A Primer on Contrastive Pretraining in Language Processing: Methods, Lessons Learned and Perspectives

Nils Rethmeier\textsuperscript{1,2}, Isabelle Augenstein\textsuperscript{2}

\textsuperscript{1}German Research Center for AI, Berlin, Germany
\textsuperscript{2}University of Copenhagen, Copenhagen, Denmark
nils.rethmeier@dfki.de, augenstein@di.ku.dk

Abstract

Modern natural language processing (NLP) methods employ self-supervised pretraining objectives such as masked language modeling to boost the performance of various application tasks. These pretraining methods are frequently extended with recurrence, adversarial or linguistic property masking, and more recently with contrastive learning objectives. Contrastive self-supervised training objectives enabled recent successes in image representation pretraining by learning to contrast input-input pairs of augmented images as either similar or dissimilar. However, in NLP, automated creation of text input augmentations is still very challenging because a single token can invert the meaning of a sentence. For this reason, some contrastive NLP pretraining methods contrast over input-label pairs, rather than over input-input pairs, using methods from Metric Learning and Energy Based Models. In this survey, we summarize recent self-supervised and supervised contrastive NLP pretraining methods and describe where they are used to improve language modeling, few or zero-shot learning, pretraining data-efficiency and specific NLP end-tasks. We introduce key contrastive learning concepts with lessons learned from prior research and structure works by applications and cross-field relations. Finally, we point to open challenges and future directions for contrastive NLP to encourage bringing contrastive NLP pretraining closer to recent successes in image representation pretraining.

1 Introduction

Current downstream machine learning applications heavily rely on the effective pretraining of representation learning models. Contrastive learning is one such technique which enables pretraining of general or task-specific data encoder models in a supervised or self-supervised fashion to increase the downstream performance of language or image representations. While contrastive pretraining in computer vision has enabled the recent successes in self-supervised image representation pretraining, the benefits and best practices of contrastive pretraining in natural language processing (NLP) are still comparatively less established [Jaiswal et al., 2021]. However, there is a first line of works on contrastive NLP methods which show strong performance and data-efficiency benefits of (self-)supervised contrastive NLP pretraining as illustrated in Fig. 1. For example, supervised contrastive pretraining enables zero-shot prediction of unseen text classes and improves few-shot performance [Pappas and Henderson, 2019]. Moreover, task-agnostic self-supervised contrastive pretraining systems have been shown to improve language modeling [Logeswaran and Lee, 2018; Clark et al., 2020; Wu et al., 2020; Giorgi et al., 2020], while [Rethmeier and Augenstein, 2020] develop a data-efficient contrastive pretraining method for improved zero-shot and long-tail learning. Others propose task-specific contrastive self-supervision for pronoun disambiguation [Klein and Nabi, 2020], discourse representation learning [Iter et al., 2020], text summarization [Duan et al., 2019] and other NLP tasks, as we will describe in §3.
Contributions: In this primer to contrastive pretraining, we therefore summarize recent (self-)supervised contrastive NLP pretraining methods and describe how they enable zero-shot learning and improve language modeling, few-shot learning, pretraining data-efficiency or rare event prediction. We cover basic concepts and crucial design lessons of contrastive NLP, while detailing the resulting benefits such as zero-shot prediction and efficient training. Then, we structure existing research as supervised or self-supervised contrastive pretraining and explain connections to energy based models (EBMs), since many works refer to EBM. Finally, we point out open challenges and outline future and underrepresented research directions in contrastive NLP pretraining.

2 Contrastive Learning Concepts and Benefits
At the core of contrastive methods is the idea of learning to contrast between pairs of similar and dissimilar data points. A pair of similar data points is called a positive sample if both data points are different representations or views of the same data instance. Negative samples are pairs where the two data points are of different instances. For contrastive learning, such data points can either be input-input \((x_i, x_j)\) or input-output \((x_i, y_j)\) pairs. While contrastive computer vision methods learn from input-input (image-image) pairs \((x_i, x_j)\) [Jaiswal et al., 2021; Chen et al., 2020b], NLP methods additionally use input-output (text, label) pairs \((x_i, y_j)\). Here \(x_i\) are input text embeddings, while \(y_j\) are label embeddings of a short text that describes a label, i.e. an extreme summarization of the input text to get two views of said text.

2.1 Noise Contrastive Estimation (NCE)
Noise contrastive estimation is the objective used by most contrastive learning approaches within NLP. Thus, we briefly outline its main variants and the core ideas behind them, while pointing to [Ma and Collins, 2018] for detailed, yet readily understandable explanations of the two main NCE variants. Both variants can intuitively be understood as a sub-sampled softmax with \(K\) negative samples \(a^i\) and one positive sample \(a^+_i\). The first variant expresses NCE as a binary objective (loss) in the form of maximum log likelihood, where only \(K\) negatives are considered.

\[
L_B(\theta, \gamma) = \log \sigma(s(x_i, a^+_i; \theta), \gamma) \tag{1}
\]

Here, \(s(x_i, a^+_i; \theta)\) is a scoring or similarity function that measures the compatibility between a single text input \(x_i\) and another sample \(a^+_i\). As mentioned above, the sample can be another input text or an output label (text), thus modeling NLP tasks as ‘text-to-text’ prediction similar to language models. The similarity function is typically a cosine similarity, a dot product or a logit (unscaled activation) produced by a input-sample matcher sub-network [Rethmeier and Augenstein, 2020]. The \(\sigma(z, \gamma)\) is a scaling function, which for use in eq. (1) is typically the sigmoid \(\sigma(z) = \frac{\exp(z - \gamma)}{(1 + \exp(z - \gamma))}\) with a hyperparameter \(\gamma \geq 0\) (temperature), that is tuned or omitted depending on the way that negative samples \(a^i\) are attained.

The other NCE objective learns to rank a single positive pair \((x_i, a^+_i)\) over \(K\) negative pairs \((x_i, a^i)\):

\[
L_R(\theta) = \frac{e^{\bar{s}(x_i, a^+_i; \theta)}}{e^{\bar{s}(x_i, a^+_i; \theta)} + \sum_k^{K} e^{\bar{s}(x_i, a^i; \theta)}} \tag{2}
\]

Here, to improve \(L_B\) or \(L_R\) performance, [Ma and Collins, 2018] propose a regularized scoring function \(\bar{s}(x_i, a^+_i) = s(x_i, a^+_i) - \log p_N(a^+_i)\) that subtracts the probability of the current sample \(a^+_i\) under a chosen noise distribution \(p_N(a^+_i)\). In practice, the noise distribution can be set to 0 [Mnih and Teh, 2012; Wu et al., 2020; Rethmeier and Augenstein, 2020] to save on computation. To robustly learn word embeddings, \(p_N(a^+_i)\) can be set as the word probability \(p_{\text{word}}\) in a corpus [Mikolov et al., 2013b], or as the probability of a sequence under a language model \(p_{LM}\) [Deng et al., 2020], when learning contrastive sequence prediction.

Generalization to an arbitrary number of positives: As [Khosla et al., 2020] mention, original contrastive formulations use only one positive pair per text instance (see e.g. [Mikolov et al., 2013b; Logeswaran and Lee, 2018]), while more recent methods mine multiple positives or use multiple gold class annotation representations for contrastive learning [Rethmeier and Augenstein, 2020; Qu et al., 2021]. This means that e.g. the positive term in eq. (1) becomes \(\sum_p^P \log \sigma(s(x_i, a^+_i; \theta, \gamma))\) to consider \(P\) positives.

Importance of negative sampling semantics and lessons learned: How positive and negative samples are generated or sampled is a key component of effective contrastive learning. [Saunshi et al., 2019] prove and empirically validate that “sampling more negatives improves performance, but only if they are sampled from the same context or block of information such as the same paragraph”. Such hard to contrast (classify) negatives, are sampled in most works [Mikolov et al., 2013b; Saunshi et al., 2019; Rethmeier and Augenstein, 2020; Iter et al., 2020]. Otherwise, performance can deteriorate due to weak contrast learning of conceptually related classes. Additionally, [Rethmeier and Augenstein, 2020] find that both positive and negative contrastive samples from a long-tail distribution are essential in predicting rare classes and in substantially boosting zero-shot performance, especially over minority classes. [Mikolov et al., 2013b] undersample negatives of frequent words to stabilize pretraining of word embeddings to a similar effect. Additional practical advice for negative sampling is mentioned in 3.1.

2.2 Contrastive Learning as Mutual Information Maximization, Inverse Data Generation and Energy Based Models:
Contrastive learning methods are closely related to at least four machine learning concepts. First, InfoNCE has been shown to maximize the lower bound of mutual information between different views of the data [van den Oord et al., 2018; Hjelm et al., 2019]. Second, [Zimmermann et al., 1992]
2021], show that contrastive learning robustly inverts a data

generation process "by uncovering the true generative factors
of variation underlying the observational data, even in prac-
tical cases, where most theoretical assumptions of the gener-
ation process are severely violated.". Third, learning similar-
ities in data connects contrastive learning to metric learning
[Musgrave et al., 2020]. Finally, many works describe con-
trastive learning as an Energy Based Model, EBM, and since
this may initially be unfamiliar, we outline popular EBM varia-
tions for supervised and self-supervised contrastive text pre-
training below.

**Input-output contrastive EBM:** The binary NCE variant
from eq. (1) is a special case of a “Contrastive Free Energy”
loss as described in [Lecun et al., 2006] Fig. 6b or in [LeCun
and Huang, 2005] Fig. 2 and Sec. 3.3 as the negative log-
likelihood loss with negative sampling. [Lecun et al., 2006]
originally state that an EBM \( E \) learns the compatibility
between input-output pairs \((x_i, y_i)\) with \(x_i \in X\) and \(y_i \in Y\)
\[
E(X, Y) = E(W, X, Y)
\]
where \(W\), or \(\theta\) in eq. (1), are model parameters that
encode inputs \(X\) and labels \(Y\). Here, \(X\) and \(Y\) are
views or augmentations of either the same data point (posi-
tives), or different data points (negatives). The energy func-
tion \(E\) measures the compatibility between its param-
ters \((X, Y)\), where \(E(\cdot)=0\) indicates optimal com-
patibility – e.g. \(E(X=Tiger, Y=felidae)=0\) means \(X\)
and \(Y\) match. Note that in the probabilistic framework
\(P(Y=felidae|X=Tiger, W)=1\). Works which use input-
output noise contrastive estimation are [Pappas and Hend-
erson, 2019; Rethmeier and Augusten, 2020], visualized in
Fig. 2. They encode an input text \(x_i\) using a text-encoder
\(T\) and a label description \(y\) via a separate label-encoder \(L\)
to then concatenate both into a single text input-output en-
coding pair \((T(x_i), L(y_i))\). Once encoded, the input-label
pair similarity is learned via a binary NCE objective \(L_B\) as

\[
L_B(a_i, a_i^*) = \exp(-E(x_i, y_i)) - \sum_{a_i^* \neq a_i} \exp(-E(x_i, a_i^*))
\]

in eq. (1). Compared to input-input models described be-
low, these approaches allow for encoding a large number of
augmented views, i.e. labels, very compute efficiently via a
small label-encoder. This allows them to scale to large sam-
ple sizes of positives and negatives, which is crucial to suc-
cessful contrastive learning. While [Pappas and Henderson,
2019] use this formulation for supervised-only pretraining on
label encodings, [Rethmeier and Augusten, 2020] addition-
ally sample input words \(x_i \in X\) as pseudo-label encodings
\(y_i^\prime=L(x_i)\) for efficient contrastive self-supervised pre-
training. Thus, the later approach unifies supervision and self-
supervision as a single task of contrasting real-label encod-
ings \(L(y_i)\) or pseudo-label encodings \(y_i^\prime=L(x_i)\). The advan-
tage of such methods is that once the NCE classifier is pre-
trained, it can be reused, i.e. zero-shot transferred, to any
downstream task, without having to initialize a new classifier.
In fact, unified prediction and zero-shot transfer are prop-
ties one would expect to have from pretraining, since most
NLP tasks fit into a ‘text-to-text’ prediction description. As
a result of contrastive pseudo-labels, input-output methods
enable efficient contrastive self-supervised pretraining [Reth-
meier and Augusten, 2020], even on very small data, with
commodity hardware, and without complicated mechanisms
like cyclic learning rate schedules, residual layers, warmup,
specialized optimizers or normalization which current large-
data pretraining approaches require as research summarized
in [Mosbach et al., 2021] shows. Finally, many input-input
contrastive methods rely on re-pretraining already otherwise
pretrained Transformer architectures [Fang and Xie, 2020;
Deng et al., 2020; Giorgi et al., 2020], since encoding aug-
mented inputs is costly in current Transformer architectures.

**Input-output contrastive EBM:** Input-output methods con-
trast input texts \(X\) from augmented input texts \(X^\prime\) rather than
from labels \(Y\) – see Fig. 3. For example, [Clark et al., 2020]
replace a subset of input text words \(x_i, w\) with other words
\(x_i, w^\prime\) sampled from the vocabulary for self-supervised con-
trastive pretraining. The original text \(x_i\) is augmented into a
text \(a_i\) to provide a positive sample augment \(a_i^+\) or a negative

---

**Figure 2:** Contrastive input-output \((X, Y)\) pretraining. Texts and
labels are encoded independently via a medium sized text encoder
and a very small label-encoder. This encodes 1 text for \(n\) labels with
minimal computation to enable large-scale \(K\) negative sampling.

**Figure 3:** Contrastive input-input \((X, X^\prime)\) Pretraining: Input-
output methods contrast an original text with augmented positive \(a_i^+\)
and negative \(a_i^-\) texts \(a_i \in X^\prime\), which requires more computation
than input-output methods. Achromatism compatible.
sample augment \( a_i^- \). Self-supervised pretraining then contrasts pairs \((x_i, a_i)\) of original texts against augmented ones via the binary NCE as in eq. (1). Similar to the EBM in eq. (3) this can be summarized as
\[
E(X, X') \lor E(W, X, X')
\]
(4)
As mentioned, current input-input contrast models are hampered by compute-intense augmentation encoding \( W(a_i) \).

**Contrastive pretraining enables zero-shot learning, improves few-shot learning and increases parameter learning efficiency:** [Radford et al., 2021] replace a Transformer by a CNN to speed up self-supervised zero-shot prediction learning by a factor of 3, and add text contrastive pretraining to speed up learning by another factor of 4. [Pappas and Henderson, 2019] show that supervised contrastive pretraining enables supervised zero-shot and improved few-shot learning. [Rethmeier and Augustein, 2020] run self-supervised contrastive pretraining for unsupervised zero-shot prediction, i.e. without human annotations, and show that this boosts learning performance on long-tail classes. This is done while pretraining on only portions of an already very small text collection of 6 to 60MB of pretraining text. They also demonstrate that rather than adding more data during pretraining, one can also increase self-supervised learning signals instead.

3 Self- or Supervised Contrastive Pretraining

The goal of contrastive pretraining is to initialize model weights for efficient zero-shot transfer or fine-tuning to downstream tasks. Pretraining is either supervised or self-supervised. Supervised contrastive pretraining methods use corpora of hand-annotated data such as paraphrased parallel sentences, textual labels or text summarizations to define text data augmentations for contrastive pretraining. Self-supervised contrastive methods aim to scale pretraining by contrasting automatically augmented input texts \( X' \) or textual output pseudo-labels \( Y' \sim P(X) \) – see §2.2 for input-input vs. input-output contrastive methods. Both self-supervised and supervised contrastive methods are used to train language encoder models from scratch, or can ‘re-pretrain’ or fine-tune an already otherwise pretrained model such as a RoBERTa [Liu et al., 2019]. Below, we structure self- and supervised contrastive pretraining by technique and application.

3.1 Self-supervised Contrastive Pretraining

**Input-input contrastive text representation pretraining via automated text augmentation:** Fig. 3 compares methods that use input-input contrastive (EBM) learning as overviewed in §2.2. [Qu et al., 2021] use a contrastive momentum encoder over combinations of recently proposed text data augmentations like “cutoff, back translation, adversarial augmentation and mixup”. They find that mixing augmentations is most useful when the augmentations provide sufficiently different views of the data. Further, since constructing text augmentations which do not alter the meaning (semantics) of a sentence is very difficult, they introduce two losses to ensure both sufficient difference and semantic consistency of sentence augmentations. They define a consistency loss to guarantee that augmentations lead to similar predictions \( y_c \) and a contrastive loss that makes augmented text representations \( a_i \) similar to the original text \( x_i \). To ensure that a sufficiently large amount of negative text augmentations are sampled, they use an augmentation-embedding memory bank. [Fang and Xie, 2020] only use back-translation. [Wu et al., 2020; Meng et al., 2021] investigate other sentence augmentation methods, [Giorgi et al., 2020] contrast text spans, [Clark et al., 2020; Meng et al., 2021] replace input words by re-sampling a language model and [Simoulin and Crabbé, 2021] investigate contrastive sentence structure pretraining. Finally, [Meng et al., 2021] also contrasts cropped sentences after augmentation via word re-sampling.

**Contrasting Next or Surrounding Sentence (or Word) Prediction (NSP, SSP)** Sentence prediction is a popular input-input contrastive method as in §2.2. Next sentence prediction, NSP, and surrounding sentence prediction, SSP, take inspiration from the skip-gram model [Mikolov et al., 2013b], where surrounding and non-surrounding words are contrastively predicted given a central word to learn word embeddings using an NCE §2.1 variant [Mikolov et al., 2013b]. Methods mostly differ in how they sample positive and negative sentences, where negative sampling strategies such as undersampling frequent words, in [Mikolov et al., 2013a], are crucial. [Logeswaran and Lee, 2018] propose contrastive NSP to predict the next sentence as a positive sample against \( n \) random negative sample sentences. Instead of generating the next sentence, they learn to discriminate which sentence encoding follows a given sentence. This allows them to train a better text encoder model with less computation, but sacrifices the ability to generate text. [Liu et al., 2019] investigate variations of the contrastive NSP objective used in the BERT model. The method contrasts a consecutive sentence as a positive text sample against multiple non-consecutive sentences from other documents as negative text samples. They find that sampling negatives from the same document during self-supervised BERT pretraining is critical to downstream performance, but that removing the original BERT NSP task improves downstream performance. [Iter et al., 2020] find that predicting surrounding sentences in a \( k \)-sized window around a given central anchor sentence “improves discourse performance of language models”. They sample surrounding sentences: (a) randomly from the corpus to construct easy negatives, and (b) from the same paragraph, but outside the context window as hard (to contrast) negative samples. Contextual negative sampling is theoretically and empirically proven by [Saunshi et al., 2019], who demonstrate that: “increased negative sampling only helps if negatives are taken from the original texts’ context or block of information”, i.e. the same document, paragraph or sentence. [Aroca-Ouellette and Rudzicz, 2020] study how to combine different variants of the NSP pretraining tasks with non-contrastive, auxiliary self-supervision signals, while [Simoulin and Crabbé, 2021] explore contrastive sentence structure learning.

**Input-output contrastive text representation pretraining:** In Fig. 2 [Rethmeier and Augustein, 2020] use output label embeddings as an alternative view \( Y \) (labels) of text input embeddings \( X \) for contrastive learning of (dis)-similar text-label embedding pairs \((X, Y)\) via binary NCE from §2.1. Using
a separate label and text encoder allows them to efficiently compute many negative label samples, while encoding the text X only once, unlike input-input view methods in Fig. 3. They pretrain with random input words as pseudo-labels for self-supervised pretraining on a very small corpus, which despite the limited pretraining data enables unsupervised zero-shot prediction, largely improved few-shot and markedly better rare concept (long-tail) learning.

**Distillation:** [Sun et al., 2020] propose CoDIR, a contrastive language model distillation method to pretrain a smaller student model from an already pretrained larger teacher such as a Masked Transformer Language Model. Compressing a pretrained language model is challenging because nuances such as interactions between the original layer representation are easily lost – without noticing. For distillation, they extract layer representations from both the large teacher and the small student network over the same or two different input texts, to create a student and teacher view of said texts. Using the constrastive InfoNCE loss [van den Oord et al., 2018b], they then learn to make the student representation similar to teacher representations for the same input texts, and dissimilar if they receive different texts. The score or similarity function in InfoNCE is measured as the cosine distance between mean pooled student and teacher Transformer layer representations. For negative sampling in pretraining, they use text inputs from the same topic, e.g., a Wikipedia article, to mine hard negative samples – i.e., they sample views from similar contexts as recommended for contrastive methods in [Saunshi et al., 2019].

**Text generation as a discriminative EBM:** [Deng et al., 2020] combine an auto-regressive language model, with a contrastive text continuation EBM model for improved text generation. During pretraining, they learn to contrast real data text continuations and language model generated text continuations via conditional NCE from §2.1. For generation, they sample the top-k text completions from the auto-regressive language model and then score the best continuation via the trained EBM, to markedly improve model perplexity. However, the current approach is computationally expensive.

**Cross-modal contrastive representation pretraining:** Representations for zero-shot image classification can be pretrained using image caption text for contrastive self-supervised pretraining. [Jia et al., 2021] automatically mine a large amount of noisy text captions for images in ALIGN, to then noise-filter and use them to construct matching and mismatching pairs of image and augmented text captions for contrastive training. [Radford et al., 2021] use the same idea in CLIP, but pretrain on a large collection of well annotated image caption datasets. Both methods allow for zero-shot image classification and image-to-text or text-to-image generation, and are inherently zero-shot capable. [Radford et al., 2021] also run a zero-shot learning efficiency analysis for CLIP and find two things. First, that using a data efficient CNN text encoder increases zero-shot image prediction convergence 3-fold compared to a Transformer text encoder, which they state to be computationally prohibitive. Second, they find that adding contrastive self-supervised text pretraining increases zero-shot image classification performance 4-fold. Thus, CLIP [Radford et al., 2021] shows that contrastive self-supervised CNN text encoder pretraining can substantially outperform current Transformer pretraining methods, while ALIGN [Jia et al., 2021] also automates the image and caption data collection process to increase data scalability.

### 3.2 Supervised Contrastive Pretraining

**Input-input contrastive supervised text representation pretraining** [Pappas and Henderson, 2019] train a two-input-lane Siamese CNN network, which encodes text as the input view $x_i$ in one lane, and labels via a label encoder in a second data view $y_i$, to learn to contrast pairs of $(x_i, y_i)$ as similar (1) or not (0). Rather than encoding labels as multi-hot vectors such as $[0, 1, 0, 0, 1]$, they express each label by a textual description of said label. These textual label descriptions can then be encoded by a label encoder subnetwork, which in the simplest case constructs a label embedding by averaging over the word embeddings of the words that describe a label. However, this requires manually describing each label. Using embeddings of supervised labels, they pretrain a contrastive text classification network on known positive and negative labels, and later apply the pretrained network to unseen classes for zero-shot prediction. Their method thus provides supervised, but zero-shot capable pretraining. While [Rethmeier and Augenstein, 2020] also support supervised contrastive input-output pretraining, they automate label descriptions construction, and conjecture that in real-world scenarios, most labels, e.g., the word ‘elephant’, are already part of the input vocabulary and can thus be pretrained as word embeddings via methods such as Word2Vec [Mikolov et al., 2013a]. They also note that: “once input words are labels, one can sample input words as pseudo label embeddings for contrastive self-supervised pretraining”, as described in section §3.1. Either method is contrastively pretrained via binary NCE as described in §2.1. Furthermore, both methods markedly boost few-shot learning and enable zero-shot predictions, while [Rethmeier and Augenstein, 2020] enables unsupervised zero-shot learning via self-supervised contrastive pretraining. The added contrastive self-supervision further boosts few-shot and long-tailed learning performance, while also increasing convergence speed over supervised-only contrastive learning in [Pappas and Henderson, 2019].

**Contrasting input views on manual text augmentation:** [Klein and Nabi, 2020] use contrastive self-supervised pretraining to refine a pretrained BERT language model to drastically increase performance on pronoun disambiguation and the Winograd Schema Commonsense Reasoning task. Their method contrasts over candidate trigger words that affect which word a pronoun refers to. They first mine trigger word candidates from text differences in paraphrased sentences and then maximize the contrastive margin between candidate pair likelihoods. This implicitly pretrains a model for common sense concepts, and is similar to contrastive self-supervision in vision [Chen et al., 2020b], with the difference of the latter generating contrastable data augmentations for a given sample. While general pretraining provides little pronoun disambiguation learning signal, their method demonstrate the de-
sign of task-specific contrastive learning to produce strong performance increases in un- and supervised commonsense reasoning.

Contrastive text summarization: [Duan et al., 2019] use a Transformer attention mechanism during abstractive sentence summarization learning to optimize two contrasting loss objectives. One loss maximizes the contributions of tokens with the most attention when predicting the summarized sentence. The other loss is connected to a second decoder head, which learns to minimize the contribution of the attention to other, non-summarization relevant, tokens. This method can perhaps best be understood as contrastive, layer attention noise reduction. The main draw back of this method is the current dual network head prediction, which introduces a larger complexity compared to other contrastive methods.

Cross and multi-modal supervised contrastive text pretraining for representation learning: Recent work from computer vision and time series prediction train with contrastive supervised losses to enable zero-shot learning or improve data-to-text generation. [Jiang et al., 2019] fuse image and text description information into the same representation space for generalized zero-shot learning – i.e. where at test time some classes are unseen, zero-shot, while other classes were seen during training. To do so, they first pretrain a supervised text-image encoder network to contrast (image, text, label) triplets of human annotated image classes. At test time, this contrastive network decides which text description best matches a given image. This works for seen and unseen classes, because classes are represented as text descriptions. [Radford et al., 2021] pretrain on manually annotated textual image descriptions to enable better generalization to unseen image classes. [Uehara et al., 2020] turn stock price value time series into textual stock change descriptions where the contrastive objectives markedly increase the fluency and non-receptiveness of generated texts, especially when trained with little data.

Datasets construction for contrastive pretraining: [Raganato et al., 2019] automatically create a corpus of contrastive sentences for word sense disambiguation in machine translation by first identifying sense ambiguous source sentence words, and then creating replacement word candidates to mine sentences for contrastive evaluation.

4 Challenges and Potential Directions

Challenge: need for many negatives. Current methods require the sampling of many negative instances for contrastive learning to work well. There is work on the benefits and harms of sampling hard to contrast negatives [Cai et al., 2020], or relevant negatives [Saunshi et al., 2019], which can boost sampling efficiency. However, as seen in [Mikolov et al., 2013b; Rethmeier and Augenstein, 2020] depending on the task, sampling diverse negatives can play an important role. To date, the importance of easy to contrast negative samples is underexplored, but insights from a metric learning survey by [Musgrave et al., 2020], suggest that hard, medium and easy samples may all be necessary, especially for generalization in open class set tasks such as pretraining.

Challenge and directions: text augmentation quality and efficiency: Self-supervised text augmentation research in NLP (§3.1) is gaining momentum and [Qu et al., 2021; Chen et al., 2020a] and many others analyze using mixes of recent text data augmentations. However, these input-output contrastive methods often use computationally expensive or non-robust mechanisms like: back translation, initializing a new prediction head per downstream task, or reliance on already otherwise pretrained models like RoBERTa. Works on input-output contrastive learning like [Pappas and Henderson, 2019; Rethmeier and Augenstein, 2020] nullify these requirements and demonstrate very data efficient pretraining, which is currently an under-researched, but very desirable property of contrastive learning. [Zimmermann et al., 2021] further solidify these insights and show that contrastive methods effectively recover data properties even from small data sets. While many self-supervised contrastive pretraining methods rely on already pretrained Transformers, works [Rethmeier and Augenstein, 2020; Clark et al., 2020; Wu et al., 2020; Meng et al., 2021] make important contributions by removing this restriction. [ Wu et al., 2020; Iter et al., 2020] propose robustly scalable input augmentation, while [Grill et al., 2020] propose BYOL, which does not require negative sampling, and potentially lends itself improving to future contrastive NLP methods.

Challenge: under-researched applications: [Deng et al., 2020] enhance a text generation language model with contrastive importance resampling of language model generated text continuations. [Duan et al., 2019] propose contrastive abstractive sentence summarization, which using Momentum Contrast can potentially improve on.

Direction: cross-modal generation: An underresearch direction for contrastive NLP are data-to-text tasks that turn non-text inputs into a textual description. For example [Uehara et al., 2020] contrastively learn to generate stock change text descriptions from stock price time series using limited data, while works like [Radford et al., 2021; Jia et al., 2021] show that contrastive text supervision and self-supervision can multiply the zero-shot learning efficiency in cross-modal representation learning.

Direction: contrastive (language) model fusion: While [Sun et al., 2020] compress a large language model, which future work can adapt to fuse multiple language model or mutually transfer knowledge between models.

Direction: commonsense contrastive learning: The contrastive word sense disambiguation (WSD) dataset construction method by [Raganato et al., 2019] is potentially adaptable to automatically mine inputs for the contrastive pronoun learning method by [Klein and Nabi, 2020].

5 Conclusion

In this primer on contrastive pretraining, we surveyed contrastive learning concepts and their relations to other fields. We also structured contrastive pretraining as self- vs. supervised learning, highlighted existing challenges and provided pointers to future research directions.
References


[Jia et al., 2021] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021.


[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. \textit{CoRR}, abs/2102.08473, 2021.


