Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Survey Track

Text Transformations in Contrastive Self-Supervised Learning: A Review

Amrita Bhattacharjee* , **Mansooreh Karami*** and **Huan Liu**
Arizona State University, Tempe, AZ
{abhatt43, mkarami, huanliu}@asu.edu

Abstract

Contrastive self-supervised learning has become a prominent technique in representation learning. The main step in these methods is to contrast semantically similar and dissimilar pairs of samples. However, in the domain of Natural Language Processing (NLP), the augmentation methods used in creating similar pairs with regard to contrastive learning (CL) assumptions are challenging. This is because, even simply modifying a word in the input might change the semantic meaning of the sentence, and hence, would violate the distributional hypothesis. In this review paper, we formalize the contrastive learning framework, emphasize the considerations that need to be addressed in the data transformation step, and review the state-of-the-art methods and evaluations for contrastive representation learning in NLP. Finally, we describe some challenges and potential directions for learning better text representations using contrastive methods.

1 Introduction

Self-supervised learning uses the data itself to provide the supervisory signals for representation learning without any other costly annotating processes. This is valuable in many real-world scenarios nowadays where vast quantities of information are easily available but the cost of annotating such data is high. Based on the objective function of the deep neural networks, the self-supervised models can be divided into three major groups: generative, contrastive, and generative-contrastive (or adversarial) [Liu *et al.*, 2021]. In this paper, we focus on contrastive self-supervised models in NLP. Unlike the generative models that apply the loss function on the output space, in contrastive models, the loss is measured in the representation space.

By creating pseudo-labels as supervision, the contrastive learning objective aims to bring the semantically similar samples close to each other and away from dissimilar instances. In a learning phase of a commonly used setup of CL in natural language, one sample from the training data acts as an anchor, its *augmented* version is labeled as a positive sample, and the rest of the examples in the training batch are tagged as negative samples. An illustration of this learning step in CL is presented in Figure 1. Unlike in images, the augmentation or transformation functions[1] used in creating the semantically similar pairs for texts are not well-defined and thus are more challenging. For example, in the task of word shuffling, '*He had his car cleaned*' versus '*He had cleaned his car*' has two different semantic implications and should not be used as similar pairs. On the other hand, a blind shuffle such as '*cleaned He car his had*' does not conform to the grammar rules for English and should not be considered as a positive sample. To this means, we formalize the CL setup for NLP tasks (§2). We collect studies to present a representative survey of this field as shown in Figure 2, specifically focusing on the different kinds of data augmentation used in creating the positive samples (§3) as well as sampling negative examples (§4). We also review the different losses and evaluation metrics used in this area (§5 and §6). We conclude with open problems and challenges of the self-supervised CL for text representations and emphasize the considerations needed for choosing *good* data transformations (§7).

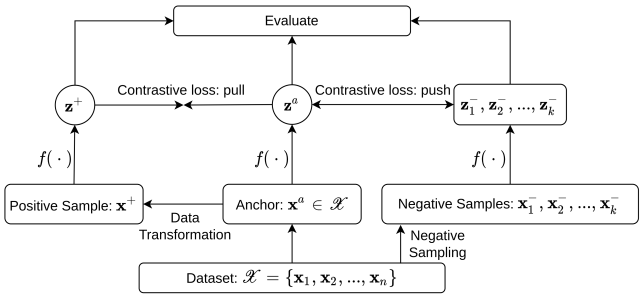


Figure 1: A learning step in CL and its effect on the representation space. The CL goal is to find a representation function (i.e., an encoder, $\mathbf{z} = f(\cdot)$) such that similar samples (i.e., the anchor, $\mathbf{x}^a$, and the positive sample, $\mathbf{x}^+$) are closer to each other and are pushed away from contrasting samples (negative-labeled instances).

Note that in this paper, we utilize bold lowercase letters for vectors (e.g., $\mathbf{x}$) as well as lowercase letters with and without

*Authors contributed equally to this work. Names are in alphabetical order.

[1] In this paper, we use data 'transformation' and data 'augmentation' interchangeably.

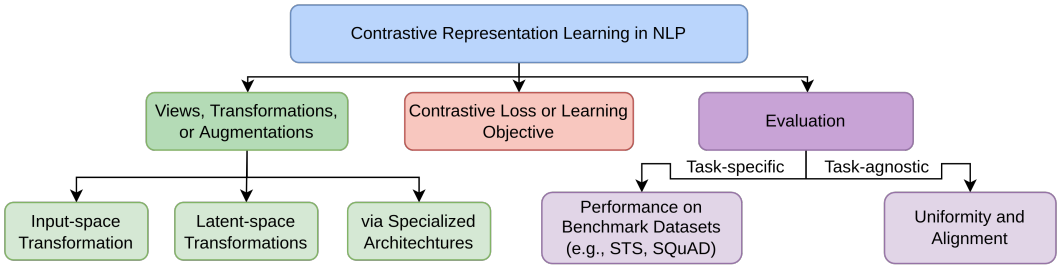


Figure 2: Taxonomy for Contrastive Learning on Text

input argument to represent functions (e.g., $f(\cdot)$) and scalars (e.g., i), respectively. Calligraphic letters are used for denoting sets and losses (e.g., $\mathcal{X}$) and Greek letters for parameters (e.g., β). To represent a sample from a set, we use subscript letters or numbers (e.g., $\mathbf{x}_k$ or $\mathbf{x}_1$). Uppercase letters in regular font is used for random variables (e.g., U). Finally, superscript T denotes the transpose of a matrix or a vector.

2 Contrastive Learning

In this section, we will describe the contrastive learning (CL) framework. In CL, the aim is to learn a representation function $f : \mathcal{X} \rightarrow \mathcal{S}^{d-1}$ that maps all the data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ that are sampled from a distribution $p(\mathbf{x})$ to a hypersphere space in $\mathbb{R}^d$. Using commonly-used noise contrastive loss (NCE) [Gutmann and Hyvärinen, 2010], which is a lower bound on the mutual information of two random variables [Poole *et al.*, 2019], the CL framework tries to pull the similar examples towards each other and push them away from the dissimilar examples. This representation learning is conducted with an assumption that we have access to the similarity, p^+, and dissimilarity, p^-, distributions. The quality of the learned representations in CL is highly dependent on informativity of the positive pairs $(\mathbf{x}^a, \mathbf{x}^+)$ and the negative samples $\{\mathbf{x}_1^-, \mathbf{x}_2^-, ..., \mathbf{x}_k^-\}$, where $\mathbf{x}^a \in \mathcal{X}$ is called anchor [Robinson *et al.*, 2020]. The samples should be paired such that they are semantically similar and dissimilar to $\mathbf{x}^a$ for the positive and negative data points.

Following the formulation in [Arora *et al.*, 2019], we formally define the concept of *semantic similarity* by assuming a set of latent classes $\mathcal{C}$. Lets assume that $p(c)$ is a distribution over the latent classes that shows the natural occurrence of these classes in the unlabeled setting, where $c \in \mathcal{C}$. We also assume similar data points (i.e., $\mathbf{x}^a$ and $\mathbf{x}^+$) are i.i.d. and drawn from the same class distribution. Then, for some class c sampled randomly from $p(c)$, the similarity and dissimilarity distributions are defined as:

$$p^+(\mathbf{x}^a, \mathbf{x}^+) = \mathop{\mathbb{E}}_{c\sim p(c)} p(\mathbf{x}^a|c)p(\mathbf{x}^+|c)p(c) \tag{1}$$

$$p^-(\mathbf{x}^-) = \mathop{\mathbb{E}}_{c\sim p(c)} p(\mathbf{x}^-|c)p(c), \tag{2}$$

in which $\mathbf{x}^-$ is sampled i.i.d. from the marginal distribution of p^+. Finally, the learning process will be maintained using the following loss:

$$\mathcal{I}(U;V) \geq \mathcal{L}_{\text{NCE}}(\mathbf{u}_i, \mathbf{v}_i) = \mathbb{E}_{(\mathbf{u}_i,\mathbf{v}_i)\sim p^+} \mathbb{E}_{\mathbf{v}_{1:k}\sim p^-} \left[\log \frac{e^{g(\mathbf{u}_i,\mathbf{v}_i)}}{\frac{1}{k+1}\sum\limits_{j\in\{i,1:k\}} e^{g(\mathbf{u}_i,\mathbf{v}_j)}} \right], \tag{3}$$

where $\mathcal{I}(U;V)$ is the mutual information between two random variables U and V, with $\mathbf{u}$ and $\mathbf{v}$ as their realizations, respectively. In our example in Figure 1, U and V are derived from the same random variable sampled from the distribution $p(\mathbf{x})$. With $\mathbf{z} = f(\cdot)$ as our encoder, $g(\mathbf{u}, \mathbf{v})$ can be defined as any similarity function between $f(\mathbf{u})$ and $f(\mathbf{v})$.

Current self-supervised CL approaches empirically try to follow the above setting, but face some challenges. First, they do not have access to the actual classes and the similarity/dissimilarity information, so equations (1) and (2) cannot be calculated directly. To this means, some heuristics have been applied to account for similarity such as user-specified transformation functions, data augmentation methods, and unsupervised clustering. On the other hand, negative instances are often sampled uniformly from the data, regardless of whether they share any semantic conformity with x^a or not. In other words, if the selected instance as the negative is semantically similar to the anchor, their representation are still pushed apart. This sampling bias would lead to a sub-optimal representation as it cannot capture the true semantic structure of the data [Li *et al.*, 2020; Chuang *et al.*, 2020]. Second, in the process of data augmentation, the i.i.d. assumption might not hold anymore. For example, in NLP, perturbing the instances in creating the positive samples might alter the semantics, change the distribution of the sample, and create a more negative pair rather than a positive one. In this paper, we overview different heuristics used in defining the positive and negative samples in text-based CL. Furthermore, we explain several caveats and hard assumptions implicit in standard CL frameworks, and hence, guide readers to delve into open problems in this area.

3 Data Augmentation

In a standard CL framework, the first step is to generate positive and negative samples for a given anchor data point. However, making transformations on the anchor to generate such positive samples is a more complicated task in the text domain, given the discrete nature of the input space. In this

Transformation or Augmentation	Loss	Tasks	Paper
BERT Siamese/Dual Encoder	Approximate nearest neighbor NCE	Dense text retrieval	[Xiong *et al.*, 2020]
Summarization of documents (no negative examples used)	NLL + multiple similarity losses	Abstractive text summarization	[Xu *et al.*, 2021]
Adversarial +ve/-ve examples by adding perturbations to the latent representation	combination of KL divergence and contrastive loss	Machine translation, text summarization, question generation	[Lee *et al.*, 2020]
Spans of text sampled from the same document, i.e. anchor and positive from the same document	InfoNCE + MLM loss	SentEval benchmark tasks	[Giorgi *et al.*, 2021]
Word deletion, Span deletion, Reordering, Synonym substitution	n-pairs contrastive loss	GLUE	[Wu *et al.*, 2020b]
Back-translation	n-pairs loss (based on momentum CL)	GLUE	[Fang *et al.*, 2020]
Different dropout masks for anchor and positive	n-pairs loss	STS tasks and SICK-R [Marelli *et al.*, 2014]	[Gao *et al.*, 2021]
Adversarial perturbations, token shuffling, cutoff, dropout	n-pairs + cross-entropy for supervised task	STS tasks and SICK-R [Marelli *et al.*, 2014]	[Yan *et al.*, 2021]
Inference relations from SNLI [Bowman *et al.*, 2015]	supervised contrastive loss + cross-entropy	STS tasks and SentEval	[Liao, 2021]
Corrupted and cropped sequences as positives	MLM + CLM + a form of n-pairs contrastive loss	GLUE and SQuAD	[Meng *et al.*, 2021]

Table 1: Types of Transformations/Augmentations along with the loss function and evaluation task, over a set of representative works (NLL: Negative Log-Likelihood, MLM: Masked Language Modeling Loss, CLM: Corrective Language Modeling Loss)

section, we review some commonly used transformation or augmentation methods used in different settings. Although a direct mapping between the downstream task and the appropriate augmentation is not straight-forward, a comparison of transformations, tasks, and losses over some representative works is given in Table 1.

3.1 Input-Space Transformations

The most straight-forward class of text transformations would be operations performed in the discrete input space, also known as instance-based transformation. Even though these transformation methods are not as intuitive as similar transformations in image (such as cropping, flipping or rotating), different approaches have been explored in literature with varying degrees of success. In their DeCLUTR method, Giorgi et al. [Giorgi *et al.*, 2021] used a *span sampling* approach and considered segments that are adjacent to, overlapped with, or subsumed the original text segment, as positive samples. The augmented samples may also be created by *lexical and sentence transformation*. Wu et al. [Wu *et al.*, 2020b] used approaches such as word deletion, span deletion, token reordering, and synonym substitution for sentence augmentation in their CL method, CLEAR. Other token-level augmentation methods that have been proposed as standard data augmentation [Wei and Zou, 2019], such as synonym replacement, random insertion, random swap, and random deletion, can also be used for generating the positive pairs. A recent work in open domain question answering [Ram *et al.*, 2021] used cross-passage recurring spans of text as the positives - one span act as the anchor (or 'query' in dense retrieval terms), while another acts as the positive.

3.2 Latent-Space Transformations

Techniques proposed for standard data augmentation in low-resource learning settings can also be used to generate positive samples for contrastive representation learning in text. Some of these methods that generally preserve the semantic meaning of the original text include *back-translation* using another intermediate language [Fang *et al.*, 2020; Xie *et al.*, 2020] and *language models* to replace selected words from the text with nearest neighbor words [Jiao *et al.*, 2020] such as word2vec [Mikolov *et al.*, 2013] or GloVe. Xu et al. [Xu *et al.*, 2021] utilized a document-level CL to train a document-level *summarization model*. For the CL scheme, the authors use the original document, its gold summary, and the generated summary as different views of the data. The choice of these views is motivated by the idea that an article and its summarization must be close to each other in the semantic space. Meng et al. [Meng *et al.*, 2021] used CL as part of their model for language model pretraining. The positive samples consist of the cropped version that keeps a random 90% contiguous span of the original sentence and the recovered sentence from the *masked language model* by randomly masking some words in the original sentence.

3.3 Transformations via Architecture and Combined Methods

Positive pairs for text may also be generated using slightly different architectures or modifying some aspect of the architecture in a certain way. One such architecture based method for text augmentation in CL utilizes *dropout noise*. Gao et al. [Gao *et al.*, 2021] create the positive pairs by feeding the sample input to the encoder twice and getting two embeddings with different dropout masks. The embeddings then is used as i and j samples in equation (4). A perturbed version of the input can be generated by *adversarial training* and tagged as a positive example. Yan et al. [Yan *et al.*, 2021] not only used lexical transformation and dropout approaches for data augmentation but also perturbed the input by applying Fast Gradient Value (FSV) [Rozsa *et al.*, 2016] as the adversarial attack method. Finally, some approaches consider *inference relations* in Natural Language Inference datasets to create the desired data. These datasets consist of a premise-hypothesis pair with three different relationships: entailment,

neutral, or contradiction. The premise acts as an anchor while the hypothesis would be labeled as positive if the relationship is entailment and negative if it is either neutral or contradiction [Liao, 2021].

4 Negative Sampling

In previous sections, we review heuristics used to create the positive samples. Unlike studies in deep metric learning [Suh *et al.*, 2019], the value of the negative samples has been understated in unsupervised contrastive representation learning. Different samples of negative datapoints have different effects on the quality of the final representation. An efficient sampling function for these negative examples can also facilitate the learning process by correcting the model's mistake more quickly. Specifically, samples that are mapped near the anchor with high propensity in having the same label can significantly help in improving the representations. These samples are known as hard negatives. When latent classes are known (i.e., the supervised case), it is easy to identify task-specific hard negatives. But, in unsupervised settings, mining the hard negatives is more challenging. In these settings, researchers often increase the batch size such that the loss function covers a diverse set of negative samples [Chen *et al.*, 2020; He *et al.*, 2020]. However, beside the heavy burden of large memory usage, Arora et al. [Arora *et al.*, 2019] prove that due to the inherent nature of CL, large number of negative samples in some cases might even decrease the performance of the downstream task. To this means, researchers proposed various methods in sampling the negative examples.

Robinson et al. [Robinson *et al.*, 2020] proposed a simple method for finding hard negative samples. The authors contemplate two ways for sampling the negatives: (1) they used heuristics to make sure that the anchor and the negative sample correspond to different latent classes, and (2) the selection of the negative samples is regulated by the parameter β that controls the degree of similarity to the anchor, $e^{\beta f(\mathbf{x}^a)^T f(\mathbf{x}^-)}$. In other words, β would up-weight the negative points that have larger inner product (i.e., small Euclidean distance) with the anchor. They examined the effectiveness of their approach on learning meaningful representations for different tasks on images, graphs, and texts. Similarly, Wu et al. [Wu *et al.*, 2020a] showed that difficult samples drawn from their proposed restricted class of distributions would pick the ones that are more similar to the anchor, hence yielding a stronger representations. Tested only on visual transfer tasks, they also defined a conditional noise CL estimator that has a lower variance than the commonly-used CL losses.

Xiong et al. [Xiong *et al.*, 2020] raised the issue of in-batch negative and hard negative sampling, as local negative sampling will lead to diminishing gradient norms, large stochastic gradient variances, and slow convergence. To overcome this problem, they proposed a new CL method named as Approximate nearest neighbor Negative Contrastive Estimation (ANCE) which selects the negative samples from the entire dataset using an asynchronously updated ANN index. In the context of vision, Kalantidis [Kalantidis *et al.*, 2020] also raised the same issue with the in-batch negative sampling as well as the time-consuming use of memory banks that needs to keep a large memory up-to-date. The authors proposed the Mixing of Contrastive Hard negatives (MoCHi) approach to synthesize hard negative features, by creating convex linear combinations of the hardest existing negatives. Their experiments show that MoCHi is able to learn generalizable representations faster than the SOTA self-supervised approaches. Comparably, Chuang et al. [Chuang *et al.*, 2020] pointed out that the selected negative samples in traditional CL might suffer from the sampling bias which can lead to significant performance drop. They proposed an unsupervised debiased contrastive loss that corrects for the sampling of datapoints with the same label. Giorgi et al. [Giorgi *et al.*, 2021] used both easy and hard negative samples from text documents. Their definition of hard negative samples is those that are in the same document as the anchor while their text is not subsumed, overlapped, or adjacent to the anchor. However, this would not guarantee that they are not semantically unrelated.

5 Contrastive Losses

Although equation (3) is a general form of the contrastive loss, several variations of contrastive loss function have been used so far. One of the earliest contrastive loss functions used was in the context of energy based model to measure similarities between faces for face verification [Chopra *et al.*, 2005]. For two data instances, this intuitive learning objective was intended to give a small value of the loss if the data instances were from the same class, and would give a large loss value if they are from different classes. Keeping convergence and training efficiency in mind, over time, different variants of contrastive loss functions have been proposed.

In a self-supervised manner, the contrastive loss for a pair of positive samples $\mathbf{x}^a$ and $\mathbf{x}^+$ is calculated as follows:

$$\mathcal{L}(\mathbf{x}^a, \mathbf{x}^+) = -\log \frac{e^{\text{sim}(f(\mathbf{x}^a), f(\mathbf{x}^+))/\tau}}{\sum_{k=1}^{2n} \mathbb{1}_{[\mathbf{x}^a \neq \mathbf{x}_k^-]} e^{\text{sim}(f(\mathbf{x}^a), f(\mathbf{x}_k^-))/\tau}}, \quad (4)$$

where n is the number of samples in one batch, $\mathbb{1}_{[\mathbf{x}^a \neq \mathbf{x}_k^-]} \in \{0, 1\}$ is an indicator function, τ denotes a temperature hyperparameter, and $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{||\mathbf{u}|| \cdot ||\mathbf{v}||}$ is the cosine similarity between two vectors.

Triplet loss [Schroff *et al.*, 2015] uses a triplet of an anchor, a positive sample (i.e. has the same label as the anchor) and a negative sample (i.e. different label from the anchor). This loss (shown in equation (5)) tries to minimize the distance between the anchor and the positive and increase the distance between the anchor and the negative.

$$\begin{aligned}\mathcal{L}(\mathbf{x}^a, \mathbf{x}^-, \mathbf{x}^+) = \max(&||f(\mathbf{x}^a) - f(\mathbf{x}^+)||^2 \\ &-||f(\mathbf{x}^a) - f(\mathbf{x}^-)||^2 + \alpha, 0),\end{aligned} \quad (5)$$

where $\mathbf{x}^a$ is the anchor datapoint, $\mathbf{x}^-$ is the negative sample, $\mathbf{x}^+$ is the positive sample, and α is the margin between positive and negative samples. Similar approaches for learning the distance metric between instances have been proposed in other works such as [Schultz and Joachims, 2004; Wang *et al.*, 2014]. Apart from the direct task of distance metric learning, contrastive loss has also been used for dimensionality reduction [Hadsell *et al.*, 2006]. However, one major problem with the triplet loss and several other similar variants of contrastive losses is that of *hard negative mining*. The

model would successfully learn the distance between positive and negative samples if the triplets are selected and constructed properly. However, as we explained in the previous section, searching for hard negatives over the entire training dataset is infeasible in practice, hence there have been several interesting approaches to solve this (as listed in §4). Oh Song et al. [Oh Song *et al.*, 2016] propose the lifted structure loss that results in more efficient training and stable optimization.

$$\mathcal{L}(\mathbf{x}^a, \mathbf{x}^+) = \mathop{\mathbb{E}}_{(\mathbf{x}^a,\mathbf{x}^+)\sim p^+} \max(0, \mathcal{J}(\mathbf{x}^a, \mathbf{x}^+))^2,$$
$$\mathcal{J}(\mathbf{x}^a, \mathbf{x}^+) = \max[\max_{\mathbf{x}^a,\forall\mathbf{x}^-} \alpha - d(\mathbf{x}^a, \mathbf{x}^-), \quad (6)$$
$$\max_{x^+,\forall\mathbf{x}^-} \alpha - d(\mathbf{x}^+, \mathbf{x}^-)] + d(\mathbf{x}^a, \mathbf{x}^+)$$

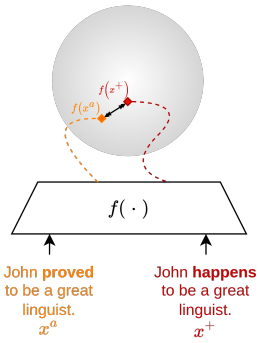
where $d(\mathbf{u}, \mathbf{v}) = ||f(\mathbf{u}) - f(\mathbf{v})||_2$ is the L2 distance between the representations of $\mathbf{u}$ and $\mathbf{v}$, and α is a margin parameter. This utilizes all the positive and negative pairs in a training batch. Furthermore, this approach tries to improve the representation learned, by looking for 'difficult' negatives for a set of randomly chosen positive samples. Another issue with triplet loss and contrastive loss is that, especially for multi-class cases, it results in unstable updates and slow convergence. This is because in each step, only one comparison is being made with only one negative sample. This slows down the convergence. To alleviate this problem, [Sohn, 2016] proposed the n-pairs loss (equation (7)), where in each step the loss is computed using a $(n+1)$ length tuple.

$$\mathcal{L}(\{\mathbf{x}^a, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{n-1}\}) =$$
$$\log\left(1 + \sum_{i=1}^{n-1} \exp\left(f(\mathbf{x}^a)^T f(\mathbf{x}_i^-) - f(\mathbf{x}^a)^T f(\mathbf{x}^+)\right)\right) \quad (7)$$

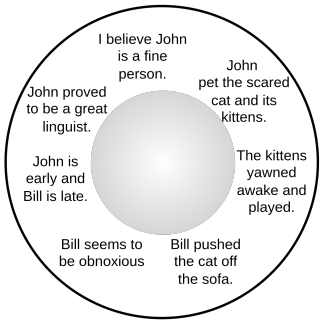
where $f(\cdot)$ is the embedding kernel of the deep neural network (i.e., the encoder). This has one anchor element $\mathbf{x}^a$, one positive element $\mathbf{x}^+$, and $(n-1)$ negative samples $\{\mathbf{x}_i\}_{i=1}^{n-1}$. The n-pairs loss may be thought of as a special case of the lifted structure loss, where the batch contains positive pairs from disjoint classes. Furthermore, unlike the n-pairs loss, the lifted structure loss uses a max-margin based distance function in the loss formulation.

6 Evaluation Metrics

Most of the methods rely on the performance of the downstream tasks (e.g., accuracy on labeled benchmark datasets), in order to evaluate the quality of the learned representations. Moreover, for some specific objectives such as measuring the balancedness of a feature space, they rely on the linear separability performance which is evaluated by the accuracy of a linear classifier over the representation vectors [Kang *et al.*, 2020; Jiang *et al.*, 2021]. However, in learning *universal* representations and in an *unsupervised* manner, we can evaluate the quality of the representation by measuring how well the CL method separate similar pairs from dissimilar samples. Wang and Isola [Wang and Isola, 2020] proposed two properties related to contrastive loss in assessing the CL representations. Since CL aims to find a representation space that the information is most shared between positive pairs as well as invariant to other noise factors, they defined these metrics:



(a) Closeness of features of positive pairs, $(\mathbf{x}^a, \mathbf{x}^+) \sim p^+$, on the hypersphere space.



(b) Uniformity in the distribution of features on a hypersphere.

Figure 3: An illustration of (a) Alignment and (b) Uniformity of text representations on the output unit hypersphere. This figure is inspired by Wang and Isola's [Wang and Isola, 2020] illustration of these metrics on images. The sentences are extracted from the GLUE dataset [Wang *et al.*, 2019] with some modifications.

- Alignment: the anchor and positive sample representations on the hypersphere space ($\mathcal{S}^{d-1}$) should be aligned and close to each other (Figure 3a), i.e., the absolute distance of the anchor and positive sample representation should be as small as possible:

$$\mathcal{L}_{\text{align}}(f;\rho) = \mathop{\mathbb{E}}_{(\mathbf{x}^a,\mathbf{x}^+)\sim p^+} \left[\| f(\mathbf{x}^a) - f(\mathbf{x}^+) \|_2^\rho\right], \rho > 0 \quad (8)$$

- Uniformity: the distribution of the representations should roughly be uniform in the hypersphere space to preserve as much information of the data as possible (Figure 3b). This can be calculated as the logarithm of the average pairwise Gaussian potential kernel (also known as the Radial Basis Function (RBF) kernel), with parameter γ, between the representations of the data points $\mathcal{X}$:

$$\mathcal{L}_{\text{uniform}}(f;\gamma) = \log \mathop{\mathbb{E}}_{(\mathbf{x}_i,\mathbf{x}_j)\overset{i.i.d}{\sim}\mathcal{X}} \left[e^{-\gamma\|f(\mathbf{x}_i)-f(\mathbf{x}_j)\|_2^2}\right] \quad (9)$$

The authors also empirically showed that both $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ are strongly agree with and causally affect downstream task performance.

7 Challenges and Open Problems

Alongside the success of CL in unsupervised settings, there has also been community-wide discussions regarding the generalizability of the representations learned by such methods, the appropiateness of the transformations, and several other related issues. In this section, we go deeper into some of the main challenges in contrastive self-supervised learning for text and point readers to potential directions for future research.

The Selection of a Good Transformation Function. Contrastive representation learning in the self-supervised setting assumes that the transformations that are done on the data points are semantically invariant, and hence are simply two 'views' of the instance. Ideally the transformations or augmentations performed should not alter the semantic meaning of the data point. Most contrastive representation learning schemes assume that the downstream task that uses the learned representations would be invariant to the transformation performed during the learning process. For example, as explained in [Xiao *et al.*, 2020], for a downstream task that does fine-grained classification of bird species, augmentations that involve modifying the color and texture of the image should not be performed, as these are useful features in identifying the species of bird. Similarly for text, augmentations that change the tone or sentiment of the sentence should not be used for learning representations in a system that does sentiment classification as the downstream task. A recent effort in this direction for images tries to learn invariant representations [Misra and Maaten, 2020]. In text, apart from the downstream task, the suitable transformation may also depend on the language.

Negative Samples and Sampling Bias. In the supervised counterpart of CL [Khosla *et al.*, 2020], sampling negative examples from truly different classes has shown to improve the performance of the representations. However, as mentioned in §4, due to CL's unsupervised manner and the lack of access to the labels, we might accidentally sample false negatives and accept examples that are in reality semantically similar to the anchor. Future work is needed to mitigate this sampling bias without relying on the actual labels of the data.

Counterfactually-Augmented Data as Positive and Negative Samples. Counterfactual examples have long been utilized and known to be useful for training, evaluating, and improving NLP models [Moraffah *et al.*, 2020; Morris *et al.*, 2020] as well as mitigating bias [Maudslay *et al.*, 2019; Kaushik *et al.*, 2019; Hu and Li, 2021]. By making sure that the counterfactual examples are plausible and not out-of-distribution to models [Hase *et al.*, 2021], there is a potential in creating augmented data that estimates the latent class for positive or negative examples and satisfies the assumptions for similarity and dissimilarity distributions. For example, by utilizing the relations between pairs of counterfactual examples we are able to find what changes in the input space are related to the change in the label [Teney *et al.*, 2020]. This technique, that is known as counterfactual data augmentation, seeks to eliminate spurious correlations using causal interventions [Kaushik *et al.*, 2019]. Moreover, label-preserving data augmentations can be used in generating examples that are similar to the anchor [Joshi and He, 2021]. Current generation methods mostly rely on human expert annotators to create the counterfactually-altered data in which they only instantiate limited types of perturbations like word substitutions. Methods such as Polyjuice [Wu *et al.*, 2021] are attempts in automatically creating fluent and diverse counterfactual examples which support various downstream tasks on different domains. However, while using counterfactual examples in the context of CL, we have to make sure that the assumptions, such as preserved latent classes, identical distribution for positive samples, and the dissimilarity distribution requirements are not violated.

Euclidean vs non-Euclidean Spaces. Most of the self-supervised NLP representation models such as word2vec [Mikolov *et al.*, 2013], GloVe [Pennington *et al.*, 2014], and skip-thought vectors [Kiros *et al.*, 2015] are trained in the Euclidean space which aim to find a representation such that the distance between the vectors corresponds to their semantic proximity. Non-Euclidean spaces have also been explored for the purpose of the text representations. For example, Nickel and Kiela [Nickel and Kiela, 2017] proposed a Poincaré embedding by utilizing the hyperbolic geometry for learning the similarity and the hierarchy of objects in predicting lexical entailment. Similarly, Dhingra et al. [Dhingra *et al.*, 2018] showed that learning a Poincaré embedding for hierarchical structures will lead to an improvement on other downstream tasks and provided some evidence on the intuition of the hyperbolic embedding for structural data. Moreover, Meng et al. [Meng *et al.*, 2019] showed that the spherical text embeddings would intrinsically capture the directional similarity. They proposed a model that would jointly learn word and paragraph embeddings. Prior to that, Batmanghelich et al. [Batmanghelich *et al.*, 2016] applied von Mises-Fisher distribution to model the density of the words over a unit sphere as well as discovering the number of topics in the data. When all is said and done, the question of which space would capture the natural representation of the text is not yet rigorously answered. One possible direction is to critique the intuitions behind using the Euclidean, Hyperbolic, or Spherical spaces and provide evidence on the smoothness of the decision boundaries.

Acknowledgments

This research is supported by the DARPA (HR001120C0123) and ONR (N00014-21-1-4002). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[Arora *et al.*, 2019] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *36th ICML*, pages 9904–9923. International Machine Learning Society (IMLS), 2019.

[Batmanghelich *et al.*, 2016] Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman.

Nonparametric spherical topic modeling with word embeddings. In *Proc. of the ACL*, page 537, 2016.

[Bowman *et al.*, 2015] Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proc. of the 2015 Conference on EMNLP*, pages 632–642, 2015.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICLR*, pages 1597–1607. PMLR, 2020.

[Chopra *et al.*, 2005] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on CVPR*, pages 539–546, 2005.

[Chuang *et al.*, 2020] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *NeurIPS*, 2020.

[Dhingra *et al.*, 2018] Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. Embedding text in hyperbolic spaces. In *TextGraphs-12 Workshop*, pages 59–69, 2018.

[Fang *et al.*, 2020] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. In *ACL*, page 7517–7523, 2020.

[Gao *et al.*, 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of EMNLP*, 2021.

[Giorgi *et al.*, 2021] John M Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. Declutr: Deep contrastive learning for unsupervised textual representations. In *ACL-IJCNLP*, page 879–895, 2021.

[Gutmann and Hyvärinen, 2010] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proc. of the thirteenth AISTATS*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[Hadsell *et al.*, 2006] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on CVPR*, pages 1735–1742, 2006.

[Hase *et al.*, 2021] Peter Hase, Harry Xie, and Mohit Bansal. The out-of-distribution problem in explainability and search methods for feature importance explanations. *Advances in NeurIPS*, 34, 2021.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the Conference on CVPR*, pages 9729–9738, 2020.

[Hu and Li, 2021] Zhiting Hu and Li Erran Li. A causal lens for controllable text generation. *NeurIPS*, 34, 2021.

[Jiang *et al.*, 2021] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Improving contrastive learning on imbalanced data via open-world sampling. *NeurIPS*, 2021.

[Jiao *et al.*, 2020] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the ACL: EMNLP 2020*, pages 4163–4174, 2020.

[Joshi and He, 2021] Nitish Joshi and He He. An investigation of the (in) effectiveness of counterfactually augmented data. *arXiv preprint arXiv:2107.00753*, 2021.

[Kalantidis *et al.*, 2020] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *34th Conference on NeurIPS*, 2020.

[Kang *et al.*, 2020] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *ICLR*, 2020.

[Kaushik *et al.*, 2019] Divyansh Kaushik, Eduard Hovy, and Z. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*, 2019.

[Khosla *et al.*, 2020] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in NeurIPS*, 33, 2020.

[Kiros *et al.*, 2015] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in NeurIPS*, pages 3294–3302, 2015.

[Lee *et al.*, 2020] Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. Contrastive learning with adversarial perturbations for conditional text generation. In *ICLR*, 2020.

[Li *et al.*, 2020] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2020.

[Liao, 2021] Danqi Liao. Sentence embeddings using supervised contrastive learning. *arXiv preprint arXiv:2106.04791*, 2021.

[Liu *et al.*, 2021] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Marelli *et al.*, 2014] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A sick cure for the evaluation of compositional distributional semantic models. In *Proc. of LREC'14*, pages 216–223, 2014.

[Maudslay *et al.*, 2019] Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. In *EMNLP-IJCNLP*, 2019.

[Meng *et al.*, 2019] Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. Spherical text embedding. *Advances in NeurIPS*, 32:8208–8217, 2019.

[Meng *et al.*, 2021] Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song.

Coco-lm: Correcting and contrasting text sequences for language model pretraining. *NeurIPS*, 2021.

[Mikolov *et al.*, 2013] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *ICLR, Workshop Track Proceedings*, 2013.

[Misra and Maaten, 2020] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proc. of the IEEE/CVF Conference on CVPR*, pages 6707–6717, 2020.

[Moraffah *et al.*, 2020] Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. Causal interpretability for machine learning-problems, methods and evaluation. *ACM SIGKDD Explorations Newsletter*, 22(1):18–33, 2020.

[Morris *et al.*, 2020] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proc. of the 2020 Conference on EMNLP*, pages 119–126, 2020.

[Nickel and Kiela, 2017] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in NeurIPS*, 30:6338–6347, 2017.

[Oh Song *et al.*, 2016] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proc. of the IEEE conference on CVPR*, pages 4004–4012, 2016.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on EMNLP*, pages 1532–1543, 2014.

[Poole *et al.*, 2019] Ben Poole, Sherjil Ozair, Aäron van den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *ICML*, 2019.

[Ram *et al.*, 2021] Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. Learning to retrieve passages without supervision. *arXiv preprint arXiv:2112.07708*, 2021.

[Robinson *et al.*, 2020] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*, 2020.

[Rozsa *et al.*, 2016] Andras Rozsa, Ethan M Rudd, and Terrance E Boult. Adversarial diversity and hard positive generation. In *Proc. of the IEEE Conference on CCVPR Workshops*, pages 25–32, 2016.

[Schroff *et al.*, 2015] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. of the IEEE conference on CVPR*, pages 815–823, 2015.

[Schultz and Joachims, 2004] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *NeurIPS*, 16:41–48, 2004.

[Sohn, 2016] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in NeurIPS*, pages 1857–1865, 2016.

[Suh *et al.*, 2019] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In *Proc. of the IEEE/CVF Conference on CVPR*, pages 7251–7259, 2019.

[Teney *et al.*, 2020] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. In *16th ECCV*, pages 580–599. Springer, 2020.

[Wang and Isola, 2020] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pages 9929–9939. PMLR, 2020.

[Wang *et al.*, 2014] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393, 2014.

[Wang *et al.*, 2019] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

[Wei and Zou, 2019] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proc. of the 2019 Conference on EMNLP-IJCNLP*, pages 6382–6388, 2019.

[Wu *et al.*, 2020a] Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. Conditional negative sampling for contrastive learning of visual representations. In *ICLR*, 2020.

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

[Wu *et al.*, 2021] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the ACL*, 2021.

[Xiao *et al.*, 2020] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. In *ICLR*, 2020.

[Xie *et al.*, 2020] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in NeurIPS*, 33:6256–6268, 2020.

[Xiong *et al.*, 2020] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *ICLR*, 2020.

[Xu *et al.*, 2021] Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. Sequence level contrastive learning for text summarization. *arXiv preprint arXiv:2109.03481*, 2021.

[Yan *et al.*, 2021] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *ACL-IJCNLP*, 2021.