Large-scale Cross-Modal Product Retrieval
Team Zebra AI[*]

Andrea Bordone Molini[†] Samadhi W. Arachchilage[†] Aditya Singh[†] Chloe Kim
andrea.bordone1@zebra.com

Abstract

This is a report corresponding to our submission for the ***CVPR 2022 AliProducts Challenge: Large-scale Cross-Modal Product Retrieval****. The aim of this challenge is to design a system which is capable of retrieving related images using text queries. The challenge lies in the scale of data, noisy image-caption pairs, and multi-lingual captions. Our solution is designed around joint image-text embedding training followed by inference time techniques such as test-time augmentation (TTA), image-text score normalization, and ensemble. This allowed us to obtain a mean Recall@5 and Recall@10 at 0.72695 on the test set.*

1. Introduction

Aliproducts2 challenge was proposed to study cross-modal representation learning to connect visual representation of an image to the high-level semantic concepts expressed in a text caption. The large-scale dataset proposed by the challenge moderators is composed of ∼4M image-caption pairs of ∼100K fine-grained classes. It includes noisy image-caption pairs, a problem to take into account during the learning phase of image-text alignments.

2. Solution

2.1. Joint embedding training

We use the popular paradigm of joint embedding learning by training an image encoder and text encoder with the provided data. There are many existing methods suitable for this task and we utilize the following 2 in our solution.

1. CLIP [11]: It trains the image, text and projection networks jointly on the image and text data. The objective function minimizes the cross-entropy between image and text encodings. We add additional loss components to the existing clip objective to explicitly pull apart each image and each text caption in a batch with respect to all images and text captions respectively. This is similar to the image and text self-supervision methods adopted in SLIP [10] and DeCLIP [6] works, without using the traditional contrastive scheme to form the positive samples.
2. ALBEF [5]: It is a 2 stage joint embedding training routine aimed at first aligning the text-image embeddings and then fine-tuning specifically for the task of retrieval. Apart from using the loss objective of CLIP it utilizes additional objectives for image-text matching and contrastive learning. We follow the official implementation of ALBEF [5] for pre-training and fine-tuning.

2.2. Data cleaning

The training data is noisy and contains many instances where the caption does not match the corresponding image. In order to remove such noisy samples from training we split the training data into 4 mutually exclusive subsets and train a CLIP model with a swin backbone corresponding to each. For a model m_i, we aggregate its response for ground-truth pairs on the remaining training set images, $X^{train} \setminus X_i^{train}$. We generate a final score as an average of the individual scores. Heuristically, we observed that similarity scores of ground-truth image-text embeddings below 0.2 corresponded to noisy samples in the training set. We compose a clean training subset, X^{clean}, with samples corresponding to similarity scores ≥ 0.3. We use this clean subset for fine-tuning models at a later stage.

2.3. Text-image consistency score normalization

The similarity score is computed as a dot product between the normalized text features and image features. Each row of the resulting similarity matrix ($S \in R^{n \times m}$) encapsulates the similarity of text to all the images in the database. The idea of consistency score normalization is to ensure that if image i is the highest scoring sample for text j, it should hold true the other way around as well. It is possible that the closest matching text to image pair might not be the same while doing image to text retrieval. To achieve this, we scale

[*] CTO-SI, Zebra Technologies, London, UK.
[†] equal contribution

Image Enc.	Text Enc.	Method	Image Pre-Tr	Text Pre-Tr	Tokenizer	TTA	R@10
Vit-B [4]	ALBEF	ALBEF	deit-base-in1k	ALBEF	BertTokenizer	✗	0.6375
swin-B [7]	ALBEF	ALBEF	in22k	ALBEF	BertTokenizer	✗	0.6632
Swin-B	BERT [3]	CLIP-m	in22k	BERT	BertTokenizer	✓	**0.7019**
Convnext-B [8]	BERT	CLIP-m	in22k	BERT	BertTokenizer	✓	0.6910
Swin-B	DistilBERT [12]	CLIP-m	in22k	BERT	DistilBertTokenizer	✓	0.6842
Convnext-B	DistilBERT	CLIP-m	in22k	BERT	DistilBertTokenizer	✓	0.6559

Table 1. We refer to our modified version of CLIP as CLIP-m as detailed in section 2.

Algorithm 1 Iterative consistency score normalization

```
# $S$: Similarity matrix ($n \times m$)
# $N$: Iterations for the normalization
for i in range(N):
    t2i_max = S.max(dim=1)[0].view(-1, 1)
    i2t_max = S.max(dim=0)[0].view(1, -1)

    t_sim = S / t2i_max
    i_sim = S / i2t_max
    S = (t_sim+i_sim)/2

return S
```

the score with the maximum w.r.t images and the maximum w.r.t the texts. We found that this normalization can boost performance upto $2-3\%$. We repeat the normalization process for few iterations and observed that it provides a further boost of $0.3-0.5\%$. A simple python based implementation is provided in Algorithm 1.

2.4. Test-time augmentation

We make use of test-time augmentations for both the image and the text to help boost the performance at test time. For images, we obtain the embedding features over 7 inferences using random crops (scale=[0.8, 1.0]). For texts, we employ EDA over 5 inferences by randomly selecting only one of the augmentations listed in section 3.

2.5. Ensemble

To combine the similarity scores of different approaches we use a weighted ensembling scheme. The weight is produced as a softmax over their corresponding validation recall@10 using a temperature of 0.1.

2.6. Results

In Table 1, we report the recall@10 on the validation set, prior to normalization, separately for all the models utilized in the final ensemble.

3. Augmentations

The original CLIP [11] work utilizes solely a random square crop (224) from resized images as data augmentation during training. Subsequent works such as Declip [6] and FILIP [14] perform data augmentation on both images and texts such as AutoAugment [2], SimCLR [1] augmentation and EDA [13], back-translation respectively.

We adopt a slightly stronger image augmentation policy with respect to original CLIP one, composed of a RandomResizedCrop with scale in the range [0.6,1.0] and a RandomErasing with Torchvision default parameters. EDA [13] is used as text augmentation strategy, which contains three types of text augmentation strategies: synonym replacement, random swap, and random deletion.

3.1. Other training configs

We strictly followed ALBEF implementation and hyperparameters for generating their corresponding models. For Clip-m, we found that the following set of hyperparameters worked the best:

- First training with 30 epochs and then fine-tuning on X_{clean} for another 10.
- We use a mini-batch size of 3440 text-image pairs in 8 GPUs (Nvidia A100 80GB) and of 3200 for Convnext-B and Swin-B models respectively, both coupled with BERT as a text encoder. The Convnext-B and Swin-B models coupled with distilBert are trained on 8 Nvidia 40GB GPUs with a mini-batch size of 1560 and 1440 pairs respectively.
- We employ AdamW [9] optimizer with weight decay 0.1 and a learning rate of 0.0003 which is linearly ramped up during the first 3 epochs. After warmup, we use the cosine learning rate decay with a final value of 0.0. When using distilBERT [12] as text encoder, the learning rate is increased to 0.0005.

4. What Did Not Work!

The following section lists some of the techniques we selected over the course of the challenge that did not result in any performance improvement. As a disclaimer we would like to add that it is possible that we did not utilize them with appropriate hyperparameters and it might be possible to get good performance with them.

- **Text Pre-processing:** We found that pre-processing the text data often led to drop in performance of a model in comparison to the one trained as is. We explored **removing digits**, **translating** non-latin characters, **splitting** concatenated words.

- **Architectures:** In our experiments, transformer based networks performed better than their convolution counterparts. We tried ResNet-50, ResNest-101, Mobilenet-v3, and Efficient-b3 but their performance was lower in comparison to ViT-base. Moreover, they often struggled to train and required tweaking of learning rate, scheduling and weight decay. However, we noted that Convnext works remarkably well when trained with appropriate regularization.

- **Augmentations:** We tried AutoAugment [2] and SimCLR [1] augmentation strategies. We speculate that the augmentations drastically changing the color are likely to harm training as, for many pairs, it is an additional cue to link text captions to images.

- **Locking vision transformer** did not provide any performance gain in our experiments. In the original work, the authors mention that employing a strong pretrained vision model and locking it during cross modal training boosts in particular the zero-shot performance.

- **Multilanguage:** Using a multilingual distillBERT resulted in a slight performance drop.

- **Back translation:** In the context of TTA we performed back translation augmentation with no further improvements upon EDA.

5. Conclusion

In this report we have presented the solution we adopted to tackle the CVPR 2022 AliProducts Challenge. We exploited state-of-the-art methodologies and architectures to achieve a score equal to **0.72695** mean recall@5 and recall@10 on the unreleased test set, resulting as the second best method in the AliProducts challenge.

6. Acknowledgement

We would like to thank Andrea Mirabile for many useful discussions, suggestions and also for motivating us to take part in the competition.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3

[2] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019. 2, 3

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805, Oct. 2018. 2

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[5] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. *arXiv e-prints*, page arXiv:2107.07651, July 2021. 1

[6] 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. *arXiv e-prints*, page arXiv:2110.05208, Oct. 2021. 1, 2

[7] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv e-prints*, page arXiv:2103.14030, Mar. 2021. 2

[8] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. *arXiv e-prints*, page arXiv:2201.03545, Jan. 2022. 2

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 2

[10] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. SLIP: Self-supervision meets Language-Image Pre-training. *arXiv e-prints*, page arXiv:2112.12750, Dec. 2021. 1

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *arXiv e-prints*, page arXiv:2103.00020, Feb. 2021. 1, 2

[12] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv e-prints*, page arXiv:1910.01108, Oct. 2019. 2

[13] Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196, 2019. 2

[14] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained Interactive Language-Image Pre-Training. *arXiv e-prints*, page arXiv:2111.07783, Nov. 2021. 2