Curriculum Learning for Data-Efficient Vision-Language Alignment

Tejas Srinivasan Xiang Ren Jesse Thomason
University of Southern California
tejas.srinivasan@usc.edu

Abstract

Aligning image and text encoders from scratch using contrastive learning requires large amounts of paired image-text data. We alleviate this need by aligning individually pre-trained language and vision representation models using a much smaller amount of paired data with a curriculum learning algorithm to learn fine-grained vision-language alignments. TOnICS *(**T**raining with **On**tology-**I**nformed **C**ontrastive **S**ampling) initially samples minibatches whose image-text pairs contain a wide variety of objects to learn object-level vision-language alignment, and progressively samples minibatches where all image-text pairs contain the same object to learn finer-grained contextual alignment. Aligning pre-trained BERT and VinVL-OD models to each other using* TOnICS *outperforms CLIP on downstream zero-shot image retrieval using $< 1\%$ as much training data.*

1. Introduction

Aligned representations for language and vision—which encode texts and images in a common latent space—are necessary to perform effective cross-modal retrieval. CLIP [7] and ALIGN [4] train individual text and image encoders from scratch to produce aligned image-text representations. They demonstrate accurate zero-shot retrieval due to strong cross-modal alignment. However, these models were trained on proprietary datasets of 400M and 1B, respectively, image-text pairs on hundreds of GPUs and TPUs, which is infeasible for non-industry practitioners.

CLIP and ALIGN align their encoders using the contrastive InfoNCE objective [5], which seeks to maximize the mutual information between image and text representations. In the InfoNCE objective, the model must correctly identify the positive image-text pair from among a set of negatives formed by the other minibatch pairs.

Since samples within a minibatch act as negative samples for each other in the InfoNCE objective, the minibatch determines the granularity of alignment that is learned. Minibatches constructed by random sampling contain a large variety of objects in the images and texts. To correctly match

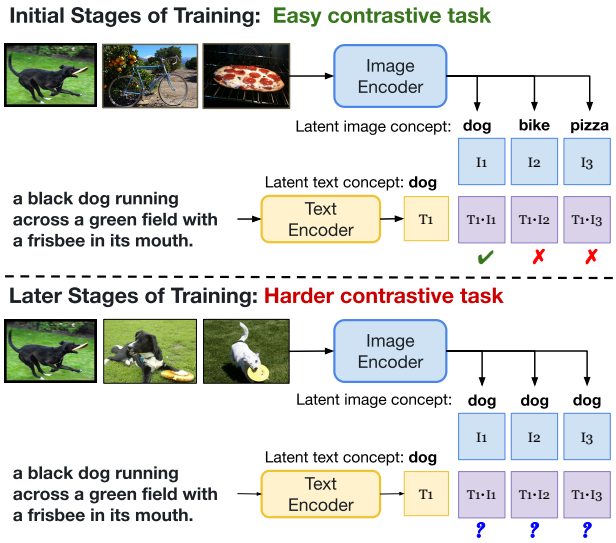

Figure 1. TOnICS is a contrastive, curriculum learning algorithm for aligning language and vision encoders.

a *dog*-related caption to its image, it is sufficient to identify that the retrieved image must contain a dog, since most randomly sampled negative images will not contain a dog. Random minibatch sampling reduces the contrastive task to just object-matching.

When minibatches are sampled such that the images contain the same objects, object-level alignments no longer suffice (Figure 1). The contrastive task can no longer be solved by identifying that the retrieved image must contain a dog, since all the negative images will also have a dog. The model must produce language and vision representations that encode shared *context*-level information, resulting in a finer-grained alignment.

In this work, we leverage rich single-modality pre-trained models—BERT [2] for language, VinVL-OD [11][1] for vision—and align them using the InfoNCE contrastive objective. We propose TOnICS, a curriculum learning algorithm which initiates training by sampling minibatches randomly

[1]We use VinVL-OD to refer to the pre-trained VinVL object detector, not the pre-trained vision-language model.

and progressively makes the contrastive task harder by constructing minibatches containing the same object class in the image and text inputs. We show that our learned representations have strong cross-modal alignment—outperforming CLIP on zero-shot Flickr30K image retrieval—while using less than 1% as much paired image-text training data.

2. Contrastive Vision-Language Alignment

We align language representations from BERT [2] and visual representations from a VinVL object detector [11]. Our BERT-VinVL Aligner model is similar to the phrase grounding model from [3].

During training, the input to the model is a minibatch of N_B triplets, where each triplet $X_i = \{t^i, v^i, w\}$ represents an image-text pair. Image caption t^i is encoded using BERT and contains a noun w with word representation h^i. A set of region features v^i are extracted from VinVL-OD, a frozen pre-trained object detector.[2] We add a learnable linear projection atop these region features.

In the cross-modal interaction, we employ a single Transformer [9] layer that uses i-th noun representation h^i as the query and j-th image features v^j as the keys and values (Figure 2). This layer outputs a visual representation $v_{att}(i,j)$, which is an attended representation of the j-th image, conditioned on the noun from the i-th caption. We compute an image-text score $s(i,j) = \phi(h^i, v_{att}(i,j))$ as the dot product between the i-th noun representation h^i and the attended representation of j-th image $v_{att}(i,j)$.

To align the noun representation h^i to its image v^i, we use the InfoNCE loss [5] to maximize the lower bound of the mutual information between h^i and $v_{att}(i,i)$. InfoNCE minimizes the cross-entropy of correctly retrieving an image v^i from the set of all minibatch images given the query noun representation h^i. We refer to the objective in this setup as the image retrieval loss, $\mathcal{L}_{IR}$:

$$\mathcal{L}_{IR}(i) = -\log \frac{\exp(s(i,i))}{\sum_{j=1}^{N_B} \exp(s(i,j))}$$

The training loss $\mathcal{L}_{IR}$ is the mean loss $\mathcal{L}_{IR}(i)$ over all minibatch instances $i = \{1...N_B\}$. We define a text retrieval loss, $\mathcal{L}_{TR}$, where the image v^i is used to retrieve the correct noun representation h^i:

$$\mathcal{L}_{TR}(i) = -\log \frac{\exp(s(i,i))}{\sum_{j=1}^{N_B} \exp(s(j,i))}$$

We experiment with training our model using just the image retrieval loss $\mathcal{L}_{IR}$, as well as the sum of the two losses $\mathcal{L}_{IR} + \mathcal{L}_{TR}$.

[2]Region features provided at https://github.com/pzzhang/VinVL/blob/main/DOWNLOAD.md

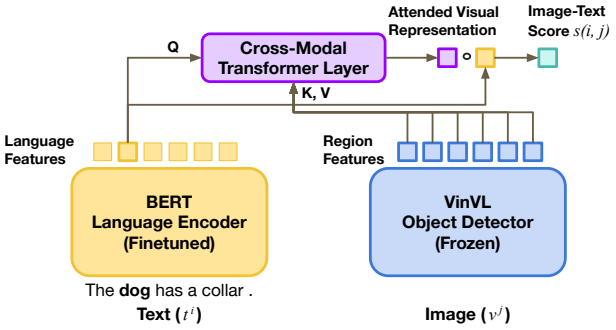

Figure 2. Our BERT-VinVL Aligner model scores every image-text combination (t^i, v^j) in the minibatch.

3. TOnICS: Training with Ontology Informed Contrastive Sampling

Negative samples for the contrastive learning objective come from other pairs in the minibatch. Therefore, *the minibatch sampling itself influences the alignment learned by the model*. We hypothesize that sampling minibatches randomly gives object-level alignments, while sampling *harder* minibatches containing the same object in the image yields finer-grained contextual alignments.

We introduce TOnICS, **T**raining with **On**tology-**I**nformed **C**ontrastive **S**ampling (Figure 3), a curriculum learning algorithm that samples minibatches according to nodes sampled from an ontology. TOnICS first performs object-level alignment via random minibatches, and later learns fine-grained alignments through harder minibatches.

Ontology Induction We begin by heuristically mapping object classes from the VinVL detector to nouns in the training captions, using point-wise mutual information estimates. This yields a set of object classes Θ, where every object class $o \in \Theta$ has a corresponding set of nouns $w(o)$. For instance, the object class *dog*'s noun set $w(o) = \{$*dog, dogs, puppy*$\}$.

Our ontology (Figure 3, left) contains an *entity* node η_e at the root, and an *object node* η_o for every object class $o \in \Theta$ as a child node. Every object node η_o has a corresponding subset of the training data $X(\eta_o)$, whose instances all contain the same object o in the image, and a noun from the corresponding noun set $w(o)$ in the caption.

TOnICS Minibatch Sampling At every training step, TOnICS samples a node η from the ontology according to a sampling probability distribution $P_S(\eta)$. If the entity node η_e is sampled, N_B instances from the full training data are sampled at random for the minibatch. If an object node η_o is sampled, the N_B instances are sampled from the corresponding set $X(\eta_o)$, ensuring the minibatch comprises images depicting object o.

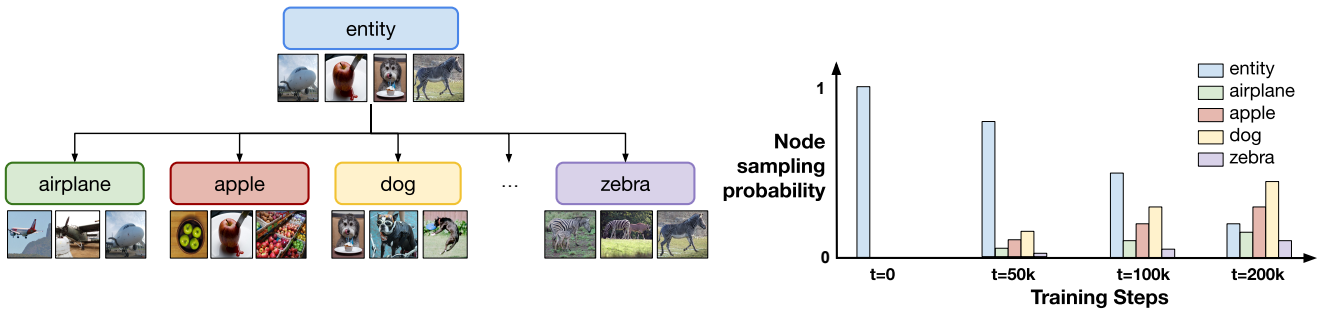

Figure 3. TOnICS selects image-text pairs for the minibatch by first sampling a node η from an ontology, according to a distribution $P_S(\eta)$. Sampling the root *entity* node yields easy minibatches containing pairs with a variety of objects, whereas sampling one of its children *object nodes* yields harder minibatches containing pairs sharing a common object, such as *apple* or *dog*, in a variety of contexts (left). TOnICS performs curriculum learning by moving node sampling mass away from the entity node to the object nodes as training progresses (right).

TOnICS Curriculum Refresh The curriculum is formed by varying the nodes' sampling probability distribution throughout training. We initialize training by setting $P_S(\eta_e) = 1$ and $P_S(\eta_o) = 0$ for all object nodes. After every fixed number of training steps, we evaluate the model's image retrieval performance on a set of held-out instances. If the held-out retrieval accuracy is greater than a certain threshold, we start introducing harder minibatches in the training by *refreshing* the curriculum. The refresh step is performed by multiplying the entity node's current sampling probability $P_S(\eta_e)$ by a factor $\alpha; \alpha < 1$. The remaining probability mass $(1-\alpha) \times P_S(\eta_e)$ is distributed among the object nodes. For each object node η_o, we update its sampling probability:

$$P_S(\eta_o) = P_S(\eta_o) + (1-\alpha)P_S(\eta_e) \times \frac{|X(\eta_o)|}{\sum |X(\eta_o)|}.$$

Object classes that are more common in the training data have more sampling probability mass distributed to their object node η_o, by weighting mass according to the size of the node's instance set, $|X(\eta_o)|$. With each curriculum refresh, sampling mass is pushed down from the entity node to the object nodes, as long as $P_S(\eta_e)$ does not fall below a fixed threshold β. Thresholding $P_S(\eta_e)$ ensures the model still sees random minibatches and does not forget the initially learned object-level alignments.

4. Experiment Details

We train our BERT-VinVL model on MS-COCO and Conceptual Captions. We compare our model against CLIP on downstream retrieval tasks.

4.1. Training Data and Ontology

We train our model on image-text pairs from a combination of MS-COCO [1] and Conceptual Captions [8]. Our triplet instances only contain nouns which we wish to explicitly align with the visual modality. We select a set of 406 nouns, each noun corresponding to one of the 244 object categories Θ (more details in Appendix A). Our final training data consists of 5.8M triplet instances corresponding to 2.84M image-text pairs from 2.4M unique images. The ontology for TOnICS is constructed by creating an object node for each of the 244 object categories, which are children of the root *entity* node.

4.2. Implementation Details

We use pre-trained BERT-base as a text encoder and frozen VinVL-OD, a pre-trained object detector that returns pooled CNN features for all regions-of-interest (ROIs), as an image encoder. We use pre-extracted ROI features, as we cannot backpropagate through the object detector.

All our models are trained for 500K iterations with a batch size of $N_B = 256$, yielding 255 negative pairs for every positive pair. We select the model checkpoint which has maximum Recall@1 on the Flickr30K validation set, evaluated after every 5K iterations. After every 5K iterations, we also evaluate retrieval over a set of 100 held-out instances and perform a curriculum refresh step if the held-out accuracy is at least 90%. When performing a refresh step, we retain $\alpha = 90\%$ of *entity*'s sampling probability, so long as the probability does not fall below $\beta = 0.2$.

Each model was trained on a single V100 GPU for 6 days, compared to CLIP which used 256 V100 GPUs for 12 days.

4.3. Baselines and Evaluation

We compare our aligned model against CLIP [7]. CLIP trains image and text encoders from scratch, using significantly more paired image-text data—400M pairs, compared to our 2.84M pairs. Our model uses the base variant of BERT, so we compare against CLIP-ViT-B/32.[3]

To evaluate the utility of our TOnICS algorithm, we also train our BERT-VinVL Aligner using a **Random** minibatch

[3]Our model trains just 116M parameters during alignment, compared to 151M trained parameters for CLIP-ViT-B32.

Model	# Image-Text Pairs	Minibatch Sampling Method	$\mathcal{L}_{TR}$	Zero-Shot Flickr30K				MS-COCO			
				Image Retrieval		Text Retrieval		Image Retrieval		Text Retrieval	
				R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP-ViT-B/32	400M	Random	-	58.66	83.38	**79.2**	**95**	30.45	56.02	50.12	75.02
BERT-VinVL Aligner	2.84M	Random	✗	58.18	84.24	22.2	47.9	42.67	74.43	15.5	37.7
	2.84M	TOnICS	✗	**60.04**	84.72	18.8	43.1	**47.68**	**77.14**	11.48	27.3
	2.84M	Random	✓	58.9	84.6	76.1	93.3	42.74	74.37	59.84	86.46
	2.84M	TOnICS	✓	59.68	**84.84**	77.4	94	47.15	76.85	**63.7**	**88.5**

Table 1. Results of our BERT-VinVL Aligner model, trained using either Random or TOnICS minibatch sampling, on image and text retrieval compared to CLIP. Numbers in bold represent the best results among all models.

sampling baseline, where the minibatch instances are always randomly sampled throughout the training process.

We directly evaluate our Aligner models and pre-trained CLIP on image and text retrieval, using the Recall@1 and Recall@5 metrics. Specifically, we evaluate zero-shot retrieval on the Flickr30K [6] test set, which contains 1,000 images. We also perform retrieval evaluation on the MS-COCO test set, which contains 5,000 images. This evaluation is not zero-shot since we train on MS-COCO training images.

5. Results and Discussion

We directly transfer both our trained BERT-VinVL Aligner model and pre-trained CLIP to the downstream task of image and text retrieval (Table 1) using the same task formulation from training time.

The Flickr30K evaluation is zero-shot for both CLIP and our BERT-VinVL Aligner model since neither model's training data contains images from the Flickr30K train set. We see that even with the Random minibatch sampling and only the image retrieval loss, $\mathcal{L}_{IR}$, our BERT-VinVL Aligner achieves approximately the same image retrieval performance as CLIP. When the Aligner is trained with our TOnICS curriculum learning algorithm, we get a 1.5% improvement on R@1 over CLIP.

However, when trained without the text retrieval loss $\mathcal{L}_{TR}$, both Aligner-Random and Aligner-TOnICS fail to do well at the text retrieval task. Adding the $\mathcal{L}_{TR}$ loss to Aligner training leads to substantial improvements in downstream text retrieval, with Aligner-Random performing only 3% worse than CLIP. We further see that training with TOnICS leads to a 1% improvement in Flickr30K text retrieval. Adding the text retrieval loss to Aligner-TOnICS slightly hurts image retrieval performance, but still does better than CLIP by 1%.

Since MS-COCO images are included in the Aligner training data, it significantly outperforms CLIP on the MS-COCO retrieval evaluation. Hence, we compare TOnICS to Random sampling on MS-COCO retrieval. We see that TOnICS leads to significant improvements in image retrieval ($\approx$ 5% over Random). We again observe that text retrieval performance is poor without the text retrieval loss during training, but improves significantly with it. TOnICS training results in a 4% improvement over Random in text retrieval.

Minibatch sampling with TOnICS results in large gains in in-distribution retrieval evaluation (MS-COCO) as well as small improvements in zero-shot retrieval (Flickr30K). Training BERT-VinVL Aligner with TOnICS yields better zero-shot image retrieval performance than CLIP, even with substantially less training data.

6. Conclusions

We align individually pre-trained language and vision encoders—BERT and VinVL-OD—using the proposed curriculum learning algorithm, TOnICS. Our aligned model is able to achieve better downstream zero-shot image retrieval performance than CLIP, despite being trained with less than 1% as many image-text training pairs. We also show that our TOnICS algorithm leads to gains in both in-domain and zero-shot retrieval tasks.

Limitations

We use language models pretrained primarily on English text, eliding the challenges of multi-lingual language-vision alignment. Further, our method relies on contrastive learning, which requires a large number of minibatch samples for training. As a consequence, we restrict the object classes in our ontology to only frequently occurring ones, meaning objects in the long tail of the distribution that do not have sufficient training instances are not aligned using our TOnICS algorithm. Finally, our induced ontology is domain-specific, and may need to be re-generated for a new domain.

References

[1] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO Captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 3

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 1, 2

[3] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. Contrastive learning for weakly supervised phrase grounding. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[4] 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 *International Conference on Machine Learning (ICML)*, 2021. 1

[5] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 2

[6] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *International Conference on Computer Vision (ICCV)*, 2015. 4

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 1, 3

[8] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Association for Computational Linguistics (ACL)*, 2018. 3

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*, 2017. 2

[10] Tian Yun, Chen Sun, and Ellie Pavlick. Does vision-and-language pretraining improve lexical grounding? In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021. 6

[11] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting visual representations in vision-language models. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

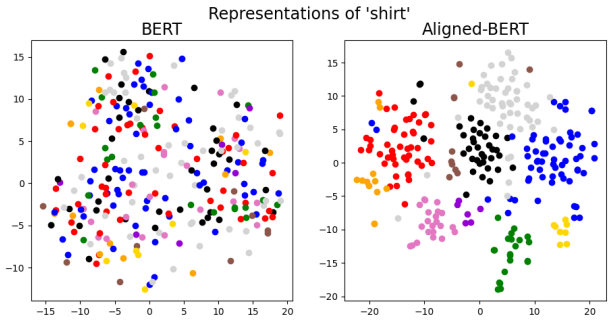

Figure 4. TSNE projections of contextual representations of the word *shirt* occuring in different color contexts. Each dot corresponds to a contextual representations of the word *shirt*, where the color of the dot corresponds to the color of the shirt described in the caption (grey dots represent representations of white shirts). We compare the TSNE visualizations of pre-trained BERT and the Aligned-BERT from our Aligner model.

A. Ontology Induction Details

The nodes in our ontology correspond to object classes that we wish to explicitly align with the visual modality. We decide this set of object classes via the following procedure.

Each noun in the training data captions is lemmatized using NLTK and mapped to the object class with maximum noun-object PMI, calculated over training pairs with object detections. These mappings are then adjusted by hand to correct erroneous mappings—that adjustment process took less than two hours of author time and resulted in a mapping of 827 nouns to 653 distinct object classes. Object classes containing fewer than 5000 instances in the training dataset are filtered out. This results in a set of 406 nouns, each noun corresponding to one of the 244 object classes Θ. For every image-text pair in the original training dataset, we create one triplet for each noun in the text that belongs in our set of 406 nouns. Finally, we form our ontology for `TOnICS` by creating one node corresponding to each of the 244 object classes Θ.

B. Analysis of Aligned Language Representations

We hypothesize that by aligning pre-trained BERT to visual representations from a pre-trained VinVL model, our aligned BERT's representations of visually-groundable objects will contain more visual context information. Similar to [10], we investigate whether noun representations extracted from our Aligned-BERT contain information about their visual attributes that are also described in the caption. Specifically, we look at representations of the word *shirt* in Flickr30K captions where the color of the shirt is also mentioned. We select 275 such captions where the shirt is described as being one of ten colors, and extract the word *shirt*'s contextual representations from both pre-trained BERT and our BERT-VinVL Aligner's text encoder, which we refer to as Aligned-BERT.

Model	Homogeneity	Completeness	V-Score
BERT	9.79 ± 1.48	9.13 ± 1.39	9.45 ± 1.43
A-BERT	42.64 ± 5.51	40.59 ± 5.24	41.58 ± 5.37
CLIP	98.39 ± 0.00	98.28 ± 0.00	98.33 ± 0.00

Table 2. K-Means Clustering metrics (K=10) for *shirt* representations across five different K-Means initializations. We present mean and standard deviation of all metrics. A-BERT is our Aligned BERT.

Figure 4 compares the TSNE visualizations of representations extracted from BERT and Aligned-BERT. We see clear clusters formed by representations of the same colored shirt in Aligned-BERT's visualization, whereas no such clusters exist in the BERT representations.

We also provide a quantitative analysis of the clustering in the representations, by performing K-Means clustering with $K = 10$. We evaluate the Homogeneity and Completeness of these clusters, which are equivalent to Set-Precision and Set-Recall respectively, as well as V-Score which is their harmonic mean. In Table 2, we see that Aligned-BERT's clusters are much more homogenous and complete than pre-trained BERT, but pre-trained CLIP's clusters are much better than both.