Can we train vision and language zero-shot classification models without syntax?

Ajinkya Tejankar 1* Maziar Sanjabi 2 Bichen Wu 2 Madian Khabsa 2
Saining Xie 2 Hamed Pirsiavash 1 Hamed Firooz 2
1 UC Davis 2 Meta AI

Abstract

Natural language supervision in the form of image captions was recently shown to be an effective way of training zero-shot image classification models. In this work, we focus on teasing out what parts of the language supervision are essential for training zero-shot models. Through extensive and careful experiments, we show that replacing intact captions with Bag-of-Words (BoW) does not significantly degrade the zero-shot performance. Surprisingly, we can even slightly improve the performance on some datasets by balancing the frequency of words in BoW.

1 Introduction

Recently, CLIP [1] showed impressive results on zero-shot learning, demonstrating the ability to generalize to unseen image classification tasks. CLIP achieves this by pre-training of vision and text encoders with cross-modal contrastive learning on a large-scale dataset scraped from the internet. Unlike supervised learning datasets that require human annotations, image-text data are easier to obtain [1] and contain richer information about images than categorical labels. As shown in [2], this rich information can provide better supervision and lead to improved generalization as compared to categorical labels. Further, cross-modal contrastive learning connects images with language which makes them useful out-of-the box for zero-shot learning, unlike unimodal contrastive learning.

However, it is not clear whether all the rich information present in textual supervision is necessary for training the model. Alternatively, is it possible to train a good model without all the rich information in textual supervision? Compare the two captions “a man standing in front of a red fire truck” and “man red standing fire”. The first one is a semantically rich signal that denotes multiple objects, their attributes, and different relationships among them. While, the second one is a sparse signal, similar to categorical labels, that simply enumerates few objects, attributes, and relationships. Is the additional information provided in the structure of the first sentence a requirement for CLIP-like models?

To answer the above question, we conduct experiments by deforming the text captions to change their language properties, and observe the impact on the trained model’s zero-shot performance. For example, we use the word shuffling operations to break a sentence to a Bag-of-Words (BoW), we drop random words, etc. We observe that applying such transformations and converting the captions to BoW does not significantly degrade the model’s zero-shot performance. Surprisingly, we can even marginally improve the model’s accuracy on some datasets with a simple strategy of balancing the frequency of words. This further indicates that the model can improve performance without relying on the dense signal of intact captions. Overall, our results show that there is room for designing better vision and language models that can exploit the rich information in intact captions such that zero-shot performance is also improved.

*Work done while interning at Meta AI. Corresponding author (atejankar@ucdavis.edu).

2 Related Works

Our work is mainly related to vision and language pretraining, zero-shot learning, and text as supervision works which are described below. However, our key idea of text deformation is also studied in [3] and [4] for natural language understanding tasks.

Vision and Language Pre-training (VLP): Here, the goal is to learn generalizable, deeply fused, vision and language (V&L) features during pre-training and then fine-tune them for complex downstream V&L tasks [5, 6, 7, 8, 9, 10, 11]. Unlike works in this area, we do not do complex modality fusion, which requires parameter sharing between both modalities, and focus on zero-shot learning for evaluation. Our work is also related to [12] which creates a new dataset and task to understand complex compositional reasoning of VLP models.

Zero-Shot Learning (ZSL): Traditionally, zero-shot learning implies a model that can classify classes without seeing any examples (zero shot) during training [13]. However, CLIP [1] defines it as generalization to novel image recognition datasets where it shows impressive performance by learning representations from scratch with a large-scale (400M) dataset of image and caption pairs. Works building upon CLIP have scaled-up the pre-training dataset to noisier 1B data points [14], improved loss function [15], added unimodal SSL losses [16]. Our training and evaluation is similar to these works [1, 16, 14], but our goal is to shed more light on different aspects of text supervision.

Text as Supervision: Our work explores natural language sentences as a source of supervision. The work in [17] uses the objective of predicting the BoW tags associated with an image to train visual features. This work is extended in [18] to predict n-grams instead of individual words. VirTex [2] uses generative modeling of text as a data-efficient way of training visual features. We show that natural language sentences can be turned into BoW without significantly hurting performance.

3 Method

The primary question explored in this work is: what parts of language supervision are necessary for training vision models for zero-shot (ZS) classification? We first introduce the cross-modal contrastive pre-training framework used in all of our experiments. Then, we introduce various techniques of deforming the text, thereby converting the captions to BoW.

Cross-modal Contrastive Pre-training: Given a pair of image and caption, denoted as \((x_i, y_i)\), we use image encoder \(f\) and text encoder \(g\) to extract a pair of feature embeddings as \((u_i, v_i)\), where \(u_i = f(x_i), v_i = g(y_i)\). The encoders \(f\) and \(g\) are initialized with self-supervised learning (SSL) models trained in respective modalities. Similar to [1] and [19], we use a cross-modal InfoNCE [20] loss. In order to increase the number of negative samples without increasing the batch size and training memory, we follow [21] and maintain a memory bank of negative samples. Embeddings in the memory bank are computed by an exponential moving average (EMA) copy of the models, which we denote as \(f’\) and \(g’\), and their outputs as \(u’_i\) and \(v’_i\). Now we can write our training losses as:

\[
L_{img}^i = -\log \frac{\exp(d(u_i, u’_i)/\tau)}{\sum_{j=1}^{K} \exp(d(u_i, u’_j)/\tau)} \quad L_{txt}^i = -\log \frac{\exp(d(v_i, v’_i)/\tau)}{\sum_{j=1}^{K} \exp(d(v_i, v’_j)/\tau)}
\]

(1)

Where, \(K\) is the size of the memory bank, \(\tau\) is the temperature parameter, and \(d(\cdot, \cdot)\) is the cosine similarity function. The loss per sample is the equally weighted average of above two losses and over all samples in the batch.

3.1 From Intact Captions to Bag-of-Words

In order to understand the importance of natural language structure in captions, we design a set of operations to change/deform them. Each operation removes some information from the original caption. If the lost information was important for zero-shot learning, then the model would degrade. We use following operations in our experiments.

- **Shuffle**: Shuffles the order of words in a given caption, thereby breaking its syntactic structure.
- **RmStopNalpha**: Removes stop words and non-alphabetical words from a given caption. The goal is to remove words that are important for syntactical structure of natural language captions.
The words are first shuffled to remove syntax by shuffle. Stop words ("all", "were", "the", etc.)
and non-alphabetical tokens ("," and ".") are removed with rmpstopnaalpha. Next, words like "cars",
"table", "rocks" are removed by rmtopfreq(1000). Note that the model still sees and learns about
them but from the small set of captions that are kept intact. Further, the blue word "irresistibly" is
removed by limittolowervocab, and purple words like "utter" are removed by keep(n).

Limittolowervocab: Limits the words in a given caption to be from the vocabulary of a base set
of captions. The goal is to restrict the diversity of words in captions. Hence, we randomly select a small
subset of all captions, 10% in our experiments, to be the "base" captions. Then, we deform the 90%
split such that any word not from the vocabulary of the base 10% split is removed.

Rmtopfreq(t): Removes top-t most frequent words. We calculate the frequency of each word as
the number of captions from the base set in which it appears. Similar to limittolowervocab, this
deformation is only applied on the 90% set of captions. The goal is to balance the frequency of words
by suppressing those that occur frequently.

Keep(n): Keeps the first n words of a given caption. The goal is to simply remove information
without discrimination. Note that except keep(n), the order in which other operations are applied
does not matter. Hence, it is always applied in the end after all other operations.

These operations can be combined in different ways to increase or decrease the strength of deforma-
tion. The default deformation for constructing BoW captions is the following cascade of operations:
shuffle + rmpstopnaalpha + limittolowervocab + rmtopfreq(1000) + keep(4). The re-
sulting dataset is referred to as the "default BoW". See Figure 1 for a comparison of intact and BoW
image caption pairs. Note that in some cases, captions may become empty if all their words are
removed during deformation. In this case, the empty caption and the corresponding image is removed
from the dataset which reduces the BoW dataset size. For instance the "default BoW", dataset is
about 20% smaller than the original intact captions dataset.

4 Experiments

We use Conceptual Captions (CC) [22] with ~ 2.9M image-caption pairs for training our models. We
use following encoders: ResNet-50 trained on unlabeled ImageNet [23] with BYOL [24] as the image
encoder and transformer models trained with DeCLUTR (sci-base) [25], BERT (base-uncased) [26],
RoBERTa (base) [27], and DistilBERT (base-uncased) [28] as text encoders. We train for 20 epochs
by default and the results are averaged over 4 runs with different seeds. For zero-shot evaluation, we
follow the procedure in CLIP [1] but only use a single prompt, "a photo of a {class name}".

Intact Captions vs. Bag-of-Words: We progressively add stronger deformations and explore their
effect on zero-shot ImageNet evaluation in Table 1. We find that shuffling the word order, removing
stop/ non-alphabetical words, and only keeping 4 original words for 100% of the captions does not
significantly hurt the model performance. Further, we show that performance can even be improved
by keeping 10% captions intact and removing the top 1000 most frequent words from the rest 90% in
Table 1: **Effect of deformations:** Each subsequent row adds a new operation on top of the previous row with lesser indentation. “mostly BoW” refers to the setting where 90% of text is converted to BoW while in “only BoW” all 100% are BoW. Note that Keep(n) is always applied in the end after all other deformations. Also, note that the application of a deformation may lead to an empty caption in which case the image-caption pair is removed from the dataset. For instance, deformation in the last row results in \(\approx 20\%\) reduction in dataset size. Despite this reduction, most deformations do not significantly hurt the performance. In fact, the last row even shows slight improvement.

<table>
<thead>
<tr>
<th>Deformations</th>
<th>Avg. Cap Len</th>
<th>Zero-Shot on ImageNet DeCLUTR</th>
<th>DistilBERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intact captions</td>
<td>10.3</td>
<td>27.9</td>
<td>29.5</td>
</tr>
<tr>
<td><strong>only BoW</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shuffle</td>
<td>10.3</td>
<td>25.9</td>
<td>26.9</td>
</tr>
<tr>
<td>+ RmStopNalpha</td>
<td>5.8</td>
<td><strong>27.8</strong></td>
<td><strong>28.5</strong></td>
</tr>
<tr>
<td>+ Keep(1)</td>
<td>1.0</td>
<td>16.7</td>
<td>18.3</td>
</tr>
<tr>
<td>+ Keep(2)</td>
<td>2.0</td>
<td>24.2</td>
<td>26.1</td>
</tr>
<tr>
<td>+ Keep(4)</td>
<td>3.8</td>
<td>27.3</td>
<td>27.9</td>
</tr>
<tr>
<td><strong>mostly BoW</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shuffle</td>
<td>10.3</td>
<td>25.7</td>
<td>28.2</td>
</tr>
<tr>
<td>+ RmStopNalpha</td>
<td>6.3</td>
<td>28.2</td>
<td>28.9</td>
</tr>
<tr>
<td>+ Keep(4)</td>
<td>4.5</td>
<td>27.8</td>
<td>29.1</td>
</tr>
<tr>
<td>+ LimitToBaseVocab</td>
<td>4.5</td>
<td>28.0</td>
<td>29.0</td>
</tr>
<tr>
<td>+ RmTopFreq(1000)</td>
<td>3.2</td>
<td><strong>29.1</strong></td>
<td><strong>30.1</strong></td>
</tr>
</tbody>
</table>

Figure 2: **Intact vs. BoW.** We compare intact and default BoW caption based models on zero-shot evaluation across 4 architectures and 8 datasets. BoW achieves comparable or higher accuracy on most settings resulting in a 7.1% relative improvement in accuracy averaged across all settings.

addition to other deformations. Next, we evaluate the model trained with “default BoW” captions on all 8 datasets in Figure 2. These datasets include fine-grained classification tasks like, food classification [29], flowers classification [32], etc. We observe that for some datasets BoW models are consistently better than their intact counterparts, while on others we observe small degradation. Overall, we find that the average relative change in accuracy is +7.1%. This shows that improvements outweigh the degradations when switching to BoW. See Appendix for more details and ablations.

### 5 Conclusion

We investigate what components of the natural language supervision in image-caption pairs are truly exploited by vision and language models for zero-shot evaluation. We design various operators to deform captions and convert them into Bag-of-Words. We find that such deformations do not hurt the model performance, and can even slightly improve it in some cases.
References


[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. 2

[16] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In International Conference on Learning Representations, 2022. 2, 8, 9


[34] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Pattern Recognition Workshop, 2004. 3


BoW vs Intact captions on retrieval evaluation: The task of caption retrieval is a harder task compared to zero-shot classification. As a result, the BoW model results in a significantly worse performance compared to its intact captions counterpart in the zero-shot setting. However, when the models are fine-tuned the gap between them is greatly reduced. These findings are similar to those of [3, 8] which show that input deformations during pre-training do not hurt the models when they are fine-tuned for downstream tasks. For Image to text retrieval with R@1, we see that the gap between Ours (BoW) and Ours (Intact) is about 6 points but when both of them are fine-tuned the gap reduces to about 1 point.

<table>
<thead>
<tr>
<th></th>
<th>Image to Text</th>
<th>Text to Image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@1</td>
<td>R@5</td>
</tr>
<tr>
<td>Zero-Shot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual N-Grams</td>
<td>8.7</td>
<td>23.1</td>
</tr>
<tr>
<td>Ours (BoW)</td>
<td>17.7</td>
<td>39.5</td>
</tr>
<tr>
<td>Ours (Intact)</td>
<td>23.0</td>
<td>48.2</td>
</tr>
<tr>
<td>CLIP</td>
<td>58.4</td>
<td>81.5</td>
</tr>
<tr>
<td>Fine-tune</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours (BoW)</td>
<td>54.4</td>
<td>81.6</td>
</tr>
<tr>
<td>Ours (Intact)</td>
<td>55.8</td>
<td>82.5</td>
</tr>
</tbody>
</table>

A Appendix

Implementation Details: We use PyTorch for implementation with SGD optimizer (weight_decay = 0, momentum = 0.9), batch size = 512, and a cosine learning rate decay with initial LR = 0.003. The model is trained on 8 A100 GPUs which requires 18hrs for 20 epochs of full 2.9M CC dataset. The temperature is set to 0.02, and the default memory bank size is 8192 (~8k). We use a linear layer on top of the vision encoder to match the dimension of its embeddings with the text embeddings. Since the dimension of text encoder embeddings is typically 768, the linear layer is 2048 x 768 since we use ResNet-50 as the vision encoder. For the creating a single sentence level embedding from a list of tokens, we use average pooling. We also perform an ablation over using [CLS] pooling but we did not find any noticeable difference in performance. The comparison between average and [CLS] pooling methods can be found in Table 7.

BoW vs intact captions on retrieval evaluation: Despite being image-to-text retrieval at the core, zero-shot evaluation only tests the ability of a model to distinguish between a set of categories or nouns. This is easier compared to the task of cross-modal (image-to-text and text-to-image) retrieval on a dataset like MS-COCO [36] where the descriptions can be much more complex. Thus, it is reasonable to expect a model trained on BoW text descriptions to perform worse than its intact counterpart in the zero-shot retrieval setting. We indeed see this behaviour in Table 2 but, we also see that when both models are fine-tuned the gap between them shrinks greatly. This finding is consistent with U-VisualBERT [8] where a vision-and-language model without any paired data has comparable performance to the one that uses paired data for fine-tuning evaluation. Further, [3] shows that transformer based language models are robust to word order shuffle during pre-training for fine-tuning evaluation.

Intact captions baselines: Here, we show that our training setup is competitive with other SOTA methods given similar number of aligned images. We compare our model trained for 40 epochs with intact captions and DistilBERT architecture. The comparison in Table 3 shows that our model trained on only ~3M aligned data points is comparable to the CLIP [1] model trained on 15M aligned data points. Moreover, our results are higher compared to those of DeCLIP [16] and CLIP (implementation from [16]) given same amount of aligned data. However, DeCLIP and CLIP models are initialized from scratch while ours are initialized from SSL models. We believe this difference is the reason for the better performance of our models.

Ablations: We explore freezing one of the encoders in Table 4. We find that even a frozen text encoder works better with BoW captions than intact ones despite being pre-trained on intact natural language sentences. We study the effect of removing different number of most frequent words from
Table 3: **Intact captions baselines.** We compare the results of our implementation with other works to show that our implementation is competitive. \(^b\) refers to results of CLIP \([1]\) method from \([16]\).

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>IN1k-ZS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP ([1])</td>
<td>YFCC-15M</td>
<td>31.3</td>
</tr>
<tr>
<td>CLIP(^b)</td>
<td>CC-3M</td>
<td>20.6</td>
</tr>
<tr>
<td>DeCLIP(^a) ([16])</td>
<td>CC-3M</td>
<td>27.2</td>
</tr>
<tr>
<td>Ours</td>
<td>CC-3M</td>
<td>30.3</td>
</tr>
</tbody>
</table>

Table 4: **Ablation of freezing different encoders.** In case of frozen text encoder, intact caption results are similar to BoW. This shows that despite being initialized from an already trained language model, presence or absence of syntax does not matter. Further, we find that freezing the image encoder hurts the model significantly less as compared to freezing the text model. This indicates that training text encoder is more important than the image encoder.

<table>
<thead>
<tr>
<th>Input</th>
<th>IN-1k-ZS</th>
<th>Freeze</th>
<th>Train</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intact captions</td>
<td>15.1</td>
<td>26.7</td>
<td>29.5</td>
</tr>
<tr>
<td>BoW captions</td>
<td>16.9</td>
<td>26.4</td>
<td>30.1</td>
</tr>
</tbody>
</table>

the base vocabulary in Table 5. We also explore the effect of training hyperparameters like epochs, memory bank size, and text encoder architectures/initializations on an intact captions model in Table 6. We can see that memory bank size = 8k and epochs = 20 work well for all architectures, and longer training only helps marginally. Thus, we chose these as the default parameters for all experiments. We chose DistilBert and DeCLUTR for other experiments since they are respectively the best and the worst. In Table 7, we study the effect of [CLS] pooling vs average pooling for caption embeddings. While average pooling is the default in our experiments, we find that switching to [CLS] does not change the message of our work. Even with [CLS] pooling, BoW models are still better than their intact counterparts by similar margins.

**Transfer linear evaluation:** Similar to \([1, 24, 37]\), we also evaluate the transferability of our trained models by training linear layers on top of frozen vision encoders. The training and evaluation procedure is exactly the same as MSF \([38]\), and the results are reported in Table 8.

**Detailed Zero-Shot:** We provide detailed zero-shot with mean and standard deviations in Table 9.

Table 5: **Ablation for removing different number of most frequent words.** We explore different parameters for RmTopFreq\((t)\) operation in the default deformation. We can see that removing too many frequent words reduces the dataset size which degrades the results. The average caption length increases for \(t > 2000\) since the overall dataset size decreases but the 10% captions are always intact.

<table>
<thead>
<tr>
<th></th>
<th>Avg. Cap Len</th>
<th>Δ Dset Size</th>
<th>IN1k-ZS De CLUTR</th>
<th>Distil BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intact captions</td>
<td>10.3</td>
<td>0%</td>
<td>27.7</td>
<td>29.6</td>
</tr>
<tr>
<td>RmTopFreq(0)</td>
<td>10.3</td>
<td>0%</td>
<td>27.8</td>
<td>29.1</td>
</tr>
<tr>
<td>RmTopFreq(500)</td>
<td>3.4</td>
<td>-10.3%</td>
<td>29.3</td>
<td>30.3</td>
</tr>
<tr>
<td>RmTopFreq(1000)</td>
<td>3.2</td>
<td>-19.6%</td>
<td>29.1</td>
<td>30.2</td>
</tr>
<tr>
<td>RmTopFreq(2000)</td>
<td>3.1</td>
<td>-34.1%</td>
<td>28.3</td>
<td>29.1</td>
</tr>
<tr>
<td>RmTopFreq(4000)</td>
<td>3.4</td>
<td>-53.6%</td>
<td>26.8</td>
<td>27.5</td>
</tr>
<tr>
<td>RmTopFreq(8000)</td>
<td>4.7</td>
<td>-73.5%</td>
<td>21.4</td>
<td>23.3</td>
</tr>
</tbody>
</table>
Table 6: Ablation of training hyperparameters. All models are trained with intact captions. We can see that 8k memory bank size and 20 epochs work well for all inits with DistilBERT being the best, and longer training only leads to marginal gains. Thus, we select 8k memory bank size and 20 epochs as the default setting for most experiments.

<table>
<thead>
<tr>
<th>Memory</th>
<th>Epochs</th>
<th>DeCLUTR</th>
<th>DistilBERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>8k</td>
<td>10</td>
<td>25.9</td>
<td>28.0</td>
</tr>
<tr>
<td>8k</td>
<td>20</td>
<td>27.7</td>
<td>29.6</td>
</tr>
<tr>
<td>8k</td>
<td>30</td>
<td>28.2</td>
<td>29.8</td>
</tr>
<tr>
<td>8k</td>
<td>40</td>
<td>28.5</td>
<td>30.3</td>
</tr>
<tr>
<td>16k</td>
<td>10</td>
<td>25.8</td>
<td>27.4</td>
</tr>
<tr>
<td>16k</td>
<td>20</td>
<td>28.5</td>
<td>29.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory</th>
<th>Epochs</th>
<th>RoBERTa</th>
<th>BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>8k</td>
<td>10</td>
<td>25.3</td>
<td>26.5</td>
</tr>
<tr>
<td>8k</td>
<td>20</td>
<td>28.2</td>
<td>28.5</td>
</tr>
<tr>
<td>16k</td>
<td>10</td>
<td>25.2</td>
<td>25.3</td>
</tr>
<tr>
<td>16k</td>
<td>20</td>
<td>27.3</td>
<td>28.0</td>
</tr>
</tbody>
</table>

Table 7: Ablation of average vs. [CLS] pooling: We explore the effect of using [CLS] pooling strategy to go from a list of tokens to a sentence level embedding. Average pooling is the default strategy, but we show that switching to [CLS] does not significantly change the results. The BoW models are still better than their intact counterparts by similar margins.

<table>
<thead>
<tr>
<th></th>
<th>IN-1k-ZS</th>
<th>Avg-pooling</th>
<th>[CLS]</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoBERTa</td>
<td>Intact</td>
<td>28.3</td>
<td>28.3</td>
</tr>
<tr>
<td></td>
<td>BoW</td>
<td>28.7</td>
<td>28.9</td>
</tr>
<tr>
<td></td>
<td>Δ</td>
<td>+0.4</td>
<td>+0.6</td>
</tr>
<tr>
<td>BERT</td>
<td>Intact</td>
<td>28.4</td>
<td>28.1</td>
</tr>
<tr>
<td></td>
<td>BoW</td>
<td>28.2</td>
<td>28.4</td>
</tr>
<tr>
<td></td>
<td>Δ</td>
<td>-0.2</td>
<td>+0.3</td>
</tr>
<tr>
<td>DeCLUTR</td>
<td>Intact</td>
<td>27.9</td>
<td>27.9</td>
</tr>
<tr>
<td></td>
<td>BoW</td>
<td>29.1</td>
<td>28.9</td>
</tr>
<tr>
<td></td>
<td>Δ</td>
<td>+1.2</td>
<td>+1.0</td>
</tr>
<tr>
<td>DistilBERT</td>
<td>Intact</td>
<td>29.5</td>
<td>29.3</td>
</tr>
<tr>
<td></td>
<td>BoW</td>
<td>30.1</td>
<td>29.8</td>
</tr>
<tr>
<td></td>
<td>Δ</td>
<td>+0.6</td>
<td>+0.5</td>
</tr>
</tbody>
</table>
Table 8: **Transfer linear evaluation:** We evaluate the models by training linear probes over frozen vision encoders. We see little difference between using BoW and intact caption models. We also see little effect of the text encoder’s architecture over the performance of the vision encoder. Note that unlike the zero-shot setting, only the vision encoder is used during evaluation. Finally, we see that both BoW and intact caption models are better than the BYOL [24] initialization. This suggests that natural language supervision improves the quality of the visual features.

<table>
<thead>
<tr>
<th></th>
<th>Food101</th>
<th>CIFAR100</th>
<th>Sun397</th>
<th>DTD</th>
<th>Pets</th>
<th>Caltech101</th>
<th>Flowers</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYOL [24]</td>
<td>75.3</td>
<td>78.4</td>
<td>62.2</td>
<td>75.5</td>
<td>90.4</td>
<td>94.2</td>
<td>96.1</td>
<td>81.7</td>
</tr>
<tr>
<td>RoBERTa Intact</td>
<td>77.7 (+0.2)</td>
<td>78.6 (+0.4)</td>
<td>66.4 (+0.1)</td>
<td>76.8 (+0.2)</td>
<td>90.6 (+0.1)</td>
<td>94.9 (+0.1)</td>
<td>96.3 (+0.1)</td>
<td>83.1</td>
</tr>
<tr>
<td>BoW</td>
<td>77.5 (+0.0)</td>
<td>78.2 (+0.1)</td>
<td>65.6 (+0.1)</td>
<td>76.0 (+0.3)</td>
<td>91.0 (+0.3)</td>
<td>94.7 (+0.2)</td>
<td>96.3 (+0.1)</td>
<td>82.8</td>
</tr>
<tr>
<td>BERT Intact</td>
<td>77.7 (+0.3)</td>
<td>79.1 (+0.3)</td>
<td>66.1 (+0.1)</td>
<td>76.7 (+0.2)</td>
<td>90.6 (+0.2)</td>
<td>94.8 (+0.0)</td>
<td>96.7 (+0.1)</td>
<td>83.1</td>
</tr>
<tr>
<td>BoW</td>
<td>77.8 (+0.2)</td>
<td>78.6 (+0.4)</td>
<td>65.5 (+0.0)</td>
<td>76.3 (+0.2)</td>
<td>91.1 (+0.1)</td>
<td>94.8 (+0.1)</td>
<td>96.4 (+0.1)</td>
<td>82.9</td>
</tr>
<tr>
<td>DeCLUTR Intact</td>
<td>77.9 (+0.1)</td>
<td>78.9 (+0.3)</td>
<td>66.3 (+0.1)</td>
<td>77.0 (+0.2)</td>
<td>90.7 (+0.1)</td>
<td>94.8 (+0.1)</td>
<td>96.4 (+0.1)</td>
<td>83.1</td>
</tr>
<tr>
<td>BoW</td>
<td>77.7 (+0.1)</td>
<td>78.5 (+0.2)</td>
<td>65.7 (+0.1)</td>
<td>76.2 (+0.3)</td>
<td>90.9 (+0.3)</td>
<td>94.6 (+0.2)</td>
<td>96.0 (+0.0)</td>
<td>82.8</td>
</tr>
<tr>
<td>DistilBERT Intact</td>
<td>77.8 (+0.2)</td>
<td>78.8 (+0.3)</td>
<td>66.5 (+0.1)</td>
<td>76.3 (+0.3)</td>
<td>90.3 (+0.8)</td>
<td>94.8 (+0.1)</td>
<td>96.3 (+0.2)</td>
<td>83.0</td>
</tr>
<tr>
<td>BoW</td>
<td>77.6 (+0.2)</td>
<td>78.2 (+0.3)</td>
<td>65.7 (+0.1)</td>
<td>75.8 (+0.3)</td>
<td>90.9 (+0.3)</td>
<td>94.7 (+0.2)</td>
<td>96.5 (+0.1)</td>
<td>82.8</td>
</tr>
</tbody>
</table>

Table 9: **Detailed zero-shot results:** We report detailed zero-shot results here. These results were used to generate the Figure 3 of the main text. We can see that the results are consistently improved for all architectures on Pets by minimum ~ 5 points and on DTD by minimum ~ 2 points. On average, BoW is better than using intact captions by 0.6 points for RoBERTa, 0.4 points for BERT, 2.2 points for DeCLUTR, and 1.9 points for DistilBERT. This shows that the model’s improvements with BoW are relatively higher compared to using intact captions.

<table>
<thead>
<tr>
<th></th>
<th>ImageNet</th>
<th>Food101</th>
<th>CIFAR100</th>
<th>Sun397</th>
<th>DTD</th>
<th>Pets</th>
<th>Caltech101</th>
<th>Flowers</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoBERTa Intact</td>
<td>28.3 (+0.4)</td>
<td>17.8 (+0.3)</td>
<td>31.4 (+0.9)</td>
<td>38.6 (+0.2)</td>
<td>23.2 (+1.1)</td>
<td>17.5 (+0.7)</td>
<td>71.7 (+1.7)</td>
<td>12.8 (+0.6)</td>
<td>30.1</td>
</tr>
<tr>
<td>BoW</td>
<td>28.7 (+0.8)</td>
<td>15.1 (+0.7)</td>
<td>32.8 (+2.7)</td>
<td>35.0 (+0.7)</td>
<td>25.6 (+1.0)</td>
<td>22.8 (+0.8)</td>
<td>71.2 (+1.2)</td>
<td>14.2 (+0.9)</td>
<td>30.7</td>
</tr>
<tr>
<td>∆</td>
<td>+0.4</td>
<td>-2.5</td>
<td>+1.7</td>
<td>+3.5</td>
<td>+2.4</td>
<td>+5.3</td>
<td>+0.5</td>
<td>+1.4</td>
<td>+0.6</td>
</tr>
<tr>
<td>BERT Intact</td>
<td>28.4 (+0.2)</td>
<td>18.0 (+0.4)</td>
<td>30.9 (+1.4)</td>
<td>35.1 (+0.5)</td>
<td>26.0 (+0.6)</td>
<td>20.4 (+0.8)</td>
<td>70.5 (+1.9)</td>
<td>16.1 (+0.8)</td>
<td>30.7</td>
</tr>
<tr>
<td>BoW</td>
<td>28.2 (+0.1)</td>
<td>17.2 (+0.8)</td>
<td>30.8 (+2.0)</td>
<td>32.4 (+0.2)</td>
<td>30.5 (+1.1)</td>
<td>26.5 (+1.0)</td>
<td>68.6 (+1.2)</td>
<td>16.6 (+0.7)</td>
<td>31.1</td>
</tr>
<tr>
<td>∆</td>
<td>+0.2</td>
<td>-0.8</td>
<td>-0.0</td>
<td>-2.7</td>
<td>+4.5</td>
<td>+6.0</td>
<td>-3.8</td>
<td>+0.5</td>
<td>+0.4</td>
</tr>
<tr>
<td>DeCLUTR Intact</td>
<td>27.9 (+0.2)</td>
<td>17.6 (+0.6)</td>
<td>28.5 (+1.1)</td>
<td>37.0 (+0.3)</td>
<td>24.3 (+1.2)</td>
<td>19.7 (+0.8)</td>
<td>69.3 (+1.3)</td>
<td>15.4 (+0.8)</td>
<td>29.9</td>
</tr>
<tr>
<td>BoW</td>
<td>29.1 (+0.2)</td>
<td>17.2 (+0.5)</td>
<td>32.3 (+1.7)</td>
<td>33.8 (+0.2)</td>
<td>31.7 (+1.1)</td>
<td>24.9 (+0.4)</td>
<td>68.4 (+1.0)</td>
<td>19.1 (+0.7)</td>
<td>32.1</td>
</tr>
<tr>
<td>∆</td>
<td>+1.2</td>
<td>+0.4</td>
<td>+3.8</td>
<td>-2.1</td>
<td>+7.4</td>
<td>+5.2</td>
<td>-0.9</td>
<td>+3.5</td>
<td>+2.2</td>
</tr>
<tr>
<td>DistilBERT Intact</td>
<td>29.5 (+0.1)</td>
<td>17.2 (+0.4)</td>
<td>30.8 (+1.4)</td>
<td>38.0 (+0.4)</td>
<td>23.6 (+1.3)</td>
<td>20.8 (+1.0)</td>
<td>72.3 (+0.7)</td>
<td>13.5 (+0.6)</td>
<td>30.7</td>
</tr>
<tr>
<td>BoW</td>
<td>30.1 (+0.1)</td>
<td>17.1 (+0.5)</td>
<td>33.7 (+1.7)</td>
<td>36.3 (+0.2)</td>
<td>29.9 (+1.0)</td>
<td>25.9 (+1.0)</td>
<td>72.0 (+0.6)</td>
<td>15.6 (+0.6)</td>
<td>32.6</td>
</tr>
<tr>
<td>∆</td>
<td>+0.6</td>
<td>-0.1</td>
<td>+2.9</td>
<td>-1.8</td>
<td>+6.3</td>
<td>+5.0</td>
<td>-0.2</td>
<td>+2.1</td>
<td>+1.9</td>
</tr>
</tbody>
</table>

A.1 Utilizing Unaligned Images

The insight that intact captions can be effectively replaced with BoW opens the opportunity to use unaligned images (images without associated captions) by pseudo-labeling them with BoW in a process similar to self-training [39]. The goal is to leverage unaligned data which can be easier to acquire and maintain in comparison to aligned data. We call this setting semi-aligned learning since only a subset of data is aligned.

**Method:** We first train a model with fully aligned set $S_a$ of image-caption pairs, and then use it to generate pseudo-labels for unaligned images. Given an unaligned image $x_i$, we augment it $c$ times with random cropping and resizing to get $c$ augmented images $\{x^b_i\}_{b=1}^c$. Next, we get features for each $x^b_i$ by passing it through the vision encoder $u_i^b = f(x_i^b)$. Then, for each $u_i^b$, we retrieve its top-$k$ nearest neighbors from a set of captions, and put them in a set $NN_i = \{y_i^j\}_{j=1}^{k_{c,k}}$. The set of captions used for retrieval are intact and come from the set $S_a$ which was also used for training the fully-aligned model. Next, we use different strategies to rank the words in the $NN_i$ and then pick first $p$ words. The goal of these word ranking strategies is to find the set of words that best describe the image. The entire process is illustrated in Figure 3 and the resulting pseudo-labels for unaligned images are visualized in Figures 4.

**Word Ranking Strategies:**
- **MaxCount:** rank each word according to the count of times it is present at least once in a caption $y_i^j \in NN_i$.
- **RmTopCount:** each word is ranked according to the MaxCount strategy, but similar to the text deformation discussed previously. 1000 most frequent words in the $S_a$ captions are removed. The goal is to focus more on the infrequent words.
- **RmTopRand:** rank
Generating pseudo BoW captions: We illustrate the process of constructing pseudo BoW captions for unaligned images in this figure. A fully aligned dataset, CC [22], to train the vision and text encoders and to obtain the intact captions used for retrieval. We find nearest neighbors in the text domain for different crops of a given unaligned image. The retrieved captions of all crops are then aggregated into a BoW. The words are then ranked according to one of the strategies mentioned in Section 2.3 of main text, and a few top words (4 in this figure) are chosen as the pseudo BoW caption. We used the WeightedCount strategy in this figure.

Table 10: Semi-aligned. We explore using additional images without captions (unaligned). We experiment with various word ranking strategies for creating BoW pseudo caption and find that WeightedCount works the best and is on-par with the CLIP model that is trained with 15M aligned data. A model from the first row is used to for pseudo-labeling. We also compare with a longer trained baseline for a fair comparison.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Unalig-</th>
<th>Epochs</th>
<th>IN1k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aligned-only (ours)</td>
<td>2.3M</td>
<td>20</td>
<td>30.1</td>
</tr>
<tr>
<td>Aligned-only (ours)</td>
<td>2.3M</td>
<td>40</td>
<td>30.7</td>
</tr>
<tr>
<td>CLIP</td>
<td>15M</td>
<td>32</td>
<td>31.3</td>
</tr>
<tr>
<td>RmTopRand</td>
<td>2.9M</td>
<td>10</td>
<td>30.5</td>
</tr>
<tr>
<td>RmTopMaxCount</td>
<td>2.9M</td>
<td>10</td>
<td>31.1</td>
</tr>
<tr>
<td>MaxCount</td>
<td>2.9M</td>
<td>10</td>
<td>31.4</td>
</tr>
<tr>
<td>WeightedCount</td>
<td>2.9M</td>
<td>10</td>
<td>31.5</td>
</tr>
</tbody>
</table>

Experiments: In addition to the fully aligned CC dataset [22] of size ~ 3M, we use 3M randomly sampled images from ImageNet-21k [23] during this experiment. We use the DistilBERT model with 30.2% on the IN-1k-ZS benchmark for pseudo-labeling. We selected this model randomly from one of the 4 models trained with above configuration. It is a BoW model trained with default hyperparameters. The pseudo-labels are only calculated once at the beginning of the training after which they remain frozen. The results are reported in the Table 10. The total number of iterations used by a semi-aligned model are larger since it relies on an already trained, aligned-only model. Thus, we compare semi-aligned models with a longer trained baseline.
Figure 4: Examples of pseudo BoW captions: We show some examples of the pseudo BoW captions generated by the WeightedCount strategy. Each row contains images from a single category. The actual category name is listed on the left, at the start of each row, and the pseudo captions are below each image.