Perturb, Predict & Paraphrase: Semi-Supervised Learning using Noisy Student for Image Captioning

Arjit Jain[1] , **Pranay Reddy Samala**[1] , **Preethi Jyothi**[1] , **Deepak Mittal**[2] and **Maneesh Singh**[2]
[1]Indian Institute of Technology Bombay
[2]Verisk Analytics
{arjit,pranayr,pjyothi}@cse.iitb.ac.in,{deepak.mittal,maneesh.singh}@verisk.com

Abstract

Recent semi-supervised learning (SSL) methods are predominantly focused on multi-class classification tasks. Classification tasks allow for easy mixing of class labels during augmentation which does not trivially extend to structured outputs such as word sequences that appear in tasks like image captioning. Noisy Student Training is a recent SSL paradigm proposed for image classification that is an extension of self-training and teacher-student learning. In this work, we provide an in-depth analysis of the noisy student SSL framework for the task of image captioning and derive state-of-the-art results. The original algorithm relies on computationally expensive data augmentation steps that involve perturbing the raw images and computing features for each perturbed image. We show that, even in the absence of raw image augmentation, the use of simple model and feature perturbations to the input images for the student model are beneficial to SSL training. We also show how a paraphrase generator could be effectively used for label augmentation to improve the quality of pseudo labels and significantly improve performance. Our final results in the limited labeled data setting (1% of the MS-COCO labeled data) outperform previous state-of-the-art approaches by 2.5 on BLEU4 and 11.5 on CIDEr scores.

1 Introduction

Semi-supervised learning (SSL) has been a long-standing problem of interest, with the potential to leverage large volumes of unlabeled data in conjunction with relatively smaller amounts of labeled data. SSL techniques have seen a resurgence in recent years by incorporating data augmentation into the training pipeline, along with loss functions that are robust to the label noise introduced by these augmentations. These techniques have been shown to yield impressive performance improvements even with using small amounts of labeled data [Berthelot *et al.*, 2019; Xie *et al.*, 2020a; Sohn *et al.*, 2020]. While prior work has predominantly focused on image classification, the application of SSL techniques to sequence prediction tasks (e.g. image captioning) has been far less explored. In this work, we present the first detailed investigation of a popular SSL technique for the task of image captioning and propose new enhancements that help significantly outperform existing state-of-the-art approaches on image captioning in limited labeled data settings.

Our proposed techniques are based on the popular SSL paradigm of self-training within a teacher-student framework. This involves a teacher model that is trained on a small amount of labeled data and subsequently used to annotate unlabeled data. A student model is then trained on the combination of labeled data and unlabeled data (with pseudo labels generated by the teacher model). Recent work introduced *Noisy Student Training* [Xie *et al.*, 2020b] that augments this training regime by introducing noise in the student during training and iterating the process. This approach has led to consistent improvements in performance across diverse tasks including image classification [Xie *et al.*, 2020b], object detection [Zoph *et al.*, 2020], machine translation [He *et al.*, 2020] and speech recognition [Park *et al.*, 2020].

In this work, we adapt and improve noisy student training for image captioning. We present two main improvements to the existing noisy student training paradigm:

- We propose "Object Dropout" that directly perturbs object features in an image for noisy student training. This technique performs at par with standard data augmentation techniques while being significantly faster than the latter to implement.
- We propose the use of label augmentation in self-training for sequential outputs using a paraphrase generator that is trained on unpaired caption text data. This module also fixes errors and improves the quality of low-confidence predictions of unlabeled images. This is a very effective augmentation step that helps improve CIDEr scores on the test set of MS-COCO over previous state-of-the-art approaches by 11.5 points.

Code, models, and datasets will be made publicly available at https://github.com/csalt-research/perturb-predict-paraphrase.

2 Related Work

2.1 Semi-Supervised Learning (SSL)

SSL is an established area of machine learning, with numerous survey articles outlining the main techniques and approaches in SSL [Zhu and Goldberg, 2009; Chapelle *et al.*,

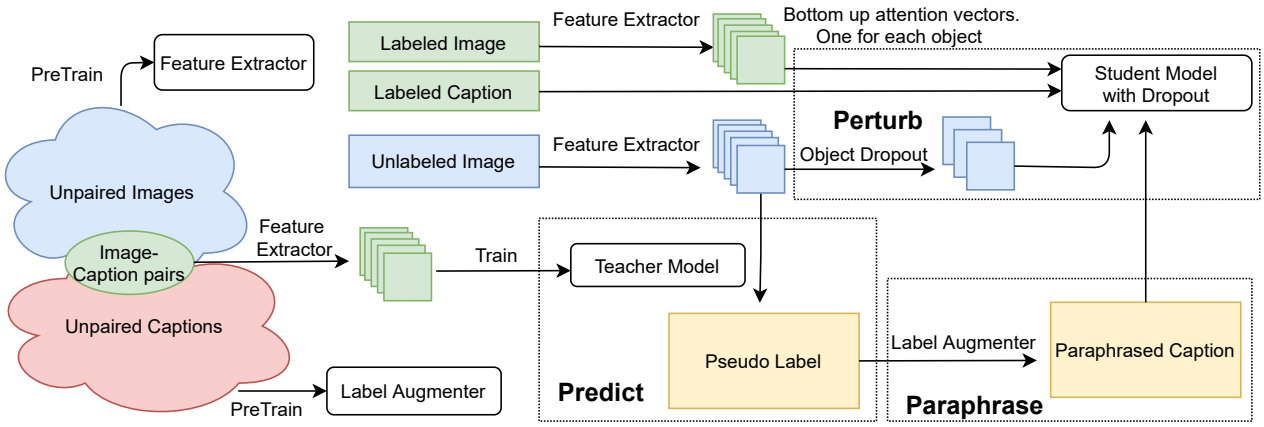

Figure 1: Illustrative overview of our approach. We use a Feature Extractor pretrained on unpaired images and a Label Augmenter finetuned on unpaired captions. We use a Teacher model, trained on paired image-caption data, to generate pseudo labels for unlabeled images. The labeled data and the perturbed unlabeled images with the corresponding paraphrased pseudo labels are used to train the student model.

2010]. Self-training is a very popular SSL paradigm that uses model predictions on unlabeled data as pseudo-labels (after suitable filtering based on confidence thresholds) [hyun Lee, 2013]. Self-training has seen success in various tasks such as domain adaptation [Zou *et al.*, 2018], speech recognition [Kahn *et al.*, 2020; Park *et al.*, 2020] and object detection [Rosenberg *et al.*, 2005; Zoph *et al.*, 2020]. A promising new technique that uses self-training and teacher-student learning is the *Noisy Student Training* framework [Xie *et al.*, 2020b]. Ours is the first work to use this framework for the task of image captioning.

A key idea in SSL, that noisy student also exploits, is that of consistency regularization. Consistency regularization utilizes unlabeled data by assuming that perturbations in inputs and model weights should not affect model predictions. This idea was initially brought to the forefront by [Sajjadi *et al.*, 2016; Tarvainen and Valpola, 2017; Laine and Aila, 2017] and has since been utilized in several recent SSL methods [Sohn *et al.*, 2020; Xie *et al.*, 2020a; Berthelot *et al.*, 2019] to obtain state-of-the-art SSL results. The perturbations involved could be of various forms including stochastic weight transformations like dropout [Srivastava *et al.*, 2014] and/or data augmentations like AutoAugment [Cubuk *et al.*, 2019]. We explore the use of feature augmentations which has not been sufficiently explored.

2.2 Image Captioning

A wide array of approaches have been employed to leverage partially annotated images and unannotated text for image captioning. [Feng *et al.*, 2019; Kim *et al.*, 2019; Gu *et al.*, 2019; Laina *et al.*, 2019] propose GAN-based techniques, [Liu *et al.*, 2018] describes a self-retrieval module and [Guo *et al.*, 2020] proposes visual concept to caption generation. Most of these techniques try to map images and captions to the same latent space by utilizing complex task-specific architectures. Orthogonal to these techniques, our method does not rely on complex task-specific modules, it is model agnostic and hence easy-to-use.

To the best of our knowledge, Self Distillation [Chen *et al.*, 2021] is most closely related to our work; they utilize self-training and share our characteristics of model agnosticism. The key novelties of our method compared to Self Distillation include a) the use of noisy student training, b) our proposal of a paraphrasing model to fix and diversify pseudo labels c) consistency-based training through novel and efficient feature augmentation techniques such as object dropout. Moreover, we significantly outperform Self Distillation on all metrics in the limited labeled data setting.

3 Our Noisy Student Training Approach

Figure 1 illustrates our approach with a schematic diagram and Algorithm 1 outlines our noisy student training algorithm for image captioning. Consider a set of labeled examples $\mathcal{L} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, a set of unlabeled images $\mathcal{U}_I = \{x_{m+1}, x_{m+2}, ..., x_{m+n}\}$, and a set of unpaired captions $\mathcal{U}_C = \{\overline{y}_1, \overline{y}_2, ..., \overline{y}_n\}$ where x refers to an image and y refers to a caption. The first step in noisy student training is to train the teacher model $\mathcal{T}$ to minimize cross-entropy loss on the labeled dataset $\mathcal{L}$; this is denoted by $\ell(y_i, \mathcal{T}(x_i))$ for the i^{th} instance, where $\mathcal{T}(x_i)$ refers to the caption predicted by $\mathcal{T}$ for x_i. This step is followed by N self-training iterations. In each iteration, the model $\mathcal{T}$ is used to predict captions for each of the unlabeled examples in $\mathcal{U}_I$ using a sequential decoder. A student model $\mathcal{S}$ is trained from scratch on the combined datasets $\mathcal{L}$ and $\mathcal{U}_I$ (with its corresponding pseudo labels $\hat{y}_{m+1}, \ldots, \hat{y}_n$) to optimize a joint cross-entropy loss function $L_{\mathcal{S}}$. At the end of each iteration, the newly trained student becomes the teacher for the next iteration.

Introducing stochasticity into student training helps it generalize better than the teacher. The sources of stochasticity can be broadly classified as input noise and model noise. In [Xie *et al.*, 2020b], dropout and stochastic depth are used for model noise and data augmentation via RandAugment [Cubuk *et al.*, 2019] is used for input noise. In [He *et al.*, 2020] that uses noisy student training for machine translation and text summarization, input word tokens are corrupted for input noise and dropout contributes to model noise. Sim-

Algorithm 1 Noisy Student Training for Captioning.

Input: N, $\mathcal{L}$, $\mathcal{U}_I$, $\mathcal{U}_C$, Paraphraser $\mathcal{P}$, Student model $\mathcal{S}$, Teacher model $\mathcal{T}$

1: Train $\mathcal{T}$ on $\mathcal{L}$ to minimize $L_{\mathcal{T}} = \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, \mathcal{T}(x_i))$
2: Fine Tune $\mathcal{P}$ using captions in $\mathcal{U}_C$
3: **for** $t \leftarrow 1$ to N **do**
4: $\quad \{\hat{y}_{m+1}, \ldots, \hat{y}_n\} \leftarrow \mathcal{T}(\mathcal{U}_I)$
5: $\quad$ Add noise to the student model $\mathcal{S}$
6: $\quad$ Train $\mathcal{S}$ to minimize $L_{\mathcal{S}}$

$$= \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, \mathcal{S}(x_i)) + \frac{1}{n}\sum_{i=m+1}^{m+n} \ell(\mathcal{P}(\hat{y}_i), \mathcal{S}(x_i))$$

7: $\quad \mathcal{T} \leftarrow \mathcal{S}$
8: **end for**

ilar to prior work, we also use dropout as a source of model noise. For input noise, we do not use data augmentation in the typical sense where image transformations (e.g. rotate, posterize, etc.) are applied to raw images. Instead, we directly perturb image features via a simple technique that we refer to as object dropout. Additionally, we introduce stochasticity in the label space with the help of a paraphrase module, that is independently trained on caption text.

3.1 Object Dropout

One of the most popular feature representations for image captioning makes use of the bottom-up attention model [Anderson *et al.*, 2018]. This feature extractor identifies image regions, that typically coincides with objects in the image, and produces corresponding feature vectors. The number of feature vectors per image can thus be variable, depending on the number of salient image regions that are identified. Randomly dropping feature vectors per image would have the effect of restricting information only to a subset of objects. More concretely, we define an object dropout probability p_{obj}, such that an object-based feature vector within an image can be entirely dropped (i.e. zeroed out) with probability $1 - p_{\text{obj}}$. We use object dropout as a source of input noise. Unlike standard data augmentation techniques that require transformations of raw images followed by bottom-up feature extraction, object dropout is a feature augmentation technique that is directly applied to the bottom-up features making it much more computationally efficient.

3.2 Label Augmentation via Paraphrasing

Unlike classification tasks where class labels can be easily augmented using simple techniques like linear interpolation [Zhang *et al.*, 2018a], augmenting sequence labels is a non-trivial task. Our solution to this problem is to use a paraphrase module trained on caption data and use it to transform predicted captions for unlabeled images into multiple *paraphrased* forms. For our paraphraser, we start with a powerful pretrained text-to-text model and further finetune it on a paraphrase task involving caption text. To train the paraphraser, one would need access to groups of similar captions which can serve as paraphrases of each other. Identifying these groups and its impact on performance is discussed further in the experiments in Section 5.5.

Apart from providing different ways in which the same caption can be expressed, the paraphraser module also acts as a language model and fixes syntactic issues in the predicted captions which are otherwise non-trivial to fix. We will elaborate on this further, along with providing illustrative examples, in Section 5.5.

4 Experimental Setup

Datasets. We conduct experiments on the MSCOCO dataset [Lin *et al.*, 2014], the standard benchmark used for image captioning. This dataset contains $123k$ images with up to 5 captions per image. We adopt the standard "Karpathy" split used in all prior work, with $113k$ images used in training, and $5k$ images each used for validation and testing. For unlabeled data, we use the "Unlabeled COCO" split from the official MSCOCO Caption challenge. (This allows for consistent usage of unlabeled data across different amounts of labeled data, as in Table 3.) This unlabeled dataset contains $123k$ images with no corresponding captions.

Evaluation Metrics. We use the standard metrics used to evaluate image captioning systems, namely BLEU, METEOR, ROUGE, SPICE, WMD and CIDEr. Consistent with prior work on image captioning, we select the model checkpoint with the best CIDEr score on the validation set to evaluate on the test set. Beam decoding is used for evaluation with the beam width set to 5.

Implementation Details. We use the Attention on Attention Network (AoANet) [Huang *et al.*, 2019] as our base model for the teacher and student. In the fully supervised setting, AoANet was recently shown to achieve state-of-the-art performance on the MSCOCO image captioning dataset, thus making it a good choice for us to adopt as our teacher/student models. A Faster-RCNN model, pretrained on ImageNet and Visual Genome, is used to extract bottom-up feature vectors of images, as described in Section 3.1. Unless specified otherwise, we use beam decoding to generate pseudo labels with a beam width of 2. For the teacher model, we use model dropout with probability $p = 0.3$, no object dropout and label smoothing with probability 0.1. The student model is randomly initialized, and trained from scratch. The student model uses the same AoANet architecture as the teacher model.The labeled batch size is 16, with 5 captions per image, and the unlabeled batch size is 96 with 1 caption per image. The number of noisy student iterations $N = 1$.For the paraphraser, a pre-trained sequence-to-sequence transformer-based model, BART [Lewis *et al.*, 2020] is fine-tuned to paraphrase captions. More details about the data used to train the paraphraser is mentioned in Section 5.5.

5 Experiments and Results

Unless specified otherwise, we mainly focus on the low labeled data setting i.e., using 1% of MSCOCO labeled data. This is similar to the setup adopted in both prior works on SSL for image captioning [Kim *et al.*, 2019; Chen *et al.*, 2021]. We also provide results on varying amounts of labeled data, including 100% of the labeled data in MSCOCO.

Approach	B@1	B@2	B@3	B@4	M	R	CIDEr	S	WMD
Unsupervised Methods									
Pivoting [Gu *et al.*, 2018]	46.2	24.0	11.2	5.4	13.2	–	17.7	–	–
GAN [Feng *et al.*, 2019]	58.9	40.3	27.0	18.6	17.9	43.1	54.9	11.1	–
SME [Laina *et al.*, 2019]	–	–	–	19.3	20.2	45.0	61.8	12.9	–
SGA [Gu *et al.*, 2019]	67.1	47.8	32.3	21.5	20.9	47.2	69.5	15.0	–
Semi-Supervised Methods									
Adversarial Learning [Kim *et al.*, 2019]	63.0	–	–	18.7	20.7	–	55.2	–	–
Deep Mutual Learning [Zhang *et al.*, 2018b]	63.7	44.9	31.1	21.6	19.5	46.2	58.3	12.3	14.1
Mean Teacher [Tarvainen and Valpola, 2017]	62.8	44.2	30.6	21.3	19.5	45.5	59.3	12.2	14.3
Self Distillation [Chen *et al.*, 2021]	67.9	49.8	35.4	25.0	21.7	49.3	73.0	14.5	16.6
Ours (w/o paraphrasing)	**69.2**	**52.1**	**37.7**	26.6	22.5	50.6	76.3	15.3	17.2
Ours (with paraphrasing)	68.8	51.4	37.6	**27.5**	**23.4**	**51.0**	**84.5**	**16.1**	**18.5**

Table 1: Comparisons with state-of-the-art unsupervised and semi-supervised methods using 1% MSCOCO labeled data. Our noisy student training-based models (with and without paraphrasing) outperform prior work on all metrics.

5.1 Comparisons with Prior Work

Table 1 presents a comparison of our best numbers using 1% labeled data with previous state-of-the-art approaches in both semi-supervised and unsupervised image captioning. For a fair comparison, we utilize the base AoANet model without fusion or ensembling, which is similar in performance to the Up-Down architecture utilized in Self Distillation [Chen *et al.*, 2021]. We observe that noisy student training, both with and without paraphrasing, performs significantly better across all metrics. Particularly notable is the large improvement in performance, especially on CIDEr scores, by using the paraphraser for label augmentation.

5.2 Proportions of Labeled and Unlabeled Data

The ratio of labeled to unlabeled data used during training is an important factor that influences the performance of SSL algorithms. We analyze the impact on performance in both directions. We first fix the amount of labeled data and vary the amount of unlabeled data. Next, we fix the amount of unlabeled data and study the impact on performance by varying the amount of labeled data.

Table 2 shows the performance of the student model on using a fixed 1% of labeled data and varying the amount of unlabeled data from 1% to 100%. There is a clear trend of improvement in performance as we increase the amount of unlabeled data. A key observation here is that when the amount of unlabeled data is comparable to the amount of labeled data, the performance of the student is worse than the teacher. However, as the amount of unlabeled data becomes an order of magnitude higher than labeled data, the student begins to outperform the teacher. (Similar observations were made in [Xie *et al.*, 2020b] as well.)

From Table 3, we find that the performance improvements of the student compared to the teacher diminish as the ratio of unlabeled to labeled data decreases. In fact in the 100% labeled data setting, we observe that the student performs worse than the teacher. (In Section 5.6, we show that having more unlabeled data results in the student outperforming the teacher, even in the high labeled data regime.)

Related to the overall proportions of labeled and unlabeled data, ratio between unlabeled and labeled batch sizes is also an important factor to be considered. More so in our setting, where we have multiple captions per image in our labeled dataset, which affects the labeled batch sizes.

5.3 Student Initialization

An important part of the noisy student algorithm is that the student be trained from scratch. Here, we compare training a student model from scratch, i.e. the student parameters are randomly initialized, with training a student model with a

% COCO Unlabeled data	B@4	CIDEr
Teacher	25.2	73.8
Student-1%	23.0	56.9
Student-10%	25.9	74.8
Student-25%	25.7	74.5
Student-50%	25.9	75.2
Student-100%	**26.6**	**76.3**

Table 2: Comparing student models with different amounts of unlabeled data and 1% of labeled data.

% COCO Labeled data	B@4	CIDEr
Teacher-1%	25.2	73.8
Student-1%	**26.6**	**76.3**
Teacher-50%	35.2	111.9
Student-50%	**36.3**	**112.8**
Teacher-100%	**37.0**	**116.5**
Student-100%	36.8	115.6

Table 3: Comparing teacher and student models with a fixed amount of unlabeled data, across different labeled data settings.

Model	B@4	CIDEr
Teacher	25.2	73.8
Student-Scratch	**26.6**	**76.3**
Student-Warm Start	25.9	74.8

Table 4: Comparison on initialization strategy used for the student model. Randomly initialized student outperforms teacher initialized student.

warm start, i.e. the trained teacher parameters are used to initialize the student.

Table 4 shows that both random initialization, and teacher initialization result in student models that outperform the teacher. However, the student trained from scratch outperforms the student with warm start. Similar observations were made for noisy student training used with image classification tasks [Xie *et al.*, 2020b].

5.4 Sources of Stochasticity

We use model dropout probability p to control model noise, and object dropout probability p_{obj} to control input noise.

In Table 5, we report scores on both the test set and validation set to show that both sources of noise appear to be useful. While the best test scores were obtained by setting $p = 0.3$, the best validation scores were obtained by setting $p_{obj} = 0.3$. Here, we only considered object dropout on the unlabeled data.

Table 6 compares image augmentation as a source of stochasticity for student training. With color jittering as the primary augmentation, we notice that inclusion of data augmentation does not provide significant gains, if any, over our approach. We also consider strengthening the baseline augmentation system by using a larger student model with more number of attention heads, including random flips and rotations for augmentation, and using object dropout. Out of these systems, student using color jittering with object dropout performs the best. Finally, we note that our approach is $\mathbf{12}\times$ faster than noisy student training with data augmentation, and achieves comparable performance.

5.5 Paraphrasing

As motivated in Section 3.2, label augmentation via paraphrasing can be extremely effective in low resource settings with sequential outputs. The paraphraser is trained on the set of unpaired captions $\mathcal{U}_C$. To train a paraphraser, we first need

Dropout	Object Dropout	B@4	CIDEr
$p = 0$	$p_{obj} = 0$	25.6(26.0)	74.8(75.8)
$p = 0.3$	$p_{obj} = 0$	**26.6**(**26.6**)	**76.3**(76.1)
$p = 0.5$	$p_{obj} = 0$	25.8(26.1)	74.9(76.0)
$p = 0$	$p_{obj} = 0.3$	26.1(**26.6**)	75.2(**76.3**)
$p = 0.3$	$p_{obj} = 0.3$	26.0(26.2)	75.8(**76.3**)

Table 5: Comparing the impact of model and input noise, controlled via dropout and object dropout respectively, on student training.

Model	B@4	CIDEr
Student with Model + Object Dropout	26.6	76.3
Student with Jitter	26.3	76.2
Student with Jitter (+ Flips/Rotations)	26.3	75.8
Student with Jitter (+ Larger model)	26.2	74.8
Student with Jitter (+ Object Dropout)	26.3	76.0

Table 6: Comparison with Image Augmentation as input noise.

to create groups of similar captions. When available, multiple captions present for each image can be used to create the respective groups. Otherwise, we can perform clustering on the available captions in a meaningful embedding space and use the resulting clusters.

Paraphrasing Fine-tune Data	B@4	CIDEr
No Finetuning	26.7	75.7
Multiple Captions	27.2	**84.2**
Clustering	**27.9**	81.4

Table 7: Comparison of datasets used to finetune the paraphraser.

Table 7 shows the impact of paraphrasers trained using both the above-mentioned grouping techniques ("Multiple Captions" and "Clustering") on student training. The number of available captions per image is 5; hence, we set the number of clusters such that each cluster has at most 5 captions. A pretrained BERT model [Devlin *et al.*, 2019] is used to compute caption embeddings, and k-means clustering [Johnson *et al.*, 2019] is used to compute clusters in this embedding space. It is clear that the use of the paraphraser leads to large improvements in performance compared to not using a paraphraser at all (as in Table 5). Interestingly, we observe that the BLEU-4 scores with clustering captions are higher than annotated caption groups, while CIDEr scores are lower. We hypothesize that the BERT-based clustering paraphraser maintains sentence semantics such as n-grams better than human annotated text and hence yields high BLEU-4 scores. However, since it is not conditioned on image objects and the clusters might tend to bias the paraphraser more towards general words (rather than content-specific words), the BERT-based clustering paraphraser does not do as well as the "Multiple Captions" paraphraser on CIDEr scores that rewards retention of content words. "No Finetuning" refers to using the BART model as-is, without any finetuning on caption text. As expected, finetuning the paraphraser on caption text is critical to derive performance improvements.

Figure 2 shows test predictions using a student model trained with and without paraphrasing. It is clear that the use of the paraphraser in the student model helps produce captions that are superior in quality compared to the student model without the paraphraser. Since the paraphraser scaffolds on BART that has a strong in-built language model, the paraphraser also has the additional benefit of fixing poor-quality caption predictions containing repeated words and ill-

T: A bear bear on a rock next to a bear bear
S: A bear bear on a rock in a rock
SP: A polar bear standing on top of a rock
GT: A polar bear walking on some rocky ground

T: A couple of people standing next to a white cake
S: A couple of people sitting at a table with a cake
SP: A man and a woman sitting at a table with cake
GT: A man and a woman sitting at a table with cake

T: A dog that is standing next to a stuffed bear
S: A man holding a stuffed bear on a wooden fence
SP: A group of sheep standing next to each other
GT: Several sheep with their front hooves on a fence

Figure 2: Qualitative examples from test evaluation of models: T (Teacher model), S (Student w/o paraphrasing), SP (Student with Paraphrasing) in the low labeled data regime, along with GT (Ground Truth) captions.

Low quality pseudo label:	After paraphrasing:
a man wearing a motorcycle on a street street a street	a man on a motorcycle is riding down the street a man riding a motorcycle on a city street a man is riding a motorcycle down the street
a desk with a desk and a desk	a desk with a laptop and a keyboard on it a desk with a laptop on it and a computer monitor a desk with a laptop on it next to a computer monitor

Figure 3: Examples of low-quality pseudo labels with errors (in red) by the teacher and the paraphrased model correcting these errors.

formed sentences. This allows us to make effective use of the unlabeled data and not have to filter out low-confidence pseudo labels. Instead, our approach using the paraphraser attempts to correct the errors in the pseudo labels. Some examples are listed in Figure 3 where the errors in red are fixed by the paraphraser.

We also do an ablation on the benefits of using multiple paraphrased captions for each unlabeled image. We see clear improvements on CIDEr scores going from 1 caption per image (82.4) to 5 paraphrased captions per image (84.2); the BLEU-4 scores stayed roughly the same.

5.6 Noisy Student and High Labeled Data

As described in Section 5.2, our approach works well in the low labeled data regime where the amount of unlabeled data is much larger than the labeled data. However, when the sizes of the unlabeled and labeled data are comparable, we find that the student fails to outperform the teacher. We identify two strategies that mitigate this issue: (1) Decaying the weight of unlabeled data during student training and (2) Using a larger unlabeled dataset. Decaying the weight of the unlabeled data in the training objective smoothly interpolates separate-training [He *et al.*, 2020] and joint-training [Xie *et al.*, 2020b]. For a larger unlabeled dataset, we consider the *Conceptual Captions Dataset* [Sharma *et al.*, 2018] that contains roughly 3.3M image-caption pairs.

Model	B@4	CIDEr
Teacher	37.0	116.5
Student-MSCOCO	36.8	115.6
Student-Conceptual Captions	37.1	**116.7**
Student-MSCOCO with weight decay	**37.6**	**116.7**

Table 8: Comparing different student models using 100% MSCOCO labeled data.

Table 8 compares the effect of the two above-mentioned strategies on student training using 100% of the MSCOCO labeled data. We use only the images from the Conceptual Captions dataset as unlabeled data. Due to computational constraints, we restricted ourselves to using 1M images. We implement the student with weight decay by linearly decaying the contribution of the unlabeled data to the training loss $L_{\mathcal{S}}$ from 1 to 0 in 10 epochs. We observe that both of these strategies successfully extend our noisy student training to be effective even in the high labeled data regime.

6 Conclusion

This is the first work to present a comprehensive analysis of the noisy student framework, a state-of-the-art approach in semi-supervised learning, for image captioning. We show the effectiveness of simple model and feature perturbations to precomputed features and show that they perform at par with data augmentation, while being significantly more computationally efficient. We additionally propose the use of label augmentation in self-training for sequential outputs, wherein instead of filtering out low confidence pseudo labels, we "fix" them using a paraphraser that is trained on unpaired caption text. We show a large performance boost using label augmentation, which is additive to input data augmentation, and significantly improves performance on low-resource image captioning over current state-of-the-art results by over 15%.

Acknowledgements

We thank Yash Shah for his insightful suggestions and help with the experimental setup.

References

[Anderson *et al.*, 2018] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[Berthelot *et al.*, 2019] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

[Chapelle *et al.*, 2010] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.

[Chen *et al.*, 2021] Xianyu Chen, Ming Jiang, and Qi Zhao. Self-distillation for few-shot image captioning. In *WACV*, 2021.

[Cubuk *et al.*, 2019] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.

[Feng *et al.*, 2019] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Unsupervised image captioning. In *CVPR*, 2019.

[Gu *et al.*, 2018] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, and Gang Wang. Unpaired image captioning by language pivoting. In *ECCV*, 2018.

[Gu *et al.*, 2019] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *ICCV*, 2019.

[Guo *et al.*, 2020] Dan Guo, Yang Wang, Peipei Song, and Meng Wang. Recurrent relational memory network for unsupervised image captioning. In *IJCAI*, 2020.

[He *et al.*, 2020] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In *ICLR*, 2020.

[Huang *et al.*, 2019] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, 2019.

[hyun Lee, 2013] Dong hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML*, 2013.

[Johnson *et al.*, 2019] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

[Kahn *et al.*, 2020] Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *ICASSP*, 2020.

[Kim *et al.*, 2019] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Image captioning with very scarce supervised data: Adversarial semi-supervised learning approach. *EMNLP-IJCNLP*, 2019.

[Laina *et al.*, 2019] Iro Laina, Christian Rupprecht, and Nassir Navab. Towards unsupervised image captioning with shared multimodal embeddings. In *ICCV*, 2019.

[Laine and Aila, 2017] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[Lewis *et al.*, 2020] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[Liu *et al.*, 2018] Xihui Liu, Hongsheng Li, Jing Shao, Dapeng Chen, and Xiaogang Wang. Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data. In *ECCV*, 2018.

[Park *et al.*, 2020] Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V. Le. Improved noisy student training for automatic speech recognition. *Interspeech*, 2020.

[Rosenberg *et al.*, 2005] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *WACV*, 2005.

[Sajjadi *et al.*, 2016] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NeurIPS*, 2016.

[Sharma *et al.*, 2018] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[Tarvainen and Valpola, 2017] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

[Xie *et al.*, 2020a] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020.

[Xie *et al.*, 2020b] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

[Zhang *et al.*, 2018a] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2018.

[Zhang *et al.*, 2018b] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018.

[Zhu and Goldberg, 2009] Xiaojin Zhu and Andrew B. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2009.

[Zoph *et al.*, 2020] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. In *NeurIPS*, 2020.

[Zou *et al.*, 2018] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.