Unsupervised Contrastive Representation Learning: A Survey

Kelsey Ball*
University of Texas at Austin

Abstract

Unsupervised contrastive representation learning uses unlabeled data to learn a feature space in which similar inputs are closer together (in Euclidean distance) than dissimilar ones. An ideal feature space encodes relevant features from the input space, reducing the amount of labeled data needed for classification. In this paper, we survey theoretical and applied results for image and text representation learning that use unsupervised contrastive methods.

Table of Contents

*kelseyball@utexas.edu

1 Introduction

Unsupervised representation learning uses unlabeled data to learn a representation function f that maps inputs to a points in a feature space. A good representation function reduces the requirement for labeled data in downstream classification tasks by replacing an input x with its representation $f(x)$. It seems reasonable that a good representation function should respect the semantic similarity of inputs; that is, semantically similar inputs should have similar representations.

Contrastive objectives formalize this hypothesis. Assume we have sample access to the following trio of inputs: an "anchor" x, a "positive" x^+ which is similar to x, and N "negatives" $\{x_i^-\}_{i=1}^N$ which are dissimilar to x. Then, we can minimize a contrastive objective of the form:

$$\mathop{\mathbb{E}}_{\substack{x\sim p\\ x^+\sim p_x^+\\ x^-\sim p_x^-}} \left[-\log \left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + \sum_{i=1}^N e^{f(x)^T f(x_i^-)}} \right) \right], \tag{1}$$

where p is a distribution over all inputs, p_x^+ is a distribution over valid positives for x, and p_x^- is a distribution over valid negatives for x. Note that, when "similarity" corresponds to a shared class label, we have that $p_x^-(\hat{x}) = p(\hat{x} \mid h(x) \neq h(\hat{x}))$, where h is the labeling function.

The objective is minimized when the numerator is almost as large as the denominator. Observe that this happens when (a) $f(x)^T f(x^+)$ is large, i.e., when the anchor and positive samples are mapped to nearby points; and (b) $f(x)^T f(x_i^-)$ is small for all i, i.e., when the negative samples are mapped to points that are far from the anchor. Hence, minimizing Eq. (1) yields the desired feature space.

While contrastive objectives can be used in both supervised and unsupervised settings, this survey will focus on unsupervised methods. In the unsupervised setting, is it reasonable to assume sample access to positives and negatives, given that the data is unlabeled? Prior knowledge of the domain offers heuristics for generating positives and negatives, given an anchor sample. For example, in the image domain, we can often rotate or blur an image without fundamentally changing its content or "meaning". The resulting "image augmentation" will be a positive sample that is semantically similar to the original. In text, we rely on the distributional hypothesis: words which co-occur tend to have similar meanings. Therefore, we can often use neighboring words and sentences as positive samples. In both domains, most approaches generate negatives using random sampling. The underlying assumption is that, given a large enough dataset, a randomly sampled other point from the dataset is unlikely to be similar in meaning to the anchor.

2 Theoretical Progress

In this section we discuss two themes in existing theoretical work on unsupervised contrastive representation learning. The first theme regards negative sampling. While simpler to implement and analyze, random negative sampling creates two issues: (1) false negatives, where a random negative belongs to the same class as the anchor, and (2) negatives which are too "easy", i.e., so dissimilar from the anchor that they provide no useful information. These issues motivate the strategy of "hard negative mining": identifying negatives which have a different class label from the anchor, but are similar to the anchor and thus mapped to a nearby point. Such examples are likely to provide more useful gradient information than "easy" negatives, which are already mapped to a faraway point. Section 2.1 discusses a line of work that studies the extent of these issues and ways of addressing them.

The second theme regards a conditional independence assumption. Early theoretical works assume that an anchor and positive sample are independent conditioned on the underlying class label. Such assumptions ease analysis but are overly strong in practice. Section 2.2 surveys some works which attempt to relax this type of assumption.

2.1 Negative Sampling

Arora et al. [AKK+19] provide a theoretical framework for unsupervised contrastive representation learning: Given a set of M points $\mathcal{X}$, they posit that there exists a set of latent classes $\mathcal{C}$, where each latent class $c \in \mathcal{C}$ has a distribution $\mathcal{D}_c$ over $\mathcal{X}$ that indicates the relevance of a point to that class. They also assume a distribution ρ over the latent classes which characterizes how the latent classes $\mathcal{C}$ occur in unlabeled data. Positive pairs are drawn from the distribution $\mathcal{D}_{sim}(x, x^+) = \mathop{\mathbb{E}}_{c\sim\rho} \mathcal{D}_c(x)\mathcal{D}_c(x^+)$, and negative samples are sampled according to $\mathcal{D}_{neg}(x^-) = \mathop{\mathbb{E}}_{c\sim\rho} \mathcal{D}_c(x^-)$. Then, assuming sample access to $\mathcal{D}_{sim}$ and $\mathcal{D}_{neg}$, they choose the representation function $\hat{f}$ from a function class $\mathcal{F}$ that minimizes the empirical version of Eq. (1) with $N = 1$.

Using their framework, Arora et al. are able to show rigorous performance guarantees on binary linear classification when the learned representation $\hat{f}$ is used. Let $L_{sup}(\hat{f})$ denote the supervised loss when the best linear classifier is used; or, more formally,

$$L_{sup}(\hat{f}) = \inf_{W} \; L_{sup}(W\hat{f}).$$

Let L_{un} denote the unsupervised contrastive loss (i.e., Eq. (1)). Arora et al. show that

$$L_{sup}(\hat{f}) \leq \alpha L_{un}(f), \quad \forall f \in \mathcal{F}, \tag{2}$$

where $\alpha \in \mathbb{R}$ is a constant factor that depends on the distribution ρ. In other words, the loss incurred in the downstream task is upper bounded by the unsupervised contrastive loss, so L_{un} is a suitable objective to minimize. Moreover, if the function class $\mathcal{F}$ contains a function with low unsupervised loss, then it will also have low supervised loss using a linear classifier.

While Eq. (2) shows that the unsupervised contrastive loss can upper bound the downstream supervised loss, it leaves open the question: is small unsupervised loss attainable in practical settings? They partially answer this question by studying the price of "class collision" or false negatives, i.e., when a randomly sampled negative comes from the same latent class as the anchor. They decompose the unsupervised contrastive loss into two parts: the loss when the anchor and positive samples come (1) from the same class ($L_{un}^{=}$) and (2) from different classes ($L_{un}^{\neq}$):

$$L_{un}(f) = \tau L_{un}^{=}(f) + (1-\tau) L_{un}^{\neq}(f), \tag{3}$$

where $\tau = \mathop{\mathbb{E}}_{c,c' \sim \rho^2} \mathbf{1}\{c = c'\}$ is the probability of class collision. They then bound $L^{=}_{un}(f)$ as

$$L^{=}_{un}(f) \leq 1 + c' s(f), \quad c' > 0, \tag{4}$$

where

$$s(f) = \mathop{\mathbb{E}}_{c \sim \rho} \left[\sqrt{\|\Sigma(f, c)\|_2} \mathop{\mathbb{E}}_{x \sim \mathcal{D}_c} \|f(x)\| \right] \tag{5}$$

is a notion of intra-class deviation, with $\Sigma(f, c)$ being the covariance matrix of $f(x)$ when $x \sim \mathcal{D}_c$.

Taken together this results in the main theorem:

$$L_{sup}(\hat{f}) \leq L^{\neq}_{un} + \beta s(f) + \eta \text{Gen}_M \tag{6}$$

where Gen_M is a generalization error term with $\text{Gen}_M \to 0$ as $M \to \infty$. If ρ is uniform, then as the number of classes $|\mathcal{C}| \to \infty$, $\eta \to 1$ and $\beta \to 0$. Intuitively, this theorem gives two sufficient conditions for unsupervised contrastive representation learning to work: if $\mathcal{F}$ is rich enough to contain a function f with low $L^{\neq}_{un}(f)$ and low intra-class variance $s(f)$, then $\hat{f}$ will have low supervised linear classification loss.

Follow-up work has used the theoretical framework of Arora et al. to propose and analyze improvements to random negative sampling. We discuss two results which leverage the same basic insight: without ground-truth labels, we don't have access to the true distribution of negatives $p^-_x(\hat{x}) = p(\hat{x} \mid h(x) \neq h(\hat{x}))$; however, we can decompose the marginal p to derive an expression for p^-_x in terms of p^+_x and p, both of which we can estimate empirically:

$$p(\hat{x}) = \tau p^+_x(\hat{x}) + (1 - \tau) p^-_x(\hat{x}) \tag{7}$$

$$\implies p^-_x(\hat{x}) = \frac{p(\hat{x}) - \tau p^+_x(\hat{x})}{1 - \tau} \tag{8}$$

where $p^+_x(\hat{x}) = p(\hat{x}|h(x) = h(\hat{x}))$ and τ is the probability of class collision as defined previously. τ can be computed trivially under a uniform assumption on ρ or estimated empirically from the data.

Instead of sampling negatives from the marginal p (as is common in practice), Chuang et al. [CRYC+20] analyze Eq. (1) when negatives are sampled from p^-_x as defined above. They call the resulting loss function "debiased contrasive loss", which essentially shifts probability mass away from positive examples thereby resulting in fewer false negatives. Their experiments show that the debiased contrastive loss with N negative samples and $M = 1$ positive sample outperforms the standard contrastive loss, and that increasing both M and N improves performance.

Robinson et al. [RCSJ20] extend the Chuang et al. by implementing hard negative mining in unsupervised contrastive representation learning. Similar to the above work, they propose sampling negatives from an alternate distribution q^-_β defined as:

$$q^-_\beta(\hat{x}) = q_\beta(\hat{x}|h(x) \neq h(\hat{x})), \quad \text{where } q_\beta(\hat{x}) \propto e^{\beta f(x)^T f(\hat{x})} p(\hat{x}),\ \beta \geq 0. \tag{9}$$

We note that $q^-_\beta(\hat{x})$ can be derived using the same kind of decomposition as in Eq. (8). The $e^{\beta f(x)^T f(\hat{x})}$ factor adds probability mass to samples in proportion with their similarity to the anchor sample. This results in harder negatives since there is a greater probability of sampling similar negative examples, while still using a debiased loss that mitigates the effect of false negatives. Robinson et al. prove generalization bounds in particular settings and present some empirical evidence which shows that certain downstream tasks are improved when harder negatives are used.

2.2 Conditional Independence

A crucial assumption in the above line of work is that positive samples are conditionally independent given their label. For example, in [AKK+19], positive samples belonging to a latent class c are assumed to be drawn from $\mathcal{D}_{sim}(x, x^+) = \mathop{\mathbb{E}}_{c\sim\rho} \mathcal{D}_c(x)\mathcal{D}_c(x^+)$. Lee et al. [LLSZ21] weaken this assumption to "approximate conditional independence": positive pairs are *approximately* conditionally independent given the label *and* some additional latent variables. However, a key difference in their work is that they study a reconstruction-based objective rather than a contrastive one of the form Eq. (1). For example, consider the task of image inpainting [PKD+16]: Let X_1 be the facade of a building, and let X_2 be a photo of the building with the facade cropped out. Image inpainting trains an encoder-decoder f which accepts an input image with missing regions and reconstructs the entire image, computing a loss over the reconstructed pixels. The reconstruction loss is given by

$$L(f) = \mathop{\mathbb{E}}_{(X_1,X_2)} \left[\|X_1 - f(X_2)\|^2\right], \tag{10}$$

where $f(X_2)$ is masked to only include the reconstructed region. In this formulation, approximate conditional independence presumes that X_1 and X_2 are approximately conditionally independent given the label (building) and some additional latent variables (e.g., the building's architectural style, height, etc.) Under these assumptions, they also derive generalization bounds for a linear image classifier trained on top of the learned reconstruction function f; however, the difference in objectives prevents direct comparison of the bounds.

Finally, HaoChen et al. [HWGM21] fully relax this conditional independence assumption, modeling data dependence with a construction called the "population augmentation graph". The vertices in this graph are image augmentations[1], and two vertices are connected if they can be derived from the same natural image. They assume that there are few edges across ground-truth classes, as images from different classes likely cannot produce the same image augmentation. Additionally, there may be disconnected sub-graphs within a class. Given such a graph, one can derive principled embeddings of image augmentations using spectral graph decomposition; that is, the top-k eigenvectors of the normalized adjacency matrix form a natural embedding matrix $F^* \in \mathbb{R}^{n\times k}$ which captures the spectral clustering of the graph. Creating and decomposing a sufficiently large population graph is not realistic in practice; instead, we can learn the embedding matrix F^* by minimizing a carefully designed loss function over the class of neural nets. Somewhat surprisingly, they show that F^* can be recovered up to a linear transformation by minimizing the following *population spectral contrastive loss*:

$$L(f) = -2 \cdot \mathop{\mathbb{E}}_{x,x^+} \left[f(x)^T f(x^+)\right] + \mathop{\mathbb{E}}_{x,x^-} \left[\left(f(x)^T f(x^-)\right)^2\right], \tag{11}$$

which is similar to the standard contrastive loss. Their main result shows that when the representation dimension k exceeds the maximum number of disconnected sub-graphs, linear classification with the learned representation function will have small error.

[1] See Section 3.1 for a more detailed definition of image augmentation.

3 Application to Images

Unsupervised contrastive representation learning has been shown to be competitive with supervised pre-training methods (e..g. pre-training on ImageNet) [CKNH20]. In this section we survey a common framework for image representation learning, as well as some techniques for addressing relevant computational challenges.

3.1 Image Augmentation

In the general unsupervised setting, it's not obvious how to identify semantically similar inputs without knowing their latent class. However, we can easily circumvent this challenge in the image domain: Given an image, we can derive a similar image by applying transformations like rotation, discoloration, or cropping. The resulting image will have the same "meaning" as the original, yet the surface form (i.e., pixel values) will differ greatly. This process of applying meaning-preserving transformations is referred to as data or image augmentation, and the resulting modified image is often referred to as an "augmentation" of the original.

3.2 Parallel Augmentation Framework

Image augmentation forms the basis of most approaches in image contrastive learning: two augmentations of an image are derived, then a contrastive loss is applied to maximize agreement between the pair, with augmented pairs from other images serving as negative examples. Due to the simplicity and representativeness of their approach, we give an overview of SimCLR [CKNH20].

Fig. 1 depicts the SimCLR framework. Two data augmentation strategies t, t' are sampled from a class of strategies T. (The authors conduct an ablation over a wide set of data augmentation techniques, finding the composition of random cropping and random color distortion to be particularly effective.) These augmentations are applied to yield two different views of an input x. The augmentations are encoded using a base encoder f and then passed through an additional projection layer g. Note that agreement is maximized in the projected space rather than in the

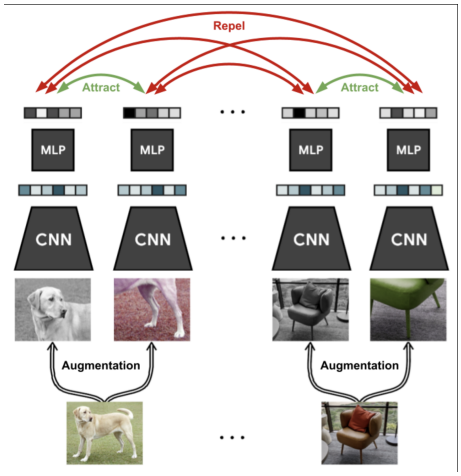


Figure 1: Parallel augmentation framework of SimCLR. Generate two augmentations of an image using various augmentation strategies and encode with an image encoder. For each image, compute contrastive loss on a projection of the representation using intra-batch negatives. Image taken from https://github.com/google-research/simclr.

representation space.

SimCLR reuses augmentation encodings from other examples in the batch as negatives. That is, for each image x in the batch, you create two augmentations; then, you apply a contrastive loss with the augmentations of x as positive examples and the augmentations of every other image x' as negative examples. Notably, the batch size must be large to get sufficiently diverse and hard negatives (the authors train with a default batch size of 4096).

3.3 Caching Strategies

Using an extremely large batch size as in [CKNH20] is often computationally prohibitive. This section covers a few papers which leverage feature representation caching in order to use large numbers of negative examples in the loss computation.

3.3.1 Instance Discrimination with Memory Bank

Instance contrastive learning [WXYL18] treats each sample as an individual class, then aims to learn a classifier using noise contrastive estimation. Computing the denominator of the softmax loss would require the feature representations of each value in the dataset, which is computationally infeasible. To reduce computation, they store image feature representations in a memory bank. Then, they approximate the softmax loss by sampling M random indices and retrieving their representations from the memory bank to use as negatives.

3.3.2 Momentum Contrast

One issue with using a memory bank is that stored image representations become stale as the parameters of the encoder change. Momentum Contrast [HFW+20] alleviates this problem by using (and periodically updating) a separate encoder for cached feature representations. This "momentum encoder" is a weighted combination of past encoders, updated with momentum. The feature representations produced by the momentum encoder are stored in a queue and re-used for negative examples. Training proceeds as follows: an image of query sample x_q is encoded by the query encoder q. An augmentation of x_q is encoded by the key encoder k and added to the queue. Finally, a contrastive loss is applied over the query sample, augmentation, and remaining (negative) representations in the queue.

4 Application to Text

In this section we survey two types of contrastive strategies for learning text representations. *Context prediction strategies* leverage proximity within a document as a signal for semantic similarity. These approaches train a model to distinguish consecutive words or sentences from random alternatives. *Data augmentation strategies* follow the general framework for image contrastive learning: perturb an input text in some way to derive a semantically similar positive example, and apply contrastive loss with other positive pairs serving as negative examples.

4.1 Context Prediction Strategies

Several word and sentence embedding models approximate a contrastive objective through context prediction. The continuous bag-of-words (CBOW) [MSC+13] algorithm predicts an input word given the surrounding context to learn word representations. Negative sampling rephrases this as a binary classification task, distinguishing whether the input word comes from the data or from a noise distribution. The masked-language-modeling objective of BERT [DCLT18] is similar to CBOW, predicting a masked token from the surrounding context, but using more powerful transformer models to learn word representations. ELECTRA [CLLM20] can be seen as the scaled-up, transformer-based version of the CBOW with negative sampling algorithm. Finally, QuickThoughts [LL18] also takes a discriminative approach but at the sentence-level, learning sentence embeddings through context prediction. Given a context sentence and a set of sentences from the same text, the model must identify which sentence follows the context sentence. Fig. 2 shows the training process for QuickThoughts, which learns a separate encoder for the context sentence and for candidate next sentences, then concatenates the representations from both encoders during inference.

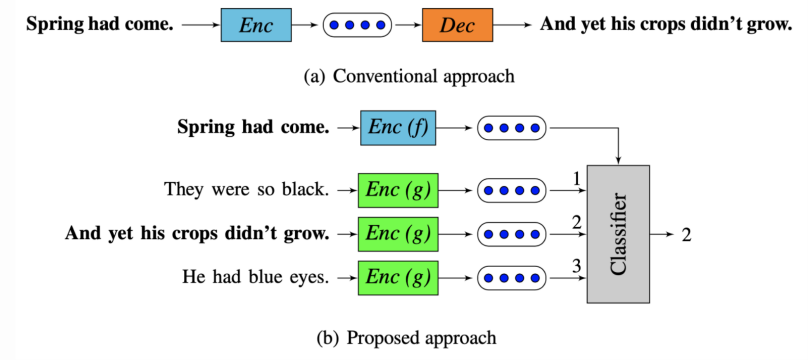


Figure 2: The QuickThoughts training objective maximizes the probability of identifying context (preceding) sentence for each sentence in the training data. At inference, representations from both encoders f and g are concatenated. Image taken from [LL18].

4.2 Data Augmentation Strategies

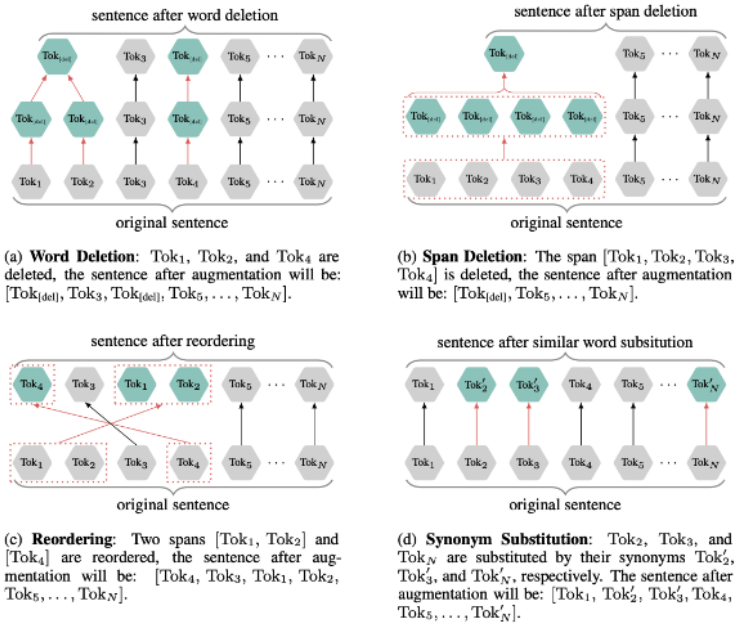
In image contrastive learning, there are many techniques for creating semantically similar images (random cropping, discoloration, blur, etc.). In contrast, there are no obvious, semantic-preserving transformations for sentences. Sentences that are close in surface form or edit distance can have very different semantics (e.g., "I did like the movie" vs. "I did not like the movie"). Nonetheless, the following line of research explores techniques for creating semantically-equivalent augmentations of a sentence or text.

4.2.1 Evaluation Method

The following papers extend or alter the pre-training of transformer-based language models to include a contrastive objective. To evaluate their pre-training methods, the authors use the GLUE [WSM+19] and/or SentEval [CK18] benchmarks. Both are collections of natural language understanding tasks designed to assess the quality of sentence representations through classification or regression tasks at the sentence level. Evaluation in the following papers is done by finetuning a given language model with an added output layer on the supervised dataset for particular task, then evaluating on the test set.

4.2.2 Contrastive LEArning for sentence Representation (CLEAR)

Given a sentence, CLEAR [WWG+20] generates a similar sentence by applying one or more edit strategies: deletion of a random word or span, synonym-substitution, or re-ordering. These strategies are depicted in Fig. 3.



(a) **Word Deletion**: Tok_1, Tok_2, and Tok_4 are deleted, the sentence after augmentation will be: $[\text{Tok}_{[\text{del}]}, \text{Tok}_3, \text{Tok}_{[\text{del}]}, \text{Tok}_5, \ldots, \text{Tok}_N]$.

(b) **Span Deletion**: The span $[\text{Tok}_1, \text{Tok}_2, \text{Tok}_3, \text{Tok}_4]$ is deleted, the sentence after augmentation will be: $[\text{Tok}_{[\text{del}]}, \text{Tok}_5, \ldots, \text{Tok}_N]$.

(c) **Reordering**: Two spans $[\text{Tok}_1, \text{Tok}_2]$ and $[\text{Tok}_4]$ are reordered, the sentence after augmentation will be: $[\text{Tok}_4, \text{Tok}_3, \text{Tok}_1, \text{Tok}_2, \text{Tok}_5, \ldots, \text{Tok}_N]$.

(d) **Synonym Substitution**: Tok_2, Tok_3, and Tok_N are substituted by their synonyms Tok'_2, Tok'_3, and Tok'_N, respectively. The sentence after augmentation will be: $[\text{Tok}_1, \text{Tok}'_2, \text{Tok}'_3, \text{Tok}_4, \text{Tok}_5, \ldots, \text{Tok}'_N]$.

Figure 3: Edit strategies used in CLEAR. Image taken from [WWG+20].

Fig. 4 shows the overall approach for CLEAR. First, one or more edit strategies are applied to the input sentence s, yielding two positive augmentations $\tilde{s}_1, \tilde{s}_2$. These augmented sentences are encoded using a transformer-based encoder. Following SimCLR [CKNH20], these sentence representations are mapped to a latent space with an additional feedforward neural network g. Finally, a contrastive loss is applied to maximize agreement between the representations in this latent space for a pair of sentence augmentations. Training utilizes intra-batch negatives and the n-pair multiclass loss.

In addition to using a contrastive loss, the authors use the masked language modeling (MLM) loss of the original BERT objective. The overall training loss is the sum of these two losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{CL}} \tag{12}$$

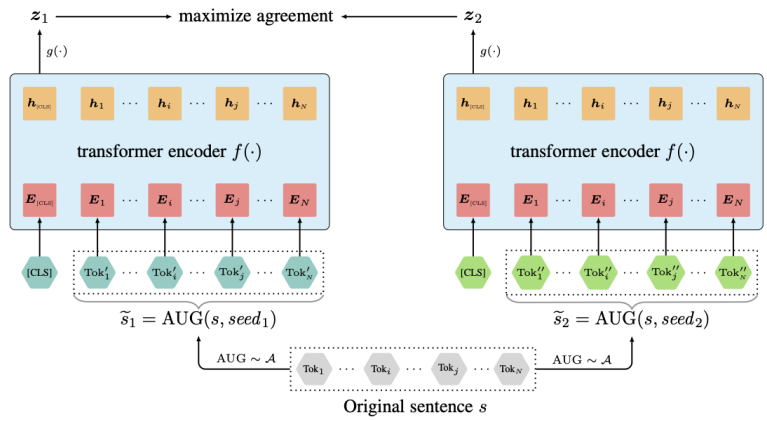


Figure 4: The data augmentation workflow of CLEAR. First, generate two different augmentations of the original sentence using two random seeds. Encode both using a transformer-based language encoder, then apply contrastive loss batch-wise. Image taken from [WWG+20].

The authors evaluate models trained on this joint loss against baseline transformer models on the GLUE dev set, demonstrating that models trained with the joint MLM + contrastive loss outperform the baseline models trained only with an MLM objective across all GLUE tasks. Further, they show that different edit strategies (and combinations of edit strategies) work better for different downstream tasks. The authors also evaluate against a subset of the SentEval benchmark: namely, the semantic textual similarity (STS) datasets. For all of these tasks, contrastive loss improves over the baseline. The authors hypothesize that this is because the objective of contrastive learning (identifying similar instances) aligns well with the STS task.

4.2.3 CERT: Contrastive Self-supervised Learning for Language Understanding

CERT [FWZ+20] generates sentence augmentations by deriving paraphrases of a sentence using back-translation. Fig. 5 illustrates the data augmentation strategy of CERT using back-translation. Back-translation is a technique for generating a paraphrase of a text by first translating it to another language, then translating it back. Given a sentence x in source language S, we can translate it into a target language T to get a translation x' of x. Then, we can translate x' back to S to get a paraphrase x'' of x. We expect the translations to be sufficiently aligned to ensure that x and x'' are semantically similar, but different enough that they will vary in surface form. The authors use German and Chinese as the target languages to generate two augmentations per sentence.

Figure 5: The data augmentation workflow of CERT using back-translation. Image taken from [FWZ+20].

CERT uses MoCo [HFW+20] to implement their contrastive training. First, they pretrain a transformer-based language model (e.g. BERT) on some large-scale input text. Then, they

continue training the model using their contrastive objective to derive the CERT model. Finally, the CERT model is finetuned and evaluated on a downstream task. They evaluate CERT on the GLUE benchmark, with CERT outperforming BERT on 7 tasks, underperforming on 2 tasks, and matching performance on 2 tasks.

4.2.4 DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations

DeCLUTR [GNBW20] generalizes the QuickThought approach beyond the sentence boundary to encode arbitrary-length spans of text. The main hypothesis is that nearby spans of text will be similar in meaning, while distant spans will be less similar. The data augmentation process proceeds by sampling an anchor span of text, then sampling one or more shorter, nearby spans of text as positive examples. Other key differences with QuickThought are: they sample one or more positive spans per anchor (rather than strictly one), and an anchor text may overlap with or subsume a positive span (rather than strictly being adjacent).

The authors take a pre-trained transformer-based language model and continue its training using this contrastive objective. The final model is evaluated on the SentEval benchmark. They compare against word-embedding baselines (GloVe, fasttext) as well as sentence embedding models (InferSent, USE, and Sent. Transformers). DeCLUTR models consistently outperform the underlying pre-trained language model. They underperform relative to supervised/semi-supervised pretraining methods, but this is unsurprising given the additional level of supervision.

5 Concluding Remarks

In this work, we surveyed recent progress in unsupervised contrastive representation learning. Existing theoretical work gives performance guarantees on downstream linear classification using various contrastive objectives. Future theoretical work could include a more detailed comparison of the objectives and generalization bounds in Arora et al., Lee et al., and HaoChen et al. We also surveyed applied results for image and text representation learning. A key strategy in both domains is to generate different views of an input which preserve its underlying semantics. This is straightforward to do for images and has been exploited to great success; analogous strategies for text representation learning are still being explored.

References

[AKK+19] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019. [pp. 3, 5]

[CK18] Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*, 2018. [p. 9]

[CKNH20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. [pp. 6, 7, 9]

[CLLM20] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020. [p. 8]

[CRYC+20] Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020. [p. 4]

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. [p. 8]

[FWZ+20] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*, 2020. [p. 10]

[GNBW20] John M Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*, 2020. [p. 11]

[HFW+20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. [pp. 7, 10]

[HWGM21] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34, 2021. [p. 5]

[LL18] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018. [p. 8]

[LLSZ21] Jason D Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021. [p. 5]

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. [p. 8]

[PKD+16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. [p. 5]

[RCSJ20] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020. [p. 4]

[WSM+19] 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 *International Conference on Learning Representations*, 2019. URL: https://openreview.net/forum?id=rJ4km2R5t7. [p. 9]

[WWG+20] 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. [pp. 9, 10]

[WXYL18] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. [p. 7]