	A man in a red uniform is making a jump in a dirt bike race	A man is running on a dirt track in the middle of a raceA woman is running in the dirt in a green and white dressA man and a woman are running in a race...(too long to omit)	A man is jumping in a mud bike race
A man is skiing down a hill and jumping a red obstacle	A man is skiing down a hill and jumping a red rope	A man is skiing down a hill and jumping a red obstacle course. red obstacle obstacle course on a mountain.. a man is course. course. course. course...(too long to omit)	. "It's a bit of a challenge," he said.	A man is skiing down a hill and jumping a red obstacle	A man is skiing down a hill and jumping a red obstacle	A man has jumped a red hill and is jumping a treeA man and a woman are skiing down the hill and are jumping a blue obstacleA woman and a man are jumping up a tree and are climbing a red tree...(too long to omit)	A man in a red jacket is skiing in the snow
A woman, who is seated, is singing a song and playing the guitar	A woman, who is seated, is singing a song and playing the guitar	, is singing a song and playing the guitar. A woman, who is seated, sings and plays the guitar.	, who is sitting, is singing a song. A man, who has been seated, plays the guitar and sings a song. . A woman sitting, plays a song, and sings. A man sitting, sings, plays, and plays...(too long to omit)	A woman is singing a song and playing the guitar	A woman, who is sitting, is singing a song and playing the guitar	A woman is playing a guitar and playing a songThe woman is singing the songA woman in a blue dress is playing the song and singing the guitarThe woman in blue dress has been singing...(too long to omit)	A woman is singing and playing a guitar
The person is slicing a clove of garlic into pieces	The person is slicing a clove of garlic into pieces	a clove of garlic into pieces. The person is slicing it into pieces and then cutting it into cloves.	. The person cuts the clove into pieces with a knife. The person cuts out a piece of the garlic with a razor blade. The razor blade cuts out the garlic and the garlic clove...(too long to omit)	The person is not slicing a clove of garlic into pieces	A person is slicing a clove of garlic into pieces	A person is cutting a garlic clove into pieces of garlicA man is slicing garlic into garlic slices into piecesOf garlic is being sliced into piecesof garlic...(too long to omit)	A person is cutting garlic into slices
A man is parking a car in a garage	A man is parking a car in a garage	in a garage. A man is parking a car in the garage..... He is parking his car... A woman is parking her car. The man is parked. A car is in garage...(too long to omit)	in front of the home of a man who was shot and killed by police on Tuesday. The man was shot in the head at about 10:30 p.m. at a home in the 400 block of West 17th Street...(too long to omit)	A man is parking a car in a garage	A man is parking a car in a garage	A man has parked a car on a parking lot and is parking it on the side of the streetA man in a white shirt is standing in front of a car and is driving on the sidewalk...(too long to omit)	A man parked a car on a garage
The monkey is brushing a bull dog	The monkey is brushing a bull dog	The monkey is brushing a bull dog dog. is scrubbing a monkey. is brushed by the bull dog... dog. dog. dog. dog. dog. dog. dog. dog... dog.. A monkey is dog...(too long to omit)	. "It's just a little bit of fun," he says. "It's kind of like, 'Oh my God, I can't believe I'm doing this.'"	A monkey is brushing a dog	A monkey is brushing a bull dog	A monkey is playing with a bulldog and is brushing the dogA bulldog is playing in the grass with a monkey...(too long to omit)	A monkey brushes a bulldog

Table A2. Sentences generated by different generation model: BART, T5 and GPT without and with fine-tune, as well as our VAEGPT-Sim.