深入了解视觉语言模型

人类学习本质上是多模态 (multi-modal) 的，因为联合利用多种感官有助于我们更好地理解和分析新信息。理所当然地，多模态学习的最新进展即是从这一人类学习过程的有效性中汲取灵感，创建可以利用图像、视频、文本、音频、肢体语言、面部表情和生理信号等各种模态信息来处理和链接信息的模型。

自 2021 年以来，我们看到大家对结合视觉和语言模态的模型 (也称为联合视觉语言模型) 的兴趣越来越浓，一个例子就是 OpenAI 的 CLIP。联合视觉语言模型在非常具有挑战性的任务中表现出了让人眼前一亮的能力，诸如图像标题生成、文本引导图像生成、文本引导图像操作以及视觉问答等。这个领域在不断发展，其零样本泛化能力也在不断改进，从而产生了各种实际应用。

本文，我们将介绍联合视觉语言模型，重点关注它们的训练方式。我们还将展示如何利用 🤗 Transformers 对该领域的最新进展进行实验。

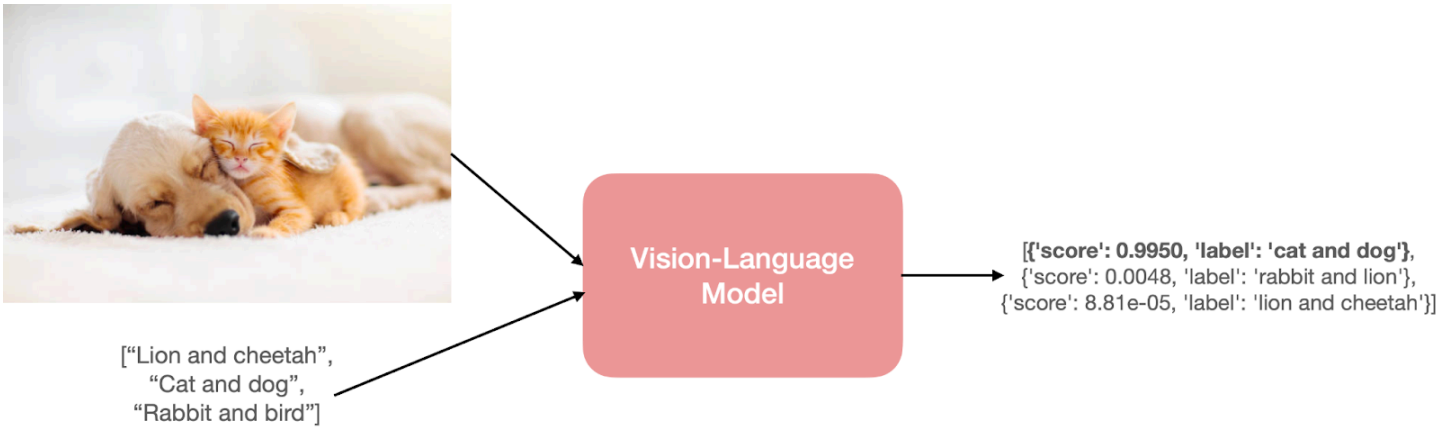
目录

简介

将模型称为 “视觉语言” 模型是什么意思？一个结合了视觉和语言模态的模型？但这到底是什么意思呢?

有助于定义此类模型的一个特性是它们处理图像 (视觉) 和自然语言文本 (语言) 的能力。而这个过程体现在输入、输出以及要求这些模型执行的任务上。

以零样本图像分类任务为例。我们将传给模型如下一张图像和一些候选提示 (prompt)，以获得与输入图像最匹配的提示。



小动物图片 出处

为了预测类似的东西，模型需要理解输入图像和文本提示。它将使用单独或融合的视觉和语言编码器来达到理解的目的。

输入和输出可以有多种形式。下面仅举几例:

- 用自然语言文本来检索图像。
- 短语关联 (Phrase grounding)，即在输入图像中检测出文本中提到的短语 (例如: 一个 **年轻人** 挥动 **球拍**)。
- 视觉问答，即在输入图像中找到自然语言问题的答案。
- 为给定图像生成标题。该任务还有一个形式就是条件文本生成，此时输入变成了两个，即自然语言提示和图像。
- 从包含图像和文本模态的社交媒体内容中检测仇恨言论。

学习策略

视觉语言模型通常由 3 个关键元素组成: 图像编码器、文本编码器以及融合两个编码器的信息的策略。这些关键元素紧密耦合在一起，因为损失函数是围绕模型架构和学习策略设计的。虽然视觉语言模型研究算不上是一个新的研究领域，但此类模型的设计随着时间的变迁发生了巨大变化。早期的研究采用手工设计的图像描述子、预训练词向量或基于频率的 TF-IDF 特征，而最新的研究主要采用 Transformer 架构的图像和文本编码器来单独或联合学习图像和文本特征。我们使用战略性的预训练目标来训练这些模型，从而使之可用于各种下游任务。

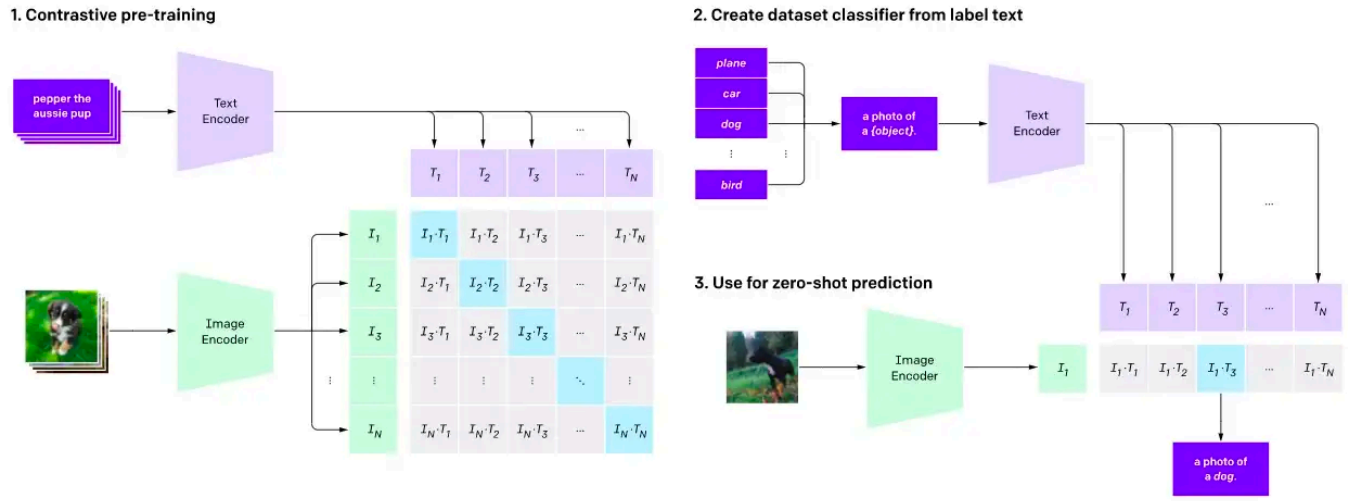
在本节中，我们将讨论视觉语言模型的一些典型预训练目标和策略，这些模型已被证明有良好的迁移性能。我们还将讨论其他有趣的东西，它们要么特定于某些预训练目标，要么可以用作预训练的通用组件。

我们将在预训练目标中涵盖以下主题:

- **对比学习**: 以对比方式将图像和文本对齐到联合特征空间
- **PrefixLM**: 通过将图像视作语言模型的前缀来联合学习图像和文本嵌入
- **基于交叉注意力的多模态融合**: 将视觉信息融合到具有交叉注意力机制的语言模型的各层中
- **MLM / ITM**: 使用掩码语言建模 (Masked-Language Modeling，MLM) 和图像文本匹配 (Image-Text Matching，ITM) 目标将图像的各部分与文本对齐
- **无训练**: 通过迭代优化来利用独立视觉和语言模型

请注意，本节并未详尽陈述所有方法，还有各种其他方法以及混合策略，例如 Unified-IO。如需更全面地了解多模态模型，请参阅 此项工作。

1) 对比学习



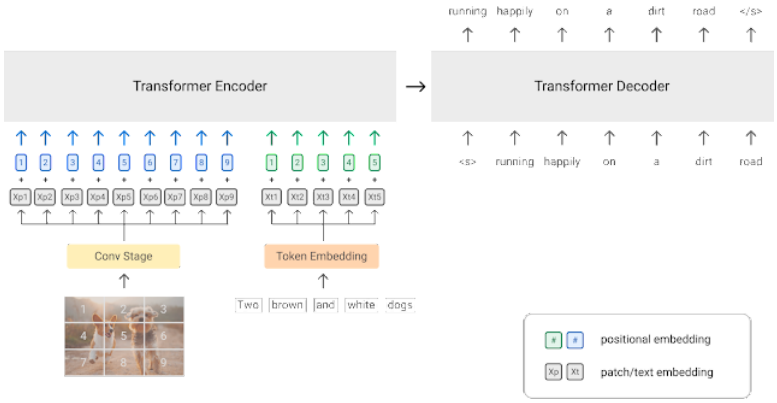
对比预训练和零样本图像分类有关内容参见 这里

对比学习是视觉模型常用的预训练目标，也已被证明同时是视觉语言模型的高效预训练目标。近期的工作如 CLIP、CLOOB、ALIGN 和 DeCLIP 在 {图像，标题} 对组成的大型数据集上，通过使用对比损失函数联合训练文本编码器和图像编码器，从而桥接视觉和语言两个模态。对比学习旨在将输入图像和文本映射到相同的特征空间，使得图像 - 文本对的嵌入之间的距离在两者匹配时最小化，而在不匹配时最大化。

CLIP 仅采用文本和图像嵌入之间的余弦距离作为距离度量。而 ALIGN 和 DeCLIP 等模型则设计了自己的距离度量，这些距离在设计时考虑了数据集是有噪声的。

另一项工作 LiT 引入了一种冻结图像编码器而仅使用 CLIP 预训练目标来微调文本编码器的简单方法。作者将这个想法解释为 一种教文本编码器更好地读懂图像编码器生成的图像嵌入的方法。这种方法已被证明是有效的，并且比 CLIP 的样本效率更高。FLAVA 等其他工作将对比学习和其他预训练策略相结合来对齐视觉和语言嵌入。

2) PrefixLM



展示 PrefixLM 预训练策略的图片 (出处)

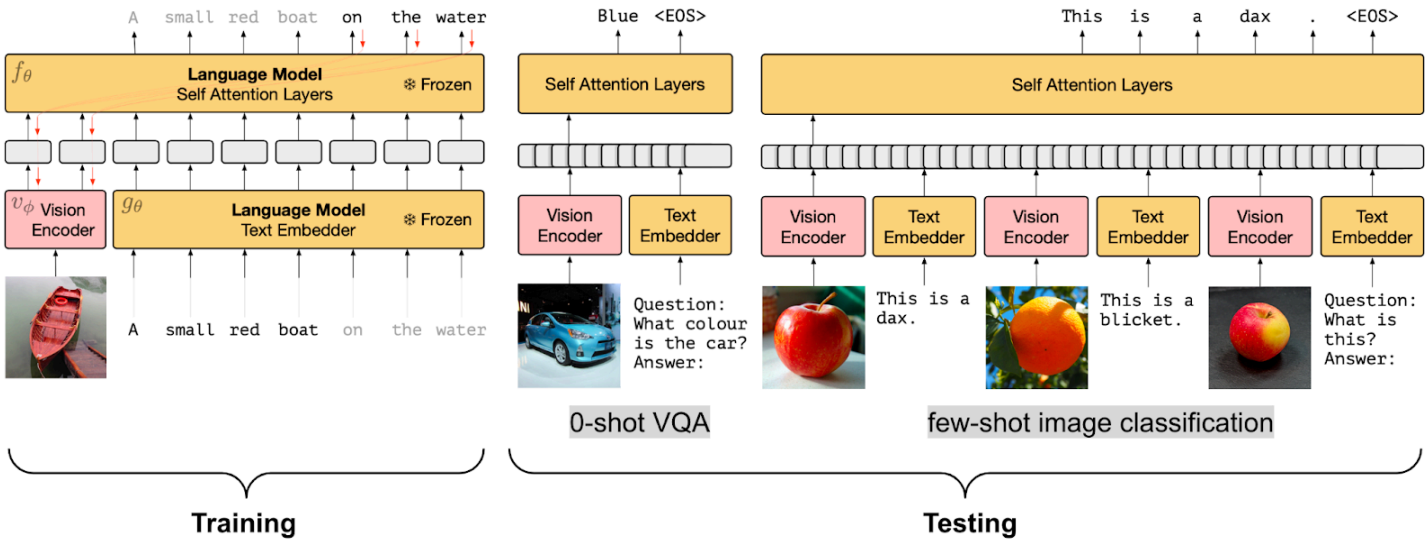
另一种训练视觉语言模型的方法是使用 PrefixLM 目标。SimVLM 和 VirTex 等模型使用该预训练目标并使用一个统一的由 transformer 编码器和 transformer 解码器组成的多模态架构，有点类似于自回归语言模型。

让我们拆解一下，看看它是如何工作的。具有前缀目标的语言模型在给定输入文本作为前缀的情况下预测下一个词。例如，给定序列 “一个男人站在墙角”，我们可以使用” 一个男人站在” 作为前缀并训练模型以预测下一个词: 可以是 “墙角” 或另一个合理的补全词。

Visual transformers (ViT) 通过将每个图像划分为多个块 (patch) 并将这些块按顺序输入给模型，从而将相同的前缀概念应用于图像。利用这个想法，SimVLM 实现了这样一种架构，将图像块序列和前缀文本序列串接起来作为最终的前缀，输入给编码器，然后由解码器来预测该文本序列的接续文本。上图描述了该思想。SimVLM 模型首先在前缀中没有图像块的文本数据集上进行预训练，然后在对齐的图像文本数据集上进行预训练。这些模型用于图生文 / 图像标题生成和 VQA 任务。

利用统一的多模态架构将视觉信息融合到语言模型 (Language Model，LM) 中，最终生成的模型在图像引导类任务中显示出令人印象深刻的能力。然而，仅使用 PrefixLM 策略的模型在应用领域上可能会受到限制，因为它们主要为图像标题生成或视觉问答这两个下游任务而设计。例如，给定一组包含人的图像，我们通过图像的描述来查询符合描述的图像 (例如，“一群人站在一起微笑着站在建筑物前”) 或使用以下视觉推理问题来查询: “有多少人穿着红色 T 恤？ ” 图像。另一方面，学习多模态表示或采用混合方法的模型可以适用于各种其他下游任务，例如目标检测和图像分割。

冻结 PrefixLM



冻结 PrefixLM 预训练策略 (图片出处)

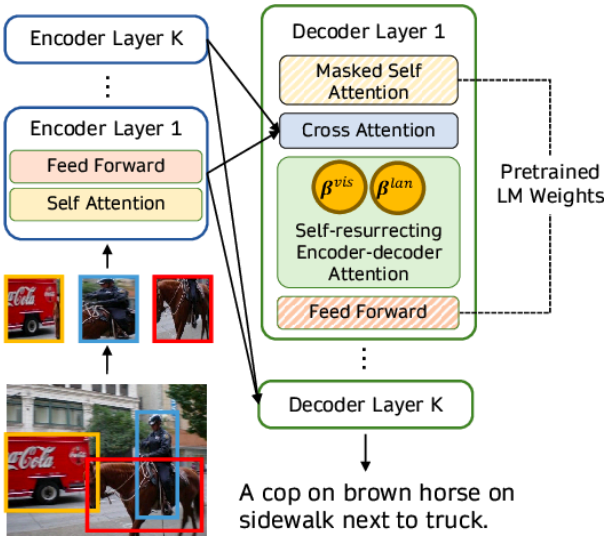
虽然将视觉信息融合到语言模型中非常有效，但能够使用预训练语言模型 (LM) 而无需微调会更有效。因此，视觉语言模型的另一个预训练目标是学习与冻结语言模型对齐的图像嵌入。

Frozen、MAPL 和 ClipCap 使用了冻结 PrefixLM 预训练目标。它们在训练时仅更新图像编码器的参数以生成图像嵌入，这些图像嵌入可以用作预训练的冻结语言模型的前缀，其方式与上面讨论的 PrefixLM 目标类似。Frozen 和 ClipCap 都在对齐的图像文本 (标题) 数据集上进行训练，目的是在给定图像嵌入和前缀文本的情况下生成标题中的下一个词。

最后，Flamingo 索性把预训练视觉编码器和语言模型都冻结了，并在一系列广泛的开放式视觉和语言任务上刷新了少样本学习的最高水平。Flamingo 通过在预训练的冻结视觉模型之上添加一个感知器重采样器 (Perceiver Resampler) 模块并在冻结的预训练 LM 层之间插入新的交叉注意层以根据视觉数据调节 LM 来达到这个性能。

冻结 PrefixLM 预训练目标的一个很好的优势是它可以使用有限的对齐图像文本数据进行训练，这对于那些没有对齐多模态数据集的领域特别有用。

3) 多模态融合与交叉注意力



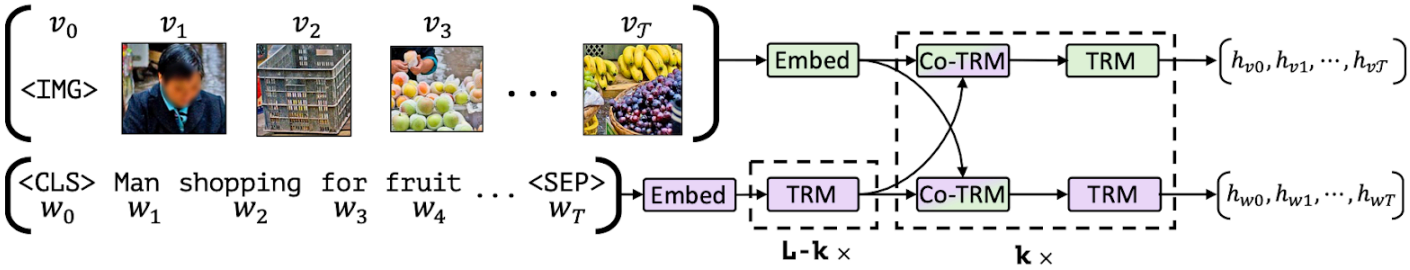
使用交叉注意力机制将视觉信息直接融合到语言模型中 (图片出处)

将预训练语言模型用于多模态任务的另一种方法是使用交叉注意机制将视觉信息直接融合到语言模型解码器的层中，而不是使用图像作为语言模型的附加前缀。VisualGPT、VC-GPT 和 Flamingo 使用此预训练策略并在图像标题任务和视觉问答任务上进行训练。此类模型的主要目标是在把视觉信息融入文本生成能力时在这两者间取得高效的平衡，这在没有大型多模态数据集的情况下非常重要。

VisualGPT 等模型使用视觉编码器来生成图像嵌入，并将视觉嵌入提供给预训练语言解码器模块的交叉注意层，以生成合理的标题。最近的一项工作 FIBER 将具有门控机制的交叉注意力层插入到视觉和语言的主干模型中，以实现更高效的多模态融合，并使能各种其他下游任务，如图文互搜、开放域 (open-vocabulary) 目标检测等。

4) 掩膜语言建模及图文匹配

另一派视觉语言模型把掩码语言建模 (MLM) 和图文匹配 (ITM) 目标组合起来使用，将图像的特定部分与文本对齐，并使能各种下游任务，例如视觉问答、视觉常识推理、文搜图以及文本引导的目标检测。遵循这种预训练设置的模型包括 VisualBERT、FLAVA、ViLBERT、LXMERT 和 BridgeTower。



将图像与文本按部分相应对齐 (图片出处)

让我们解释一下 MLM 和 ITM 目标。给定一个部分遮盖的标题，MLM 的目标是根据相应的图像预测遮盖的单词。请注意，MLM 目标需要使用带有边界框的标注丰富的多模态数据集，或者使用目标检测模型为部分输入文本生成候选目标区域。

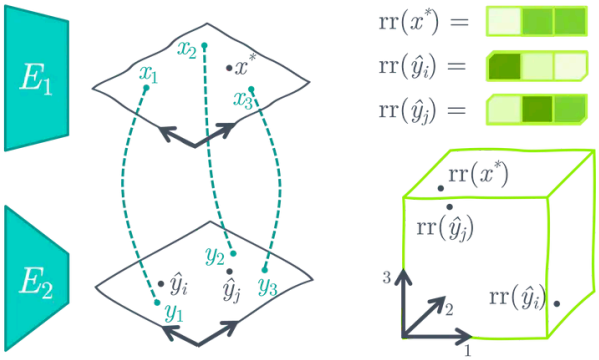
对于 ITM 目标，给定图像和标题对，任务是预测标题是否与图像匹配。负样本通常是从数据集中随机抽取的。MLM 和 ITM 目标通常在多模态模型的预训练期间结合使用。例如，VisualBERT 提出了一种类似 BERT 的架构，它使用预训练的目标检测模型 Faster-RCNN 来检测目标。VisualBERT 在预训练期间结合了 MLM 和 ITM 目标，通过自注意力机制隐式对齐输入文本的元素和相应输入图像中的区域。

另一项工作 FLAVA 由一个图像编码器、一个文本编码器和一个多模态编码器组成，用于融合和对齐图像和文本表示以进行多模态推理，所有这些都基于 transformers。为了实现这一点，FLAVA 使用了多种预训练目标: MLM、ITM，以及 掩膜图像建模 (Masked-Image Modeling，MIM) 和对比学习。

5) 无训练

最后，各种优化策略旨在使用预训练的图像和文本模型来桥接图像和文本表示，或者使预训练的多模态模型能够在无需额外训练的情况下适应新的下游任务。

例如，MaGiC 提出通过预训练的自回归语言模型进行迭代优化，为输入图像生成标题。为此，MaGiC 使用生成的词的 CLIP 嵌入和输入图像的 CLIP 嵌入来计算基于 CLIP 的 “魔法分数 (magic score) ”。



用预训练的冻结的单模态图像和文本编码器创建一个相似性搜索空间 (图片出处)

ASIF 提出了一种简单的方法，可以使用相对较小的多模态数据集将预训练的单模态图像和文本模型转换为多模态模型来用于图像标题生成，无需附加训练。ASIF 背后的关键直觉是相似图像的标题也彼此相似。因此，我们可以通过使用小型数据集里的真实多模态对的来构建一个相对表示空间，然后在该空间执行基于相似性的搜索。

数据集

视觉语言模型通常根据预训练目标在结构各异的大型图像和文本数据集上进行训练。在对它们进行预训练后，再使用特定于任务的数据集进一步针对各种下游任务进行微调。本节概述了一些用于训练和评估视觉语言模型的流行的预训练和下游数据集。

预训练数据集

一般来讲，我们从网上收集大量的多模态数据并将它们组织成图像 / 视频 - 文本对数据集。这些数据集中的文本数据可以是人工生成的标题、自动生成的标题、图像元数据或简单的目标类别标签。此类大型数据集有 PMD 和 LAION-5B 等。PMD 数据集结合了多个较小的数据集，例如 Flickr30K、COCO 和 Conceptual Captions 数据集。COCO 检测和图像标题 (>330K 图像) 数据集分别由图像实例和其所含目标的文本标签及描述对组成。Conceptual Captions (> 3.3M images) 和 Flickr30K (> 31K images) 数据集中的图像以及它们的对应的用自然语言描述图像的标题都是从网上爬取的。

即使是那些人工生成标题的图像文本数据集 (例如 Flickr30K) 也存在固有的噪声，因为用户并不总是为其图像编写描述性或反应图像内容的标题。为了克服这个问题，LAION-5B 等数据集利用 CLIP 或其他预训练的多模态模型来过滤噪声数据并创建高质量的多模态数据集。此外，一些视觉语言模型，如 ALIGN，提出了进一步的预处理步骤并创建了自己的高质量数据集。还有些视觉语言数据集包含了视频和文本双模态，例如 LSVTD 和 WebVid 数据集，虽然它们规模较小。

下游数据集

预训练视觉语言模型通常还会针对各种下游任务进行训练，例如视觉问答、文本引导目标检测、文本引导图像修复、多模态分类以及各种独立的 NLP 和计算机视觉任务。

针对问答类下游任务进行微调的模型，例如 ViLT 和 GLIP，一般使用 VQA (视觉问答) 、VQA v2、NLVR2、OKVQA、TextVQA、TextCaps 和 VizWiz 数据集。这些数据集的图像通常都配有多个开放式问题和答案。此外，VizWiz 和 TextCaps 等数据集也可用于图像分割和目标定位这些下游任务。其他一些有趣的多模态下游数据集有，用于多模态分类的 Hateful Memes，用于视觉蕴含预测的 SNLI-VE for visual entailment prediction, and Winoground，以及用于视觉语言组合推理的 Winoground。

请注意，视觉语言模型也可用于各种经典的 NLP 和计算机视觉任务，例如文本或图像分类。此时，通常使用单模态数据集如 SST2、ImageNet-1k 来完成此类下游任务。此外，COCO 和 Conceptual Captions 等数据集也常用于预训练模型以及标题生成等下游任务。

在 🤗 Transformers 中支持视觉语言模型

使用 Hugging Face Transformers，你可以轻松下载、运行和微调各种预训练视觉语言模型，或者混合搭配预训练视觉模型和预训练语言模型来搭建你自己的模型。🤗 Transformers 支持的一些视觉语言模型有:

- CLIP
- FLAVA
- GIT
- BridgeTower
- GroupViT
- BLIP
- OWL-ViT
- CLIPSeg
- X-CLIP
- VisualBERT
- ViLT
- LiT (an instance of the `VisionTextDualEncoder`)
- TrOCR (an instance of the `VisionEncoderDecoderModel`)
- `VisionTextDualEncoder`
- `VisionEncoderDecoderModel`

这里 CLIP、FLAVA、BridgeTower、BLIP、LiT 和 `VisionEncoderDecoder` 等模型会生成联合图像 - 文本嵌入，可用之于零样本图像分类等下游任务，而其他模型则针对有趣的下游任务进行训练。此外，FLAVA 是基于单模态和多模态两个预训练目标训练的，因此可用于单模态视觉或语言任务以及多模态任务。

例如，OWL-ViT 使能 了零样本 - 文本引导目标检测和单样本 - 图像引导目标检测任务，CLIPSeg 和 GroupViT 使能 了文本和图像引导的图像分割任务，VisualBERT、GIT 和 ViLT 使能 了视觉问答以及其他各种任务。X-CLIP 是一种使用视频和文本模态进行训练的多模态模型，它能够 使能 类似于 CLIP 的零样本图像分类的视频分类任务。

与其他模型不同，`VisionEncoderDecoderModel` 是一个标准化的模型，可用于初始化任意图像转文本模型，这类模型可以使用任何预训练的基于 Transformer 的视觉模型作为编码器 (例如 ViT、BEiT、DeiT、Swin) 以及任何预训练的语言模型作为解码器 (例如 RoBERTa、GPT2、BERT、DistilBERT)。事实上，TrOCR 是这个标准类的一个实例。

让我们继续试验其中的一些模型。我们将使用 ViLT 进行视觉问答，使用 CLIPSeg 进行零样本图像分割。首先，我们要安装 🤗 Transformers: pip install transformers。

基于 ViLT 的 VQA

让我们从 ViLT 开始，下载一个在 VQA 数据集上预训练的模型。我们可以简单地初始化相应的模型类然后调用 `from_pretrained()` 方法来下载想要的 checkpoint。

```
from transformers import ViltProcessor, ViltForQuestionAnswering

model = ViltForQuestionAnswering.from_pretrained("dandelin/vilt-b32-finetuned-vqa")
```

接下来，我们随便下载一张有两只猫的图像，并对该图像和我们的查询问题进行预处理，将它们转换为模型期望的输入格式。为此，我们可以方便地使用相应的预处理器类 (`ViltProcessor`) 并使用相应 checkpoint 的预处理配置对其进行初始化。

```
import requests
from PIL import Image

processor = ViltProcessor.from_pretrained("dandelin/vilt-b32-finetuned-vqa")

# download an input image
url = "http://images.cocodataset.org/val2017/000000039769.jpg"
image = Image.open(requests.get(url, stream=True).raw)
text = "How many cats are there?"

# prepare inputs
inputs = processor(image, text, return_tensors="pt")
```

最后，我们可以使用预处理后的图像和问题作为输入进行推理，并打印出预测答案。但是，要牢记的重要一点是确保你的文本输入与训练时所用的问题模板相似。你可以参考 论文和数据集 来了解如何生成这些问题。

```
import torch

# forward pass
with torch.no_grad():
    outputs = model(**inputs)

logits = outputs.logits
idx = logits.argmax(-1).item()
print("Predicted answer:", model.config.id2label[idx])
```

直截了当，对吧？让我们用 CLIPSeg 做另一个演示，看看我们如何用几行代码执行零样本图像分割。

使用 CLIPSeg 做零样本图像分割

我们将从初始化 `CLIPSegForImageSegmentation` 及其相应的预处理类开始，并加载我们的预训练模型。

```
from transformers import CLIPSegProcessor, CLIPSegForImageSegmentation

processor = CLIPSegProcessor.from_pretrained("CIDAS/clipseg-rd64-refined")
model = CLIPSegForImageSegmentation.from_pretrained("CIDAS/clipseg-rd64-refined")
```

接下来，我们将使用相同的输入图像，并用描述待分割目标的文本来查询模型。与其他预处理器类似，`CLIPSegProcessor` 将输入转换为模型期望的格式。由于我们要分割多个目标，我们分别对每个描述文本都使用相同的输入图像。

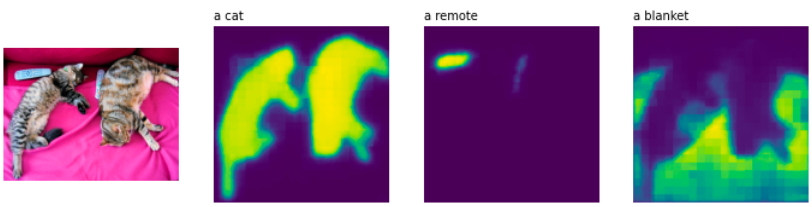
```
from PIL import Image
import requests

url = "http://images.cocodataset.org/val2017/000000039769.jpg"
image = Image.open(requests.get(url, stream=True).raw)
texts = ["a cat", "a remote", "a blanket"]

inputs = processor(text=texts, images=[image] * len(texts), padding=True, return_tensors="pt")
```

与 ViLT 类似，重要的是要参考 原作，看看他们用什么样的文本提示来训练模型，以便在推理时获得最佳性能。虽然 CLIPSeg 在简单的对象描述 (例如 “汽车”) 上进行训练的，但其 CLIP 主干是在设计好的文本模板 (例如 “汽车图像”、“汽车照片”) 上预训练的，并在随后的训练中冻结。输入经过预处理后，我们可以执行推理以获得每个文本查询的二值分割图。

```
import torch

with torch.no_grad():
    outputs = model(**inputs)

logits = outputs.logits
print(logits.shape)
>>> torch.Size([3, 352, 352])
```

让我们可视化一下结果，看看 CLIPSeg 的表现如何 (代码改编自 这篇文章)。

```
import matplotlib.pyplot as plt

logits = logits.unsqueeze(1)

_, ax = plt.subplots(1, len(texts) + 1, figsize=(3*(len(texts) + 1), 12))
[a.axis('off') for a in ax.flatten()]
ax[0].imshow(image)
[ax[i+1].imshow(torch.sigmoid(logits[i][0])) for i in range(len(texts))];
[ax[i+1].text(0, -15, prompt) for i, prompt in enumerate(texts)]
```



太棒了，不是吗？

视觉语言模型支持大量有用且有趣的用例，并不仅限于 VQA 和零样本分割。我们鼓励你尝试将本节中提到的模型用于不同的应用。有关示例代码，请参阅模型的相应文档。

新兴研究领域

伴随着视觉语言模型的巨大进步，我们看到了新的下游任务和应用领域的出现，例如医学和机器人技术。例如，视觉语言模型越来越多地被用于医疗，产生了诸如 Clinical-BERT 之类的工作来根据放射照片来进行医学诊断和报告生成，以及 MedFuseNet 来用于医学领域的视觉问答。

我们还看到大量将联合视觉语言表示应用于各种领域的工作，如用于图像处理 (例如，StyleCLIP、StyleMC，DiffusionCLIP)、基于文本的视频检索 (例如，X-CLIP) 、基于文本的操作 (例如，Text2Live 以及 基于文本的 3D 形状和纹理操作 (例如，AvatarCLIP，CLIP-NeRF, Latent3D, CLIPFace, Text2Mesh)。在类似的工作中，MVT 提出了一种联合 3D 场景 - 文本表示模型，可用于各种下游任务，例如 3D 场景补全。

虽然机器人研究尚未大规模利用视觉语言模型，但我们看到 CLIPort 等工作利用联合视觉语言表示进行端到端模仿学习，并宣称比之前的 SOTA 有了很大的改进。我们还看到，大型语言模型越来越多地被用于机器人任务，例如常识推理、导航和任务规划。例如，ProgPrompt 提出了一个使用大语言模型 (Large Language Model，LLM) 生成情境机器人任务计划的框架。同样，SayCan 使用 LLM 根据给定的环境及环境中物体的视觉描述，选择最合理的动作。尽管这些进展令人印象深刻，但由于目标检测数据集的限制，机器人研究仍然局限在有限的环境和目标集中。随着 OWL-ViT 和 GLIP 等开放集目标检测模型的出现，我们可以期待多模态模型与机器人导航、推理、操作和任务规划框架的集成会更紧密。

结论

近年来，多模态模型取得了令人难以置信的进步，视觉语言模型在性能、用例以及应用的多样性方面取得了显著的飞跃。在这篇文章中，我们讨论了视觉语言模型的最新进展，可用的多模态数据集以及我们可以使用哪些预训练策略来训练和微调此类模型。我们还展示了如何将这些模型集成到 🤗 Transformers 中，以及如何使用它们通过几行代码来执行各种任务。

我们将继续集成最具影响力的计算机视觉和多模态模型，并希望收到你的回音。要了解多模态研究的最新消息，欢迎在 Twitter 上关注我们: @adirik, @NielsRogge, @apsdehal, @a_e_roberts, @RisingSayak, 和 @huggingface.

致谢: 我们感谢 Amanpreet Singh 和 Amy Roberts 的严格审查。此外，还要感谢 Niels Rogge、Younes Belkada 和 Suraj Patil，以及 Hugging Face 的许多其他人，他们为促进基于 Transformers 的多模态模型的使用奠定了基础。