甘晴void 靖姐 凡哥 小袁喵

github链接：https://github.com/OFA-Sys/Chinese-CLIP

原理部分

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford[*1] Jong Wook Kim[*1] Chris Hallacy[1] Aditya Ramesh[1] Gabriel Goh[1] Sandhini Agarwal[1]
Girish Sastry[1] Amanda Askell[1] Pamela Mishkin[1] Jack Clark[1] Gretchen Krueger[1] Ilya Sutskever[1]

Abstract

State-of-the-art computer vision systems are trained to predict a fixed set of predetermined object categories. This restricted form of supervision limits their generality and usability since additional labeled data is needed to specify any other visual concept. Learning directly from raw text about images is a promising alternative which leverages a much broader source of supervision. We demonstrate that the simple pre-training task of predicting which caption goes with which image is an efficient and scalable way to learn SOTA image representations from scratch on a dataset of 400 million (image, text) pairs collected from the internet. After pre-training, natural language is used to reference learned visual concepts (or describe new ones) enabling zero-shot transfer of the model to downstream tasks. We study the performance of this approach by benchmarking on over 30 different existing computer vision datasets, spanning tasks such as OCR, action recognition in videos, geo-localization, and many types of fine-grained object classification. The model transfers non-trivially to most tasks and is often competitive with a fully supervised baseline without the need for any dataset specific training. For instance, we match the accuracy of the original ResNet-50 on ImageNet zero-shot without needing to use any of the 1.28 million training examples it was trained on. We release our code and pre-trained model weights at https://github.com/OpenAI/CLIP.

1. Introduction and Motivating Work

Pre-training methods which learn directly from raw text have revolutionized NLP over the last few years (Dai & Le, 2015; Peters et al., 2018; Howard & Ruder, 2018; Radford et al., 2018; Devlin et al., 2018; Raffel et al., 2019). Task-agnostic objectives such as autoregressive and masked language modeling have scaled across many orders of magnitude in compute, model capacity, and data, steadily improving capabilities. The development of "text-to-text" as a standardized input-output interface (McCann et al., 2018; Radford et al., 2019; Raffel et al., 2019) has enabled task-agnostic architectures to zero-shot transfer to downstream datasets removing the need for specialized output heads or dataset specific customization. Flagship systems like GPT-3 (Brown et al., 2020) are now competitive across many tasks with bespoke models while requiring little to no dataset specific training data.

These results suggest that the aggregate supervision accessible to modern pre-training methods within web-scale collections of text surpasses that of high-quality crowd-labeled NLP datasets. However, in other fields such as computer vision it is still standard practice to pre-train models on crowd-labeled datasets such as ImageNet (Deng et al., 2009). Could scalable pre-training methods which learn directly from web text result in a similar breakthrough in computer vision? Prior work is encouraging.

Over 20 years ago Mori et al. (1999) explored improving content based image retrieval by training a model to predict the nouns and adjectives in text documents paired with images. Quattoni et al. (2007) demonstrated it was possible to learn more data efficient image representations via manifold learning in the weight space of classifiers trained to predict words in captions associated with images. Srivastava & Salakhutdinov (2012) explored deep representation learning by training multimodal Deep Boltzmann Machines on top of low-level image and text tag features. Joulin et al. (2016) modernized this line of work and demonstrated that CNNs trained to predict words in image captions learn useful image representations. They converted the title, description, and hashtag metadata of images in the YFCC100M dataset (Thomee et al., 2016) into a bag-of-words multi-label classification task and showed that pre-training AlexNet (Krizhevsky et al., 2012) to predict these labels learned representations which preformed similarly to ImageNet-based pre-training on transfer tasks. Li et al. (2017) then extended this approach to predicting phrase n-grams in addition to individual words and demonstrated the ability of their system to zero-shot transfer to other image

[*]Equal contribution [1]OpenAI, San Francisco, CA 94110, USA. Correspondence to: <{alec, jongwook}@openai.com>.

2021 Clip由OpenAI提出

Chinese CLIP: Contrastive Vision-Language Pretraining in Chinese

An Yang[*1], Junshu Pan[*1,2†], Junyang Lin[*1],
Rui Men[1], Yichang Zhang[1], Jingren Zhou[1], Chang Zhou[1♣]
[1]DAMO Academy, Alibaba Group
[2]Beihang University
{ya235025, panjunshu.pjs, junyang.ljy, ericzhou.zc}@alibaba-inc.com

Abstract

The tremendous success of vision-language foundation models has promoted the research and application of computer vision and multimodal representation leearning. However, it is still difficult to effectively transfer such foundation models to language-specific scenarios. In this work, we propose Chinese CLIP with the two-stage pretraining method which trains the model with locked-image tuning in the first stage and contrastive tuning in the second one. Specifically, we have developed 5 Chinese CLIP models of multiple sizes, spanning from 77 to 958 million parameters, and we have pretrained them on a collected large-scale dataset of Chinese image-text pairs. Our comprehensive experiments demonstrate that Chinese CLIP can achieve the state-of-the-art performance on MUGE, Flickr30K-CN, and COCO-CN in the setups of zero-shot learning and finetuning, and it is able to achieve competitive performance in zero-shot image classification based on the evaluation on the ELEVATER benchmark. We have released our codes, models, and demos[1].

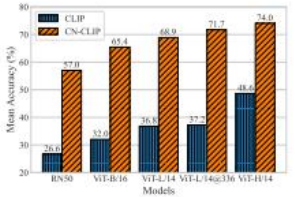


Figure 1: **Comparison of CLIP and Chinese CLIP models on the Chinese native retrieval benchmark MUGE.** On the benchmark based on the native data (which are mostly crawled from the language-native websites, in contrast with the translated data from websites of other countries.), CLIP performs far worse than our Chinese CLIP. Note that CLIP$_{\text{ViT-H/14}}$ is not released, we use the model from OpenCLIP (Ilharco et al., 2021) instead.

1 Introduction

Starting from the burst of pretraining in NLP, foundation models have attracted attention from multiple research communities. Foundation models that learn from large-scale unsupervised or weakly supervised data play as the basis of downstream models. A milestone of foundation models (Bommasani et al., 2021) in multimodal representation learning is CLIP (Radford et al., 2021). Different from the conventional generative pretraining, CLIP is a contrastive-learning-based model pretrained on a large-scale dataset of around 400 million image-text pair data collected from the web. Despite the simplicity of the method, CLIP not only achieved outstanding performance in vision-language retrieval but more importantly played as a vision foundation model and demonstrated state-of-the-art performance in zero-shot image classification across a series of datasets. CLIP which builds a connection between vision and language has been transforming the research in both multimodal representation learning and computer vision.

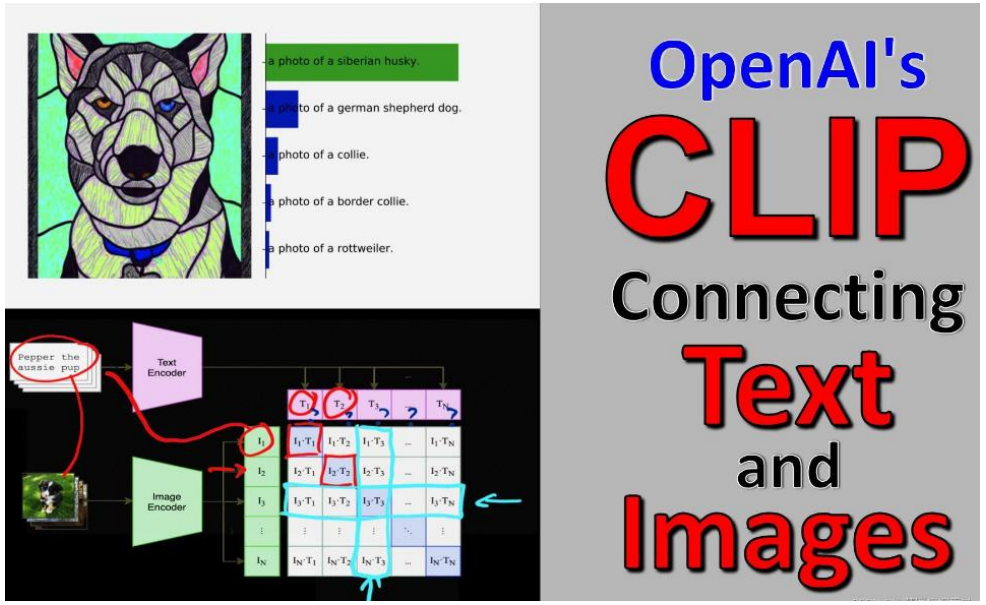
Be that as it may, it is difficult to efficiently transfer a cross-modal pretrained model to another language for several causes. First, learning to model the distribution of language-native vision and language data is significant for the transfer. Though CLIP performs as a strong foundation model in most scenarios, we find that it is hard for CLIP with machine translation to perform well on the Chinese-native cross-modal retrieval benchmark.

[*]Co-first authors.
[♣]Corresponding author.
[†]Work done as an intern in DAMO Academy.
[1]Github: https://github.com/OFA-Sys/Chinese-CLIP; ModelScope: https://www.modelscope.cn/models

2022.7.8 Chinese-CLIP项目正式开源

CLIP模型

- CLIP全名Contrastive Language-Image Pretraining，在2021年由OpenAI提出，其核心理念为图文对比学习预训练，是一种多模态学习模型，旨在将图像和文本进行关联，它可以快速实现图文特征相似度计算、跨模态检索、零样本图片分类等任务。
- 与传统的视觉模型不同，CLIP的预训练数据并非标注的图像数据，而是从网络上大量采集的弱监督图文对数据，即我们常说的图片和标题（字幕）。CLIP收集了4亿规模的图文对数据，旨在通过预训练建模图像与文本的联系。



CLIP模型

- 数据集：来源于互联网上搜集的4亿个image-text对，涵盖了50万个qurey，并尽量保持不同qurey的数据量均衡。
- 相比传统复杂的交互式的多模态预训练，CLIP的模型极为简单，即双塔模型，分别包括图像塔和文本塔。
- 图像塔负责提取图像表征，一般为Vision Transformer，即常说的ViT，文本塔则负责提取文本特征，使用经典Transformer架构。
- 核心思想：将image-text对当做一个整体，基于对比学习的方法，模型训练时尽可能地提高image与对应text的特征相似度，尽可能的降低image与不配对text的相似度。

Clip核心思想（类python代码表示）

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

CLIP效果

更好的线性探测性能

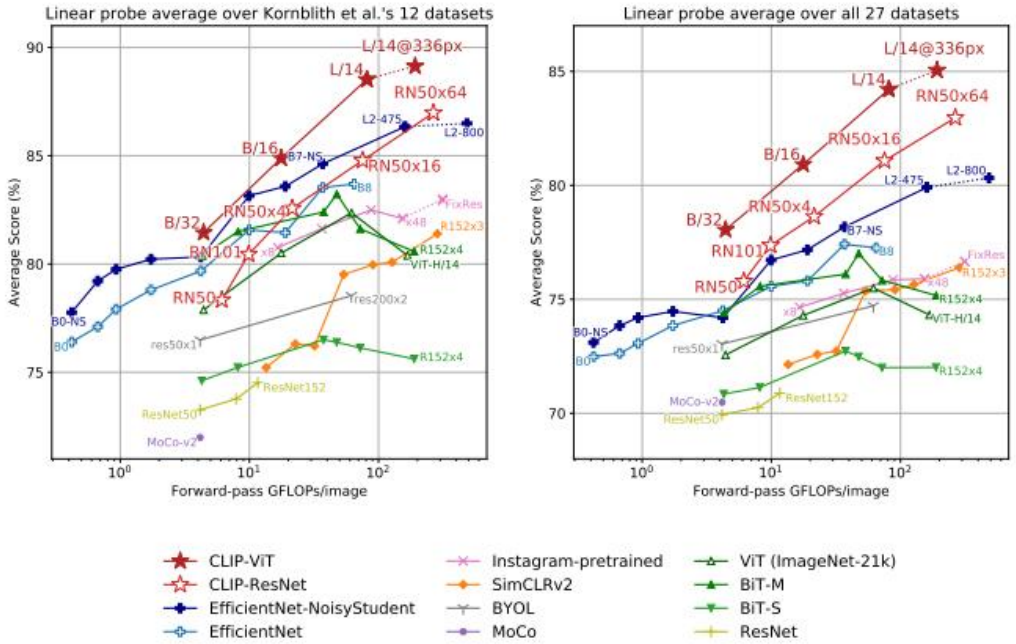


Figure 10. **Linear probe performance of CLIP models in comparison with state-of-the-art computer vision models**, including EfficientNet (Tan & Le, 2019; Xie et al., 2020), MoCo (Chen et al., 2020d), Instagram-pretrained ResNeXt models (Mahajan et al., 2018; Touvron et al., 2019), BiT (Kolesnikov et al., 2019), ViT (Dosovitskiy et al., 2020), SimCLRv2 (Chen et al., 2020c), BYOL (Grill et al., 2020), and the original ResNet models (He et al., 2016b). (Left) Scores are averaged over 12 datasets studied by Kornblith et al. (2019). (Right) Scores are averaged over 27 datasets that contain a wider variety of distributions. Dotted lines indicate models fine-tuned or evaluated on images at a higher-resolution than pre-training. See Table 10 for individual scores and Figure 20 for plots for each dataset.

更好的鲁棒性

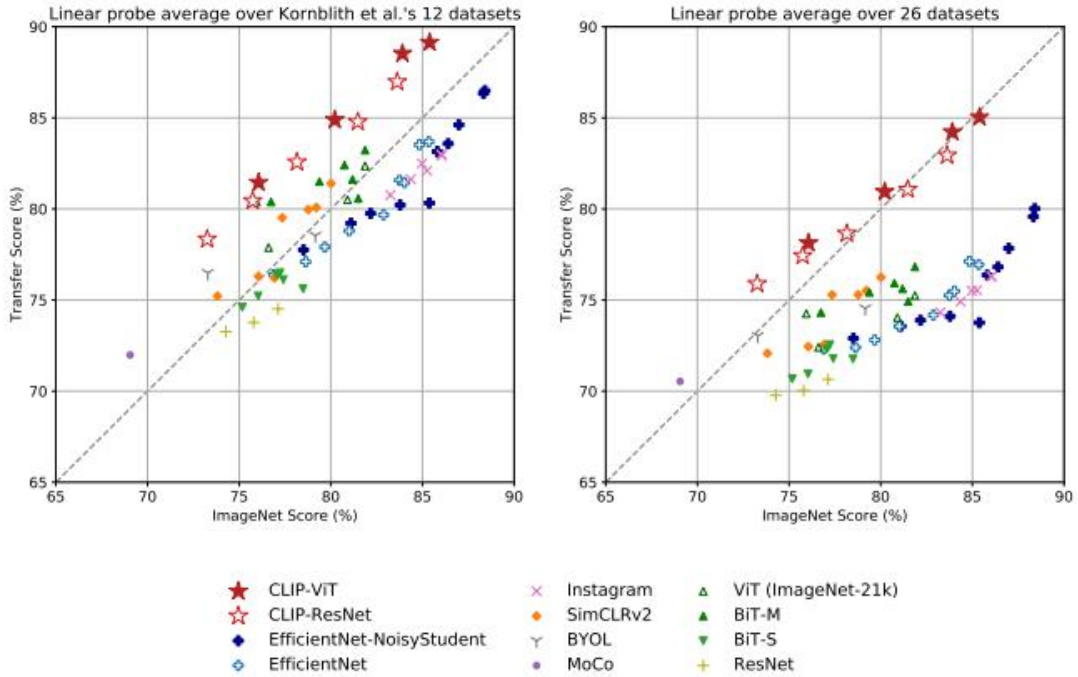
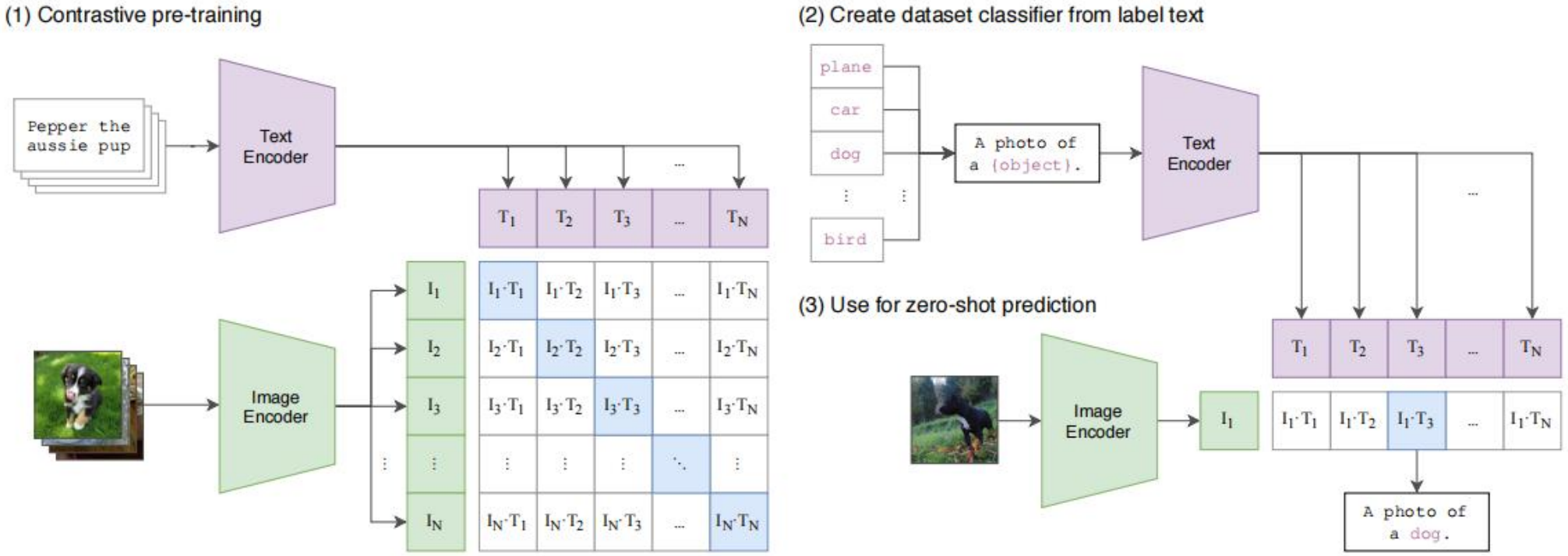


Figure 12. **CLIP's features are more robust to task shift when compared to models pre-trained on ImageNet.** For both dataset splits, the transfer scores of linear probes trained on the representations of CLIP models are higher than other models with similar ImageNet performance. This suggests that the representations of models trained on ImageNet are somewhat overfit to their task.

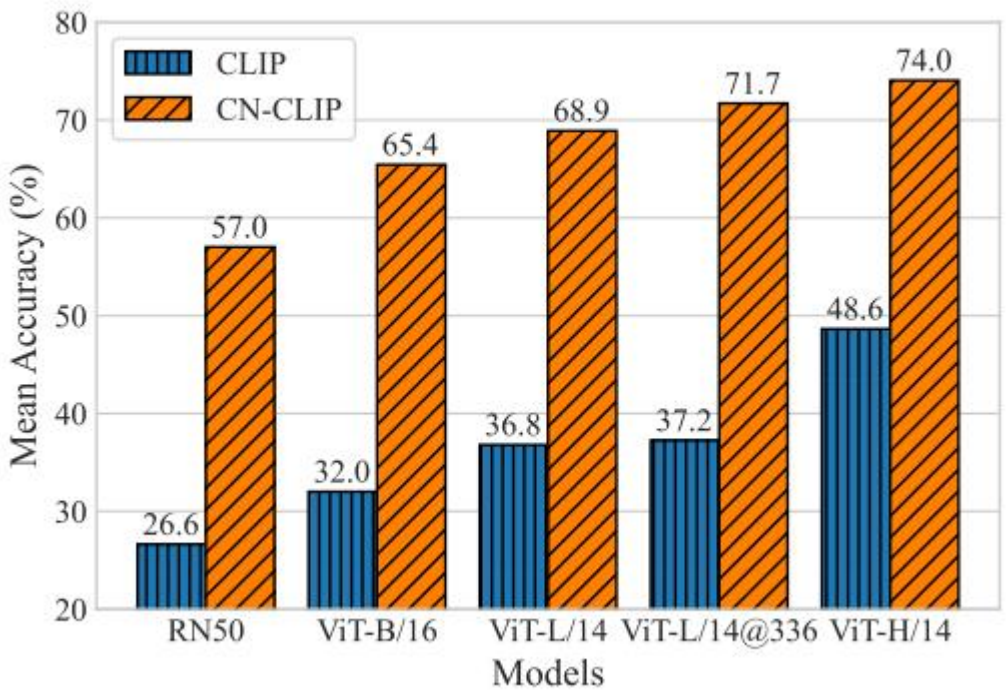
CLIP模型架构



标准图像模型联合训练图像特征提取器和线性分类器来预测某些标签，而CLIP联合训练图像编码器和文本编码器来预测一批（图像、文本）训练示例的正确配对。在测试时，学习到的文本编码器通过嵌入目标数据集的类的名称或描述来合成一个zero-shot的线性分类器。

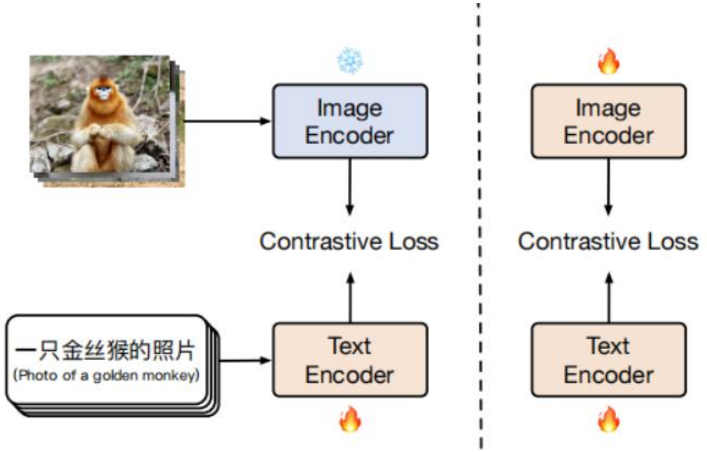
Chinese-CLIP

- 2022年阿里达摩院开源的基于2亿中文原生图文对的多模态预训练模型。
- 数据集：大规模的中文图文对数据（约 2 亿规模），其中包括来自 LAION-5B 中文子集、Wukong 的中文数据、以及来自 COCO、Visual Genome 的翻译图文数据等。
- CLIP和中文CLIP模型的中文本地检索基准MUGE模型的比较（右图）。
- 可见，Chinese-Clip在中文相关问题上效果显著好于原Clip

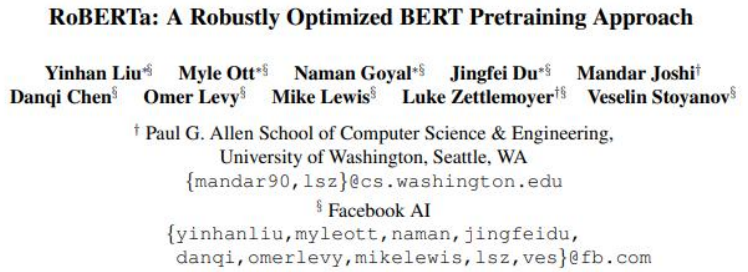


Chinese-CLIP预训练原理

- 为了提升训练效率和模型效果，Chinese-CLIP基于两阶段流程进行训练，具体如下：
- （1）第一阶段，使用已有的图像预训练模型和文本预训练模型分别初始化 Chinese-CLIP 的双塔，并冻结图像侧参数，让语言模型关联上已有的图像预训练表示空间。这是由于CLIP的文本编码器是英文版本训练得到，所以需要对CLIP的文本表征模型进行重新训练。
- （2）第二阶段，解冻图像侧参数，通过对比学习微调中文原生的图像和文本数据。
- Image Encoder: 原有OpenAI CLIP（即VisionTransformer或Resnet）
- Text Encoder:中文RoBERTa（2019）



A Robustly Optimized BERT，即简单粗暴称为强力优化的BERT方法


RoBERTa: A Robustly Optimized BERT Pretraining Approach

Yinhan Liu[*§] Myle Ott[*§] Naman Goyal[*§] Jingfei Du[*§] Mandar Joshi[†]
Danqi Chen[§] Omer Levy[§] Mike Lewis[§] Luke Zettlemoyer[†§] Veselin Stoyanov[§]

[†] Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA
{mandar90,lsz}@cs.washington.edu

[§] Facebook AI
{yinhanliu,myleott,naman,jingfeidu,
danqi,omerlevy,mikelewis,lsz,ves}@fb.com

RoBERTa

RoBERTa: A Robustly Optimized BERT Pretraining Approach

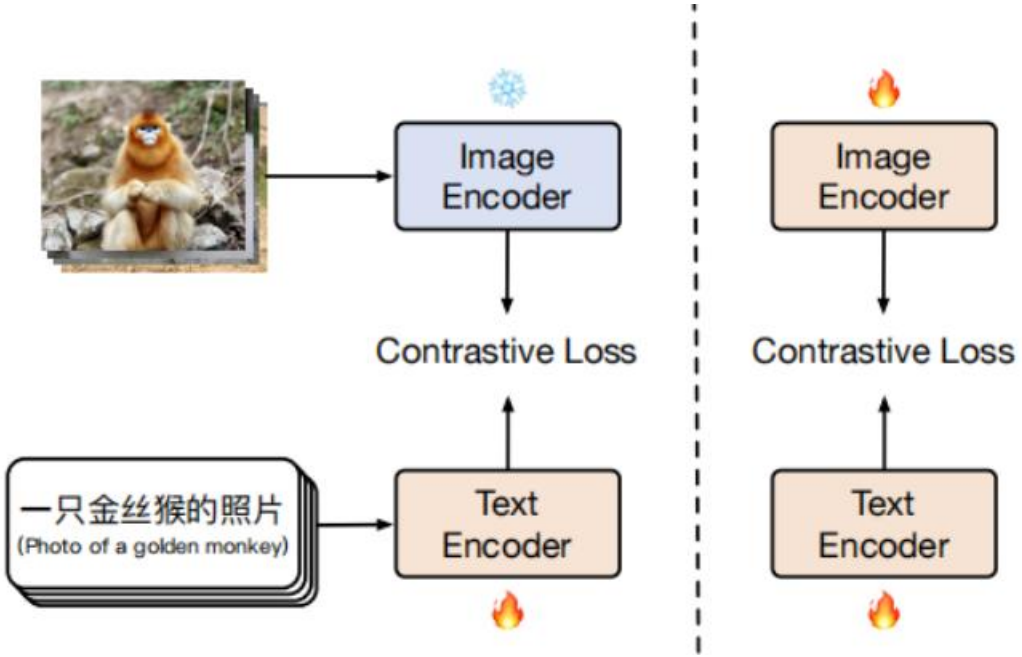
Yinhan Liu[*§] Myle Ott[*§] Naman Goyal[*§] Jingfei Du[*§] Mandar Joshi[†]
Danqi Chen[§] Omer Levy[§] Mike Lewis[§] Luke Zettlemoyer[†§] Veselin Stoyanov[§]

[†] Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA
{mandar90,lsz}@cs.washington.edu

[§] Facebook AI
{yinhanliu,myleott,naman,jingfeidu,
danqi,omerlevy,mikelewis,lsz,ves}@fb.com

- A Robustly Optimized BERT，即简单粗暴称为强力优化的BERT方法
- 在模型规模、算力和数据上，与BERT相比主要有以下几点改进：
- 更大的模型参数量（论文提供的训练时间来看，模型使用 1024 块 V100 GPU 训练了 1 天的时间）
- 更大bacth size。RoBERTa 在训练过程中使用了更大的bacth size。尝试过从 256 到 8000 不等的bacth size。
- 更多的训练数据（包括：CC-NEWS 等在内的 160GB 纯文本。而最初的BERT使用16GB BookCorpus数据集和英语维基百科进行训练）

- 另外，RoBERTa在训练方法上有以下改进：
- 去掉下一句预测(NSP)任务
- 动态掩码。BERT 依赖随机掩码和预测 token。原版的 BERT 实现在数据预处理期间执行一次掩码，得到一个静态掩码。 而 RoBERTa 使用了动态掩码：每次向模型输入一个序列时都会生成新的掩码模式。这样，在大量数据不断输入的过程中，模型会逐渐适应不同的掩码策略，学习不同的语言表征。
- 文本编码。Byte-Pair Encoding（BPE）是字符级和词级别表征的混合，支持处理自然语言语料库中的众多常见词汇。原版的 BERT 实现使用字符级别的 BPE 词汇，大小为 30K，是在利用启发式分词规则对输入进行预处理之后学得的。Facebook 研究者没有采用这种方式，而是考虑用更大的 byte 级别 BPE 词汇表来训练 BERT，这一词汇表包含 50K 的 subword 单元，且没有对输入作任何额外的预处理或分词。



两阶段的训练流程优点如下：

（1）相比从头开始做预训练，该方法在多个下游任务上均展现显著更优的实验效果，而其显著更高的收敛效率也意味着更小的训练成本。

（2）相比全程只训练文本侧做一阶段训练，加入第二阶段微调训练能有效在图文下游任务，尤其是中文原生的图文任务上进一步提升效果。

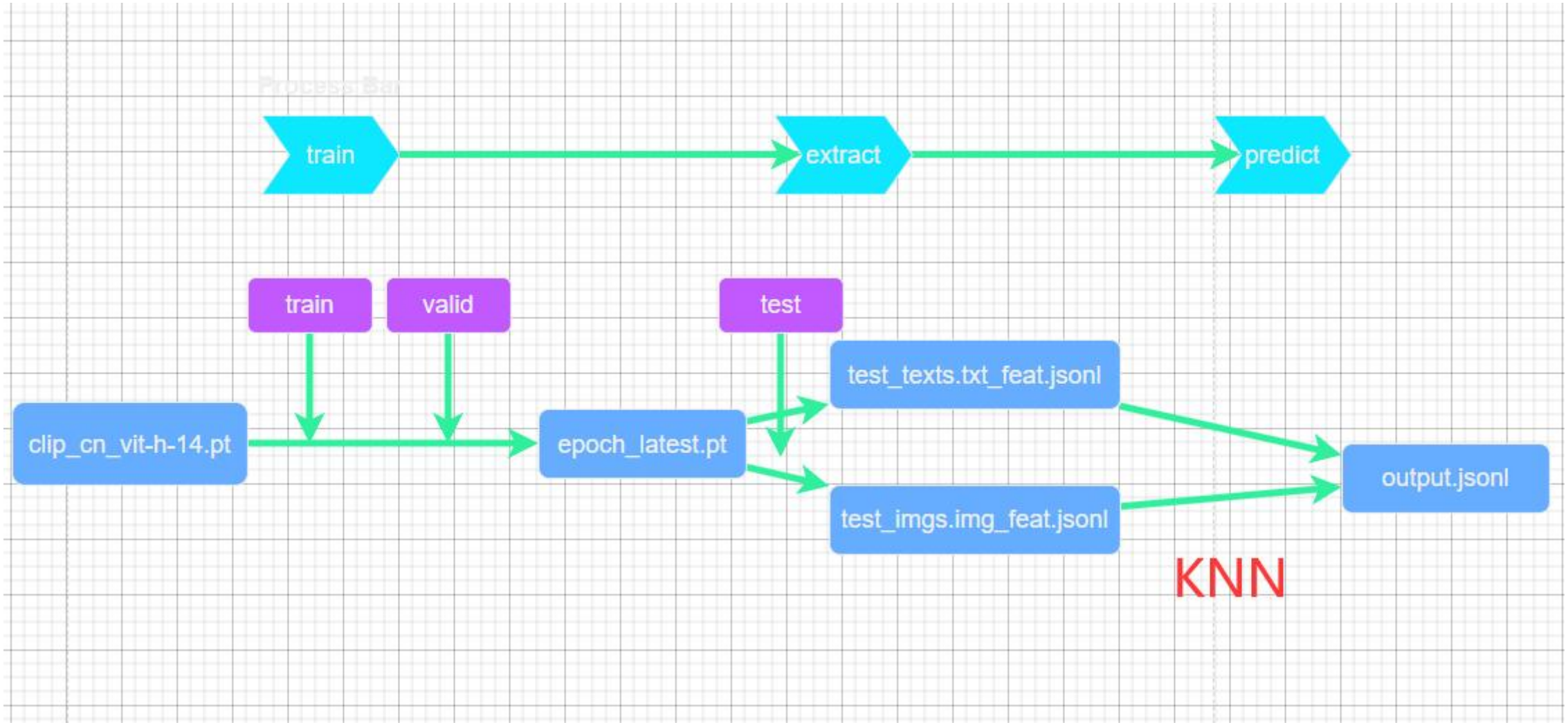
Chinese-CLIP效果

Model	CIFAR10	CIFAR100	DTD	EuroSAT	FER	FGVC	KITTI	MNIST	PC	VOC
Original benchmark										
DeCLIP	90.9	66.8	44.9	39.9	23.3	9.0	39.7	13.6	55.3	80.6
GIT	88.5	61.1	42.9	43.4	41.4	6.7	22.1	68.9	50.0	80.2
ALIGN	**94.9**	76.8	**66.1**	52.1	**50.8**	25.0	**41.2**	74.0	55.2	83.0
OpenCLIP	93.5	76.2	56.4	53.7	50.3	20.8	28.8	70.9	50.5	82.3
CLIP	**94.9**	**77.0**	56.0	**63.0**	48.3	**33.3**	11.5	**79.0**	**62.3**	**84.0**
Translated benchmark										
BriVL	72.3	35.9	18.8	25.5	-	-	-	-	-	-
Wukong	95.4	77.1	40.9	50.3	-	-	-	-	-	-
CN-CLIP	**96.0**	**79.7**	**51.2**	**52.0**	**55.1**	**26.2**	**49.9**	**79.4**	**63.5**	**84.9**

Tasks	Zero-shot				Finetuning			
Metrics	R@1	R@5	R@10	MR	R@1	R@5	R@10	MR
Tiny-size Model								
CN-CLIP$_{\text{RN50}}$	**42.6**	**68.6**	**77.9**	**63.0**	**48.6**	**75.1**	**84.0**	**69.2**
Base-size Models								
Wukong$_{\text{ViT-B/32}}$	33.4	59.3	69.7	54.1	39.2	66.9	77.4	61.2
R2D2$_{\text{ViT-B}}$	-	-	-	-	47.4	75.1	83.5	68.7
CN-CLIP$_{\text{ViT-B/16}}$	**52.1**	**76.7**	**84.4**	**71.1**	**58.4**	**83.6**	**90.0**	**77.4**
Large-size Models								
Wukong$_{\text{ViT-L/14}}$	42.7	69.0	78.0	63.2	52.7	77.9	85.6	72.1
R2D2$_{\text{ViT-L/14}}$	49.5	75.7	83.2	69.5	60.1	82.9	89.4	77.5
CN-CLIP$_{\text{ViT-L/14}}$	56.3	79.8	86.2	74.1	63.3	85.6	91.3	80.1
CN-CLIP$_{\text{ViT-L/14@336px}}$	**59.0**	**81.4**	**87.8**	**76.1**	**65.3**	**86.7**	**92.1**	**81.3**
Huge-size Models								
CN-CLIP$_{\text{ViT-H/14}}$	**63.0**	**84.1**	**89.2**	**78.8**	**68.9**	**88.7**	**93.1**	**83.6**

Tasks	Text-to-Image						Image-to-Text					
Setups	Zero-shot			Finetuning			Zero-shot			Finetuning		
Metrics	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Tiny-size Model												
CN-CLIP$_{\text{RN50}}$	**48.8**	**76.0**	**84.6**	**66.7**	**89.4**	**94.1**	**60.0**	**85.9**	**92.0**	**84.2**	**96.7**	**98.0**
Base-size Models												
Wukong$_{\text{ViT-B/32}}$	45.7	73.8	82.2	67.6	89.6	94.2	66.2	88.7	94.3	83.9	97.6	99.0
R2D2$_{\text{ViT-B}}$	-	-	-	78.3	94.6	97.0	-	-	-	92.6	**99.1**	**99.8**
CN-CLIP$_{\text{ViT-B/16}}$	**62.7**	**86.9**	**92.8**	**79.1**	**94.8**	**97.4**	**74.6**	**93.5**	**97.1**	**93.5**	99.0	99.5
Large-size Models												
Wukong$_{\text{ViT-L/14}}$	51.7	78.9	86.3	77.4	94.5	97.0	76.1	94.8	97.5	92.7	99.1	99.6
R2D2$_{\text{ViT-L/14}}$	60.9	86.8	92.7	**84.4**	96.7	98.4	77.6	96.7	**98.9**	95.6	**99.8**	**100.0**
CN-CLIP$_{\text{ViT-L/14}}$	68.0	89.7	94.4	82.7	96.7	98.6	80.2	96.6	98.2	96.1	99.5	99.9
CN-CLIP$_{\text{ViT-L/14@336px}}$	**69.0**	**90.7**	**95.4**	**84.4**	**97.1**	**98.7**	**83.3**	**97.2**	98.5	**96.6**	**99.8**	**100.0**
Huge-size Models												
CN-CLIP$_{\text{ViT-H/14}}$	**71.2**	**91.4**	**95.5**	**83.8**	**96.9**	**98.6**	**81.6**	**97.5**	**98.8**	**95.3**	**99.7**	**100.0**

实现部分


train
extract
predict
train
valid
test
clip_cn_vit-h-14.pt
epoch_latest.pt
test_texts.txt_feat.jsonl
test_imgs.img_feat.jsonl
output.jsonl
KNN

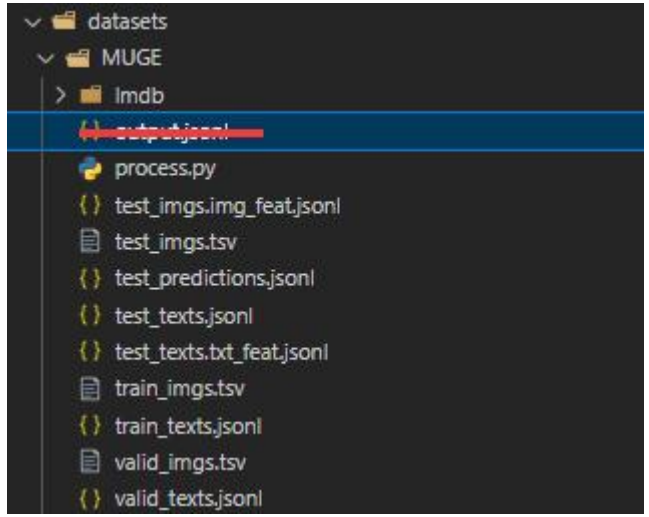
准备

安装相关依赖：pip install -r Chinese-CLIP/requirements.txt

预处理数据：

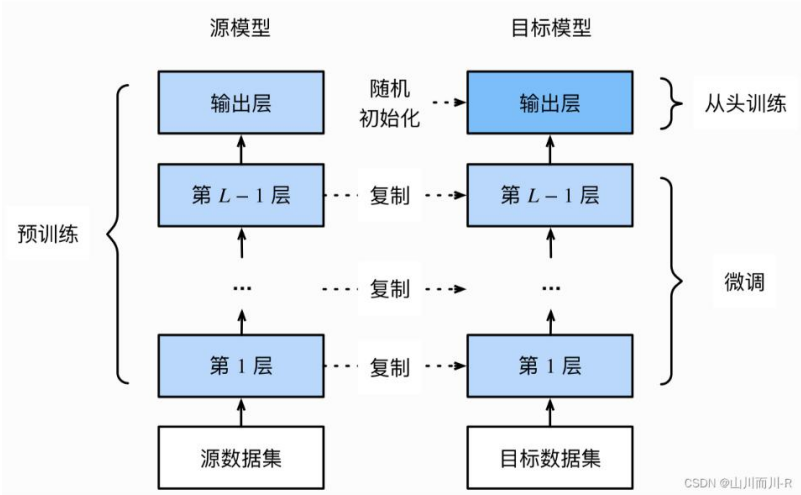
数据集下载文件为：Multimodal_Retrieval.zip，包括：

- MR_train_imgs.tsv：训练集图片集合（base64编码）
- MR_train_queries.jsonl：训练集搜索query及对应商品id
- MR_valid_imgs.tsv：验证集图片集合（base64编码）
- MR_valid_queries.jsonl：验证集搜索query及对应商品id
- MR_test_imgs.tsv：测试集图片集合（base64编码）
- MR_test_queries.jsonl：测试集搜索query，需预测的文件，选手需要补充"item_ids"字段，为list类型。
- example_pred.jsonl：测试集提交结果示例
- README.txt: 说明文件



下载Chinese-CLIP预训练参数（ckpt）

wget https://clip-cn-beijing.oss-cn-beijing.aliyuncs.com/checkpoints/clip_cn_vit-b-16.pt



finetune

```
(base) root@8f788dfb2acd:/home/datatry/cclip-task/Chinese-CLIP# ./run_scripts/my-fintune.sh
W0531 12:50:57.900870 140054594544448 torch/distributed/run.py:757]
W0531 12:50:57.900870 140054594544448 torch/distributed/run.py:757] *****************************************
W0531 12:50:57.900870 140054594544448 torch/distributed/run.py:757] Setting OMP_NUM_THREADS environment variable for each process to be 1 in default, to avoid your syste
timal performance in your application as needed.
W0531 12:50:57.900870 140054594544448 torch/distributed/run.py:757] *****************************************
Loading vision model config from cn_clip/clip/model_configs/ViT-H-14.json
Loading text model config from cn_clip/clip/model_configs/RoBERTa-wwm-ext-large-chinese.json
Loading vision model config from cn_clip/clip/model_configs/ViT-H-14.json
Loading text model config from cn_clip/clip/model_configs/RoBERTa-wwm-ext-large-chinese.json
Loading vision model config from cn_clip/clip/model_configs/ViT-H-14.json
Loading text model config from cn_clip/clip/model_configs/RoBERTa-wwm-ext-large-chinese.json
Loading vision model config from cn_clip/clip/model_configs/ViT-H-14.json
Loading text model config from cn_clip/clip/model_configs/RoBERTa-wwm-ext-large-chinese.json
2024-05-31,12:51:11 | INFO | Rank 0 | Grad-checkpointing activated.
2024-05-31,12:51:11 | INFO | Rank 3 | Grad-checkpointing activated.
2024-05-31,12:51:11 | INFO | Rank 2 | Grad-checkpointing activated.
2024-05-31,12:51:11 | INFO | Rank 1 | Grad-checkpointing activated.
2024-05-31,12:51:13 | INFO | Rank 3 | train LMDB file contains 129380 images and 250314 pairs.
2024-05-31,12:51:13 | INFO | Rank 2 | train LMDB file contains 129380 images and 250314 pairs.
2024-05-31,12:51:13 | INFO | Rank 0 | train LMDB file contains 129380 images and 250314 pairs.
2024-05-31,12:51:13 | INFO | Rank 3 | val LMDB file contains 29806 images and 30588 pairs.
2024-05-31,12:51:13 | INFO | Rank 3 | Use GPU: 3 for training
2024-05-31,12:51:13 | INFO | Rank 3 | => begin to load checkpoint '../datapath//pretrained_weights/clip_cn_vit-h-14.pt'
2024-05-31,12:51:13 | INFO | Rank 2 | val LMDB file contains 29806 images and 30588 pairs.
2024-05-31,12:51:13 | INFO | Rank 2 | Use GPU: 2 for training
2024-05-31,12:51:13 | INFO | Rank 2 | => begin to load checkpoint '../datapath//pretrained_weights/clip_cn_vit-h-14.pt'
2024-05-31,12:51:13 | INFO | Rank 0 | val LMDB file contains 29806 images and 30588 pairs.
2024-05-31,12:51:13 | INFO | Rank 0 | Params:
2024-05-31,12:51:13 | INFO | Rank 0 |   accum_freq: 1
2024-05-31,12:51:13 | INFO | Rank 0 |   aggregate: True
2024-05-31,12:51:13 | INFO | Rank 0 |   batch_size: 48
2024-05-31,12:51:13 | INFO | Rank 0 |   bert_weight_path: None
2024-05-31,12:51:13 | INFO | Rank 0 |   beta1: 0.9
2024-05-31,12:51:13 | INFO | Rank 0 |   beta2: 0.98
2024-05-31,12:51:13 | INFO | Rank 0 |   checkpoint_path: ./logs/muge_finetune_vit-h-14_roberta-huge_bs48_1gpu/checkpoints
2024-05-31,12:51:13 | INFO | Rank 0 |   clip_weight_path: None
2024-05-31,12:51:13 | INFO | Rank 0 |   context_length: 52
2024-05-31,12:51:13 | INFO | Rank 0 |   debug: False
2024-05-31,12:51:13 | INFO | Rank 0 |   device: cuda:0
2024-05-31,12:51:13 | INFO | Rank 0 |   distllation: False
2024-05-31,12:51:13 | INFO | Rank 0 |   eps: 1e-06
2024-05-31,12:51:13 | INFO | Rank 0 |   freeze_vision: False
2024-05-31,12:51:13 | INFO | Rank 0 |   gather_with_grad: False
2024-05-31,12:51:13 | INFO | Rank 0 |   grad_checkpointing: True
2024-05-31,12:51:13 | INFO | Rank 0 |   kd_loss_weight: 0.5
2024-05-31,12:51:13 | INFO | Rank 0 |   local_device_rank: 0
2024-05-31,12:51:13 | INFO | Rank 0 |   log_interval: 10
2024-05-31,12:51:13 | INFO | Rank 0 |   log_level: 20
2024-05-31,12:51:13 | INFO | Rank 0 |   log_path: ./logs/muge_finetune_vit-h-14_roberta-huge_bs48_1gpu/out_2024-05-31-12-51-01.log
2024-05-31,12:51:13 | INFO | Rank 0 |   logs: ./logs/
2024-05-31,12:51:13 | INFO | Rank 0 |   lr: 3e-06
2024-05-31,12:51:13 | INFO | Rank 0 |   mask_ratio: 0
2024-05-31,12:51:13 | INFO | Rank 0 |   max_epochs: 1
```

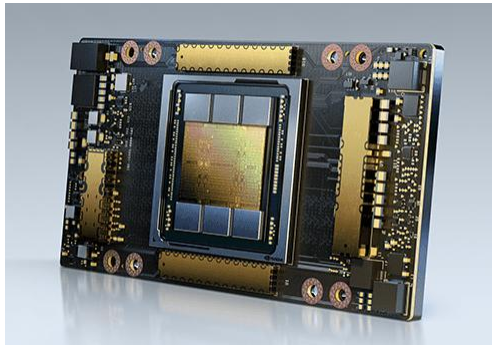
```
export PYTHONPATH=${PYTHONPATH}:`pwd`/cn_clip;
torchrun --nproc_per_node=4
--nnodes=1 --node_rank=0
--master_addr=localhost
--master_port=9010 cn_clip/training/main.py
--train-data=../datapath//datasets/MUGE/lmdb/train
--val-data=../datapath//datasets/MUGE/lmdb/valid
--num-workers=4
--valid-num-workers=4
--resume=../datapath//pretrained_weights/clip_cn_vit-h-14.pt
--reset-data-offset
--reset-optimizer
--name=muge_finetune_vit-h-14_roberta-huge_bs48_1gpu
--save-step-frequency=999999
--save-epoch-frequency=1
--log-interval=10
--report-training-batch-acc
--context-length=52
--warmup=100
--batch-size=48
--valid-batch-size=48
--valid-step-interval=1000
--valid-epoch-interval=1
--lr=3e-06
--wd=0.001
--max-epochs=1
--vision-model=ViT-H-14
--use-augment
--grad-checkpointing
--text-model=RoBERTa-wwm-ext-large-chinese
```

性能分析

```
2024-05-31,12:51:25 | INFO | Rank 0 | Global Steps: 10/1304 | Train Epoch: 1 [1920/250368 (1%)] | Loss: 0.804866 | Image2Text Acc: 76.56 | Text2Image Acc: 79.69 | Data Time: 0.064s | Batch Time: 0.789s | LR: 0.000000 | logit_sca
le: 4.605 | Global Batch Size: 192
2024-05-31,12:51:33 | INFO | Rank 0 | Global Steps: 20/1304 | Train Epoch: 1 [3840/250368 (2%)] | Loss: 1.095239 | Image2Text Acc: 75.00 | Text2Image Acc: 75.00 | Data Time: 0.063s | Batch Time: 0.791s | LR: 0.000001 | logit_sca
le: 4.605 | Global Batch Size: 192
2024-05-31,12:51:41 | INFO | Rank 0 | Global Steps: 30/1304 | Train Epoch: 1 [5760/250368 (2%)] | Loss: 0.724124 | Image2Text Acc: 81.25 | Text2Image Acc: 81.77 | Data Time: 0.063s | Batch Time: 0.791s | LR: 0.000001 | logit_sca
le: 4.605 | Global Batch Size: 192
2024-05-31,12:51:49 | INFO | Rank 0 | Global Steps: 40/1304 | Train Epoch: 1 [7680/250368 (3%)] | Loss: 0.700446 | Image2Text Acc: 82.81 | Text2Image Acc: 84.38 | Data Time: 0.063s | Batch Time: 0.790s | LR: 0.000001 | logit_sca
le: 4.605 | Global Batch Size: 192
2024-05-31,12:51:57 | INFO | Rank 0 | Global Steps: 50/1304 | Train Epoch: 1 [9600/250368 (4%)] | Loss: 0.637871 | Image2Text Acc: 79.69 | Text2Image Acc: 83.85 | Data Time: 0.064s | Batch Time: 0.790s | LR: 0.000002 | logit_sca
le: 4.605 | Global Batch Size: 192
2024-05-31,12:52:05 | INFO | Rank 0 | Global Steps: 60/1304 | Train Epoch: 1 [11520/250368 (5%)] | Loss: 0.677985 | Image2Text Acc: 82.29 | Text2Image Acc: 83.33 | Data Time: 0.063s | Batch Time: 0.792s | LR: 0.000002 | logit_sc
ale: 4.605 | Global Batch Size: 192
2024-05-31,12:52:13 | INFO | Rank 0 | Global Steps: 70/1304 | Train Epoch: 1 [13440/250368 (5%)] | Loss: 0.692282 | Image2Text Acc: 79.17 | Text2Image Acc: 78.65 | Data Time: 0.063s | Batch Time: 0.823s | LR: 0.000002 | logit_sc
```

```
2024-05-31,13:10:36 | INFO | Rank 0 | Global Steps: 1230/1304 | Train Epoch: 1 [236160/250368 (94%)] | Loss: 0.540962 | Image2Text Acc: 82.29 | Text2Image Acc: 88.02 | Data Time: 0.066s | Batch Time: 0.802s | LR: 0.000000 | logi
t_scale: 4.605 | Global Batch Size: 192
2024-05-31,13:10:44 | INFO | Rank 0 | Global Steps: 1240/1304 | Train Epoch: 1 [238080/250368 (95%)] | Loss: 0.373845 | Image2Text Acc: 91.15 | Text2Image Acc: 88.02 | Data Time: 0.065s | Batch Time: 0.803s | LR: 0.000000 | logi
t_scale: 4.605 | Global Batch Size: 192
2024-05-31,13:10:52 | INFO | Rank 0 | Global Steps: 1250/1304 | Train Epoch: 1 [240000/250368 (96%)] | Loss: 0.580477 | Image2Text Acc: 82.81 | Text2Image Acc: 84.38 | Data Time: 0.064s | Batch Time: 0.853s | LR: 0.000000 | logi
t_scale: 4.605 | Global Batch Size: 192
2024-05-31,13:11:00 | INFO | Rank 0 | Global Steps: 1260/1304 | Train Epoch: 1 [241920/250368 (97%)] | Loss: 0.437408 | Image2Text Acc: 88.02 | Text2Image Acc: 89.06 | Data Time: 0.065s | Batch Time: 0.804s | LR: 0.000000 | logi
t_scale: 4.605 | Global Batch Size: 192
2024-05-31,13:11:08 | INFO | Rank 0 | Global Steps: 1270/1304 | Train Epoch: 1 [243840/250368 (97%)] | Loss: 0.525449 | Image2Text Acc: 84.90 | Text2Image Acc: 85.94 | Data Time: 0.065s | Batch Time: 0.802s | LR: 0.000000 | logi
t_scale: 4.605 | Global Batch Size: 192
2024-05-31,13:11:16 | INFO | Rank 0 | Global Steps: 1280/1304 | Train Epoch: 1 [245760/250368 (98%)] | Loss: 0.531416 | Image2Text Acc: 84.38 | Text2Image Acc: 82.29 | Data Time: 0.065s | Batch Time: 0.792s | LR: 0.000000 | logi
t_scale: 4.605 | Global Batch Size: 192
2024-05-31,13:11:24 | INFO | Rank 0 | Global Steps: 1290/1304 | Train Epoch: 1 [247680/250368 (99%)] | Loss: 0.459088 | Image2Text Acc: 84.38 | Text2Image Acc: 85.94 | Data Time: 0.065s | Batch Time: 0.803s | LR: 0.000000 | logi
t_scale: 4.605 | Global Batch Size: 192
2024-05-31,13:11:32 | INFO | Rank 0 | Global Steps: 1300/1304 | Train Epoch: 1 [249600/250368 (100%)] | Loss: 0.397251 | Image2Text Acc: 88.02 | Text2Image Acc: 89.06 | Data Time: 0.065s | Batch Time: 0.795s | LR: 0.000000 | log
it_scale: 4.605 | Global Batch Size: 192
2024-05-31,13:11:35 | INFO | Rank 0 | Begin to eval on validation set (epoch 1 @ 1304 steps)...
2024-05-31,13:11:35 | INFO | Rank 1 | Begin to eval on validation set (epoch 1 @ 1304 steps)...
2024-05-31,13:11:36 | INFO | Rank 3 | Begin to eval on validation set (epoch 1 @ 1304 steps)...
2024-05-31,13:11:36 | INFO | Rank 2 | Begin to eval on validation set (epoch 1 @ 1304 steps)...
2024-05-31,13:13:21 | INFO | Rank 2 | Evaluated 100/160 batches...
2024-05-31,13:13:22 | INFO | Rank 3 | Evaluated 100/160 batches...
2024-05-31,13:13:22 | INFO | Rank 0 | Evaluated 100/160 batches...
2024-05-31,13:13:23 | INFO | Rank 1 | Evaluated 100/160 batches...
2024-05-31,13:14:28 | INFO | Rank 2 | Validation Result (epoch 1 @ 1304 steps) | Valid Loss: 0.152322 | Image2Text Acc: 95.44 | Text2Image Acc: 94.74 | logit_scale: 4.605 | Valid Batch Size: 48
2024-05-31,13:14:28 | INFO | Rank 0 | Validation Result (epoch 1 @ 1304 steps) | Valid Loss: 0.152322 | Image2Text Acc: 95.44 | Text2Image Acc: 94.74 | logit_scale: 4.605 | Valid Batch Size: 48
2024-05-31,13:14:28 | INFO | Rank 3 | Validation Result (epoch 1 @ 1304 steps) | Valid Loss: 0.152322 | Image2Text Acc: 95.44 | Text2Image Acc: 94.74 | logit_scale: 4.605 | Valid Batch Size: 48
2024-05-31,13:14:28 | INFO | Rank 1 | Validation Result (epoch 1 @ 1304 steps) | Valid Loss: 0.152322 | Image2Text Acc: 95.44 | Text2Image Acc: 94.74 | logit_scale: 4.605 | Valid Batch Size: 48
Exception in thread Exception in thread 2024-05-31,13:14:35 | INFO | Rank 0 | Saved checkpoint ./logs/muge_finetune_vit-h-14_roberta-huge_bs48_1gpu/checkpoints/epoch1.pt (epoch 1 @ 1304 steps) (writing took 7.041405200958252 sec
onds)
2024-05-31,13:14:41 | INFO | Rank 0 | Saved checkpoint ./logs/muge_finetune_vit-h-14_roberta-huge_bs48_1gpu/checkpoints/epoch_latest.pt (epoch 1 @ 1304 steps) (writing took 6.167497396469116 seconds)
Exception in thread (base) root@8f788dfb2acd:/home/datatry/cclip-task/Chinese-CLIP# ./run_scripts/my-extract.sh
```

finetune阶段耗时约20mins
（NVIDIA A100-SXM4-40GB）



特征提取

export PYTHONPATH=${PYTHONPATH}:`pwd`/cn_clip;
python -u cn_clip/eval/extract_features.py --extract-image-feats
--extract-text-feats --image-data="../datapath/datasets/MUGE/lmdb/test/imgs"
--text-data="../datapath/datasets/MUGE/test_texts.jsonl" --img-batch-size=32
--text-batch-size=32 --context-length=52
--resume=logs/muge_finetune_vit-h-14_roberta-huge_bs48_1gpu/checkpoints/epoch_latest.pt --vision-model=ViT-H-14
--text-model=RoBERTa-wwm-ext-large-chinese

```
(base) root@8f788dfb2acd:/home/datatry/cclip-task/Chinese-CLIP# ./run_scripts/my-extract.sh
Params:
  context_length: 52
  debug: False
  extract_image_feats: True
  extract_text_feats: True
  image_data: ../datapath/datasets/MUGE/lmdb/test/imgs
  image_feat_output_path: None
  img_batch_size: 32
  precision: amp
  resume: logs/muge_finetune_vit-h-14_roberta-huge_bs48_1gpu/checkpoints/epoch_latest.pt
  text_batch_size: 32
  text_data: ../datapath/datasets/MUGE/test_texts.jsonl
  text_feat_output_path: None
  text_model: RoBERTa-wwm-ext-large-chinese
  vision_model: ViT-H-14
Loading vision model config from cn_clip/clip/model_configs/ViT-H-14.json
Loading text model config from cn_clip/clip/model_configs/RoBERTa-wwm-ext-large-chinese.json
Preparing image inference dataset.
Preparing text inference dataset.
Begin to load model checkpoint from logs/muge_finetune_vit-h-14_roberta-huge_bs48_1gpu/checkpoints/epoch_latest.pt.
=> loaded checkpoint 'logs/muge_finetune_vit-h-14_roberta-huge_bs48_1gpu/checkpoints/epoch_latest.pt' (epoch 1 @ 1304 steps)
Make inference for texts...
100%|                                                                  | 157
/157 [00:15<00:00, 10.02it/s]
5004 text features are stored in ../datapath/datasets/MUGE/test_texts.txt_feat.jsonl
Make inference for images...
100%|                                                                  | 950
/950 [11:09<00:00,  1.42it/s]
30399 image features are stored in ../datapath/datasets/MUGE/test_imgs.img_feat.jsonl
Done!
```

预测

```
(base) root@8f788dfb2acd:/home/datatry/cclip-task/Chinese-CLIP# ./run_scripts/my-predict.sh
Params:
  eval_batch_size: 32768
  image_feats: ../datapath/datasets/MUGE/test_imgs.img_feat.jsonl
  output: ../datapath/datasets/MUGE/test_predictions.jsonl
  text_feats: ../datapath/datasets/MUGE/test_texts.txt_feat.jsonl
  top_k: 10
Begin to load image features...
30399it [00:10, 2784.06it/s]
Finished loading image features.
Begin to compute top-10 predictions for texts...
5004it [01:54, 43.73it/s]
Top-10 predictions are saved in ../datapath/datasets/MUGE/test_predictions.jsonl
Done!
```

```
export PYTHONPATH=${PYTHONPATH}:`pwd`/cn_clip;
python -u cn_clip/eval/make_topk_predictions.py
--image-
feats="../datapath/datasets/MUGE/test_imgs.img_feat.jsonl"
--text-
feats="../datapath/datasets/MUGE/test_texts.txt_feat.jsonl"
--top-k=10
--eval-batch-size=32768
--output="../datapath/datasets/MUGE/test_predictions.jsonl"
```

评测

日期: 2024-05-31 21:53:56

MeanRecall: 81.7812

Recall@1: 67.2662

Recall@5: 86.4309

Recall@10: 91.6467

排名	参与者	组织	MeanRecall	Recall@1	Recall@5	Rec
1	fhudafdkbv	西安电子科技大学	93.88	89.73	95.04	96.8
2	🐱照猫画虎搭积木	南京大学	84.32	70.92	88.83	93.2
3	🐱Jaskr		84.09	70.38	88.85	93.0
3	jasperisgood		84.09	70.38	88.85	93.0
5	🐱tttyyy		83.25	69.46	87.75	92.5
6	🐱超级小火龙	湖南大学	82.58	68.29	87.23	92.2
7	aliyun5364406865	国防科大	82.43	68.27	86.87	92.1
8	🐱404_NOT_FOUND	武汉理工	82.29	67.95	86.89	92.0
9	🐱66666666666	清华大学	82.08	67.91	86.57	91.7
9	游客ym2rvefokjr6i		82.08	67.59	86.93	91.7
11	🐱transformer_is_all_u_need	湖南大学	81.78	67.27	86.43	91.6
12	aliyun1077268045	深圳大学	81.73	66.61	86.93	91.6
13	🐱深度学习小组		81.37	66.51	86.33	91.2
14	🐱X.L.D.	湖南大学	81.36	66.41	86.47	91.2
15	🐱BJTUqll	北京交通大学	81.01	66.23	85.89	90.9