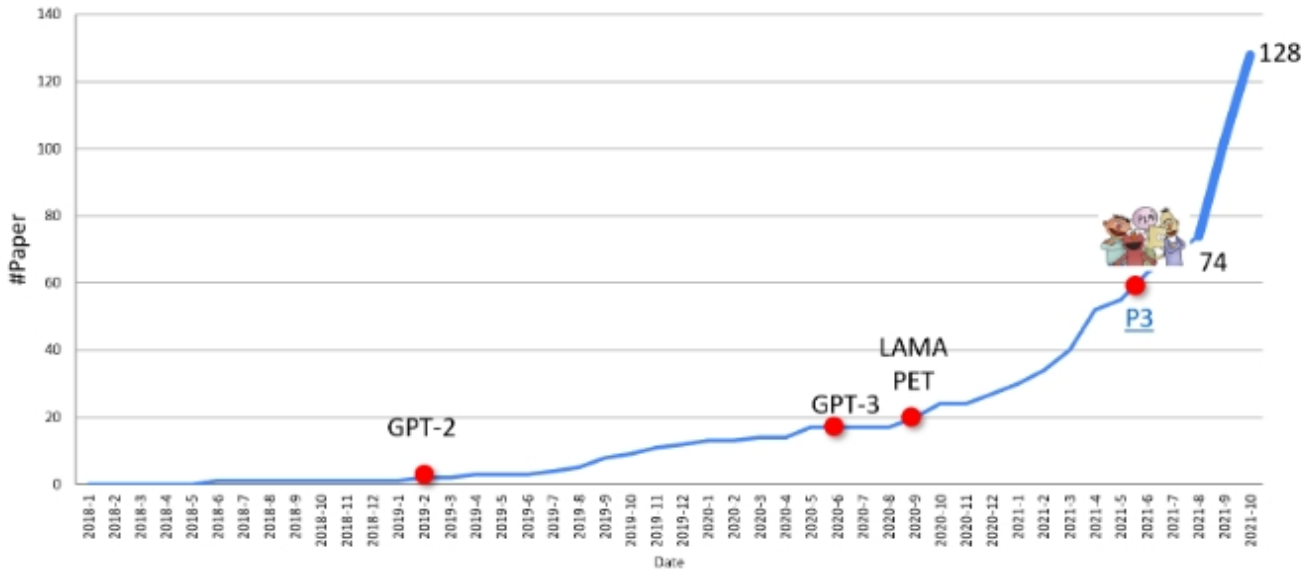
简介

prompt在自2021年大模型的兴起之后，逐渐得到了广泛的关注和研究.Pre-train, Prompt, and Predict在刘鹏飞的关于NLP prompt综述[^1]内甚至被成为NLP的第四范式。随着大模型的不断进展，针对大模型的prompt的研究也在不断地深入。



prompt类似于fine-tune，他们的共同目标都是使得预训练模型和下游任务（在CV方向就是目标检测、图像生成等）尽可能地相近。回忆起在fine-tune预训练模型地做法，我们往往在一个已经在ImageNet-1k上的得到验证的模型backbone，如果想要使用这个模型来做目标检测，我们需要在backbone上添加一个RPN网络，然后再添加一个ROIHead，最后再添加一个分类器，然后对这个backbone做一些`loss func`、`optimizer`的修改，引入下游任务的一些先验的信息（如目标检测就引入一些检测框的位置信息）。在进行fine-tune时候，一般不会对输入数据做任何的修改，我们只希望backbone模型通过训练，能够蕴含更多的下游任务的信息。

而prompt则刚好相反，关于prompt的工作则大部分集中于对于输入数据的修改，prompt通过在数据的输入和输出进行修改，使得下游任务的输入和输出更加靠近backbone，而不是使得backbone具备更多下游信息的知识。因此，使用prompt有一个必要的前提条件，使用prompt的模型必须已经具备了十分丰富的相关知识，使得我们能够接受修改下游任务的输入和输出来匹配模型。 借用综述[^1]对于prompt的定义： 1. 我们需要定义一个函数 f_{prompt}，将下游任务数据输入x改造为backbone的输入x',也被称为Prompt Engineering。2. 选取合适的模型backbone,并将x'输入backbone得到输出y,。3. 定义一个函数g_{prompt}，将backbone的输出y改造为下游任务的输出y',也被称为Answer Engineering。

针对prompt的研究主要集中在一下几个方向：

- 选取模型，选择合适的模型进行prompt，例如GPT、BERT、UniLM等模型，这些模型都具备很大的参数量，并且已经在海量的数据集上充分进行了训练

- 如何定义f_{prompt}和g_{prompt}，这两个函数的定义直接决定了prompt的效果，也是prompt的核心。目前大部分的研究工作都在寻找一个合适的f_{prompt}和g_{prompt}.这个方向的研究具有着一下趋势：从hand-craft到automatic，从single-prompt到multi-prompt.

具体来说，针对多模态的prompt具有以下的研究方向：

1. 预训练模型的选择。
    - 判别式的预训练模型CLIP
    - 生成式(MLM based)的预训练模型
2. Prompt Engineering，即prompt的设计
    - Hand-crafted.针对每一个特定下游任务，手动设计一个template。
    - Discrete.类似于Embedding的形式，将输入映射到一个离散的空间中，事实上在NLP领域，Discrete方法表现不如Continuous和hand-crafted,因此在多模态领域这方面的论文也比较少。但是离散的prompt具有更强的可解释性，但相比于连续的prompt更难优化
    - Continuous.将输入映射到一个连续的空间中，这种方法的优势它既具备hand-crafted的性能优势，也不必为每一个下游任务设计一个template，但是需要设计一个泛化能力很强的prompt很难。
3. Answer Engineering，即将下游任务的输出重构成合适的形式
    - 多模态的输出都可以很好的转化为文本形式text label，因此关于Answer Engineering的工作也比较少，CPT[^2]这篇论文是一个特例
4. Multi-prompt：如何设计多个 prompt 获得更好的效果
    - 目前这方向多模态领域的工作比较少。
5. prompt 范式下的训练策略
    - 模型是否fine-tine,事实上，大部分prompt工作都是不具备训练训练CLIP、GPT的算力和数据的，因此即便要修改预训练模型，也只会修改其中一部分，例如Image-Encoder
    - prompt是否具备额外参数。
    - 是否添加额外的网络架构。

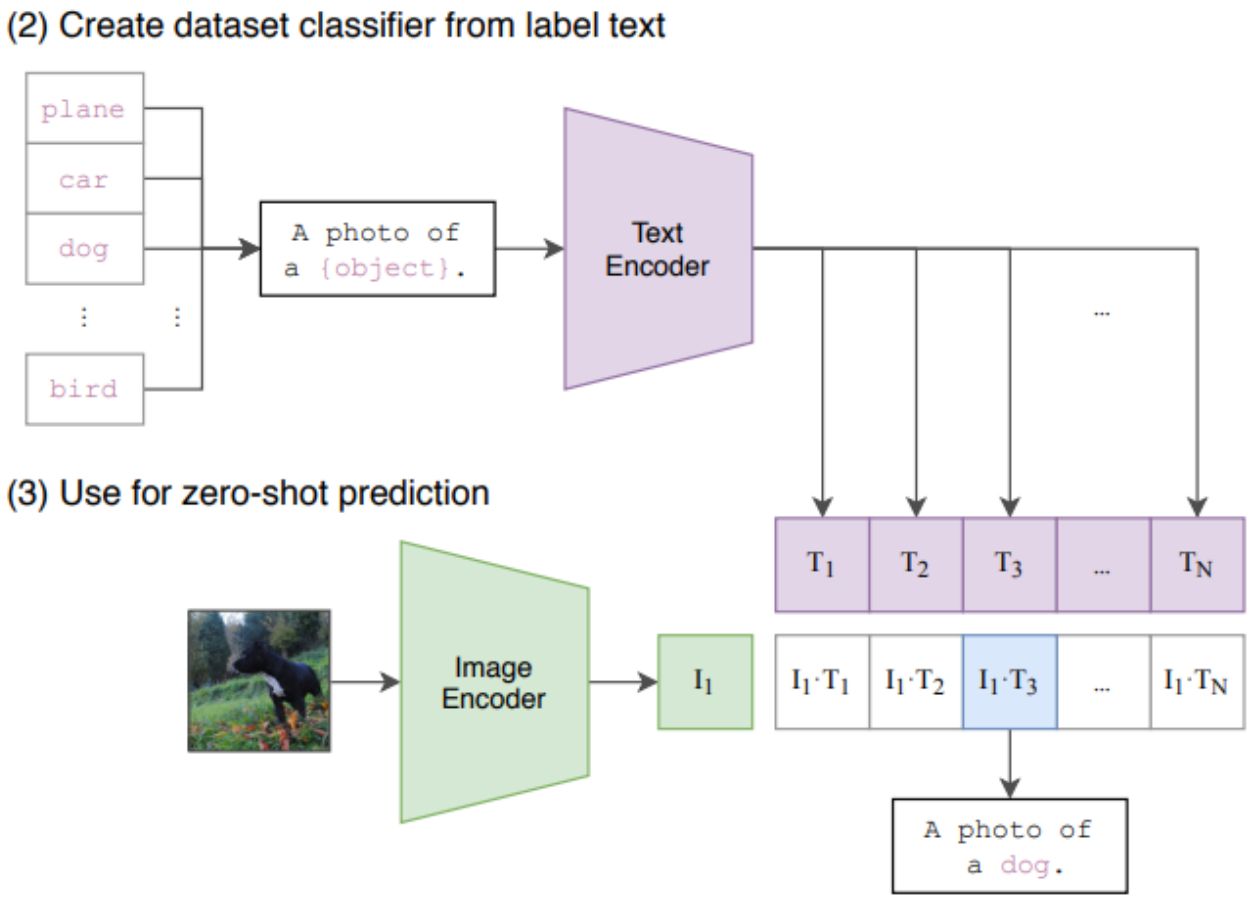
目前prompt的工作主要集中于NLP方面，特别是GPT一类的语言大模型（LLM）。针对多模态领域的prompt工作则相对较少，并且大部分工作尝试将NLP领域的一些trick迁移到多模态中来。并且多模态的prompt相关的论文还没有一个系统的综述，因此我大致按照时间顺序来对多模态的prompt进行介绍。

基于CLIP的prompt

hand-crafted prompt

CLIP: Learning Transferable Visual Models From Natural Language Supervision

clip[^3]在提出的时候就利用了简单的prompt技巧，在通过图像、文本的对比训练得到一个Image-Encoder之后，如何在Image-1k上测试它的zero-shot性能，作者发现相比于使用Image-Encoder得到的特征，使用一个全连接层做分类的办法得到的性能，不如将`A photo of X`（`X`填入各种类别）输入text-Encoder，然后使用和训练类似的方法得到相似度，最终选取相似度最高的类别作为分类输出。CLIP在将通过对比训练得到的预训练的模型用于图像分类的下游任务的方法就是一个hand-crafted的prompt。

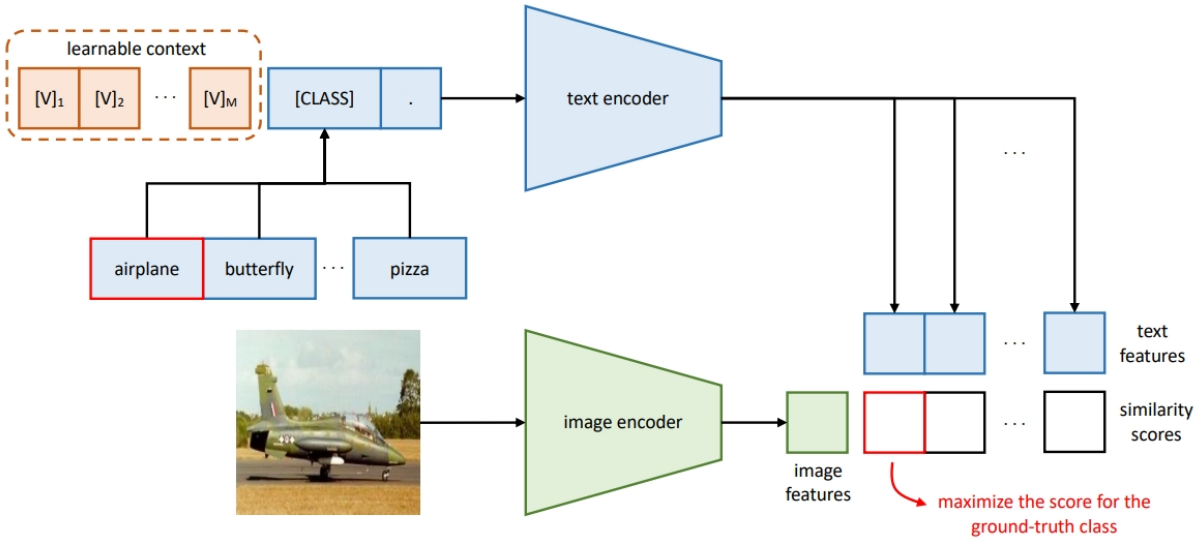


如果采用传统的pretrain-finetune的范式，我们如果想要将clip模型应用于zero-shot的分类任务，应该在Image—Encoder之后添加一个全连接层，固定Image-Encoder，使用一些下游数据集对FC层进行训练，然后用于分类任务。

CLIP则采用了prompt范式，尽管是一种十分简单的`A photo of X`的形式，但是它的效果却比传统的pretrain-finetune的范式要好，这也说明了prompt的有效性。

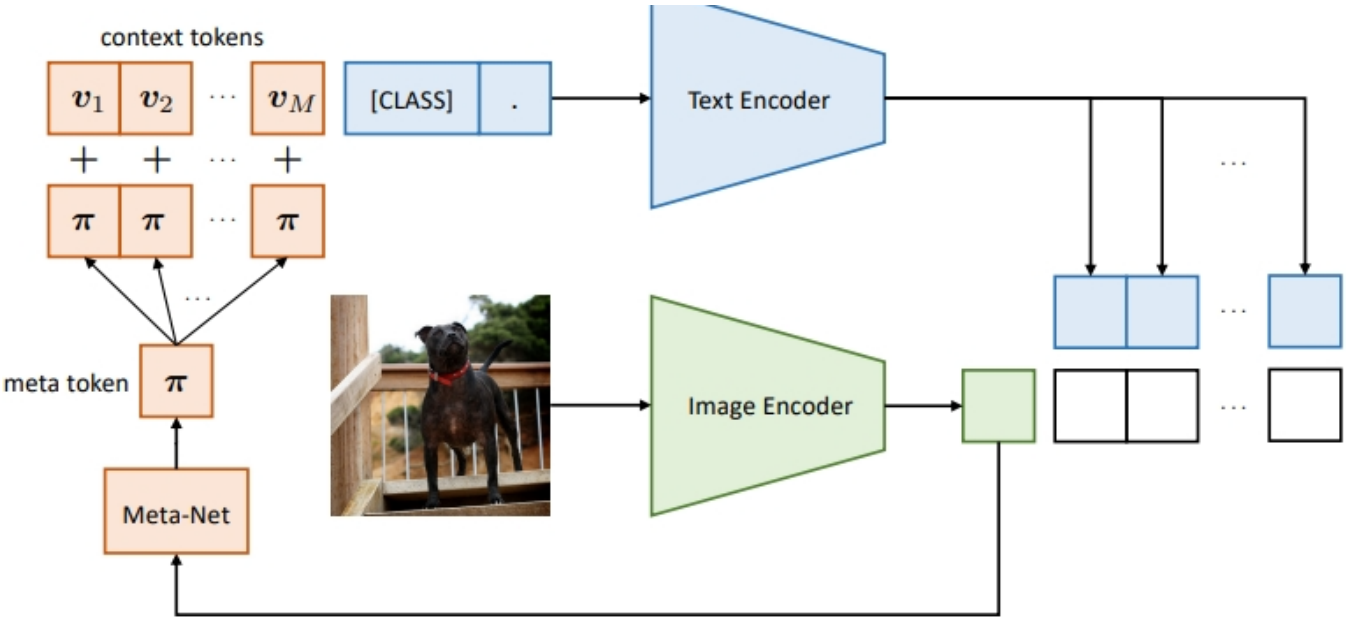
Continuous prompt

Classfication: CoOp[^4]的作者在对CLIP的prompt进行实验的时候发现，对于同一张图片的基本一致的自然语言描述会在最后得到不同的结果。例如一张猫的图片，可以有多张不同的prompt,`a photo of cat`, `a cat`,`a class of cat`。CoOp提出了一种Context Optimization的方法，将NLP的continuous prompt的方法引入到多模态的prompt中来，摒弃了为每个下游任务设计template的方法。CoOp也是第一篇将NLP的prompt方法引入到多模态的prompt中来的论文。

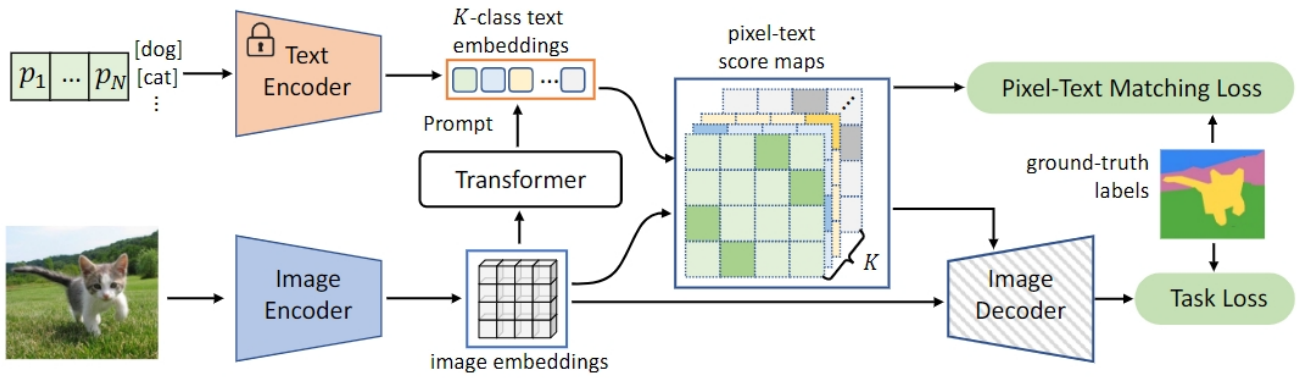


这篇论文还尝试了其他的嵌入class的方式，例如class嵌入到learnable context之中，或者是为每一个class设计单独的learnable context。最终得出结论，对于细粒度的分类任务，单独设计context的效果更好，而对于粗粒度的分类任务，所有class 共享context的效果更好。

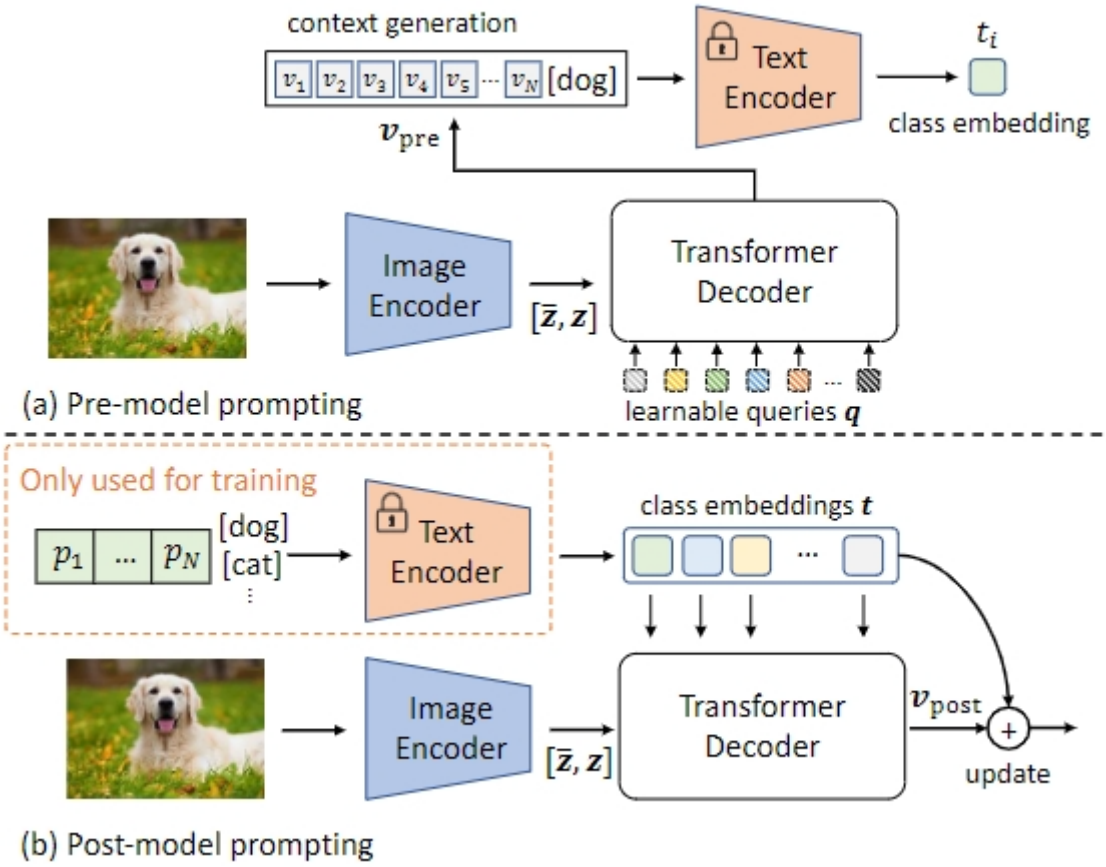
CoCoOp[^6]是该团队在CoOp的基础上提出的一种conditional prompt的方法，CoOp的可学习提示向量泛化能力不强，作者认为CoOp在基类上产生了过拟合。CoCoOp希望解决可学习向量在陌生类别上泛化能力弱的问题。cocoop论文额外添加了一个meta-net，将image-encoder产生的视觉特征通过meta-net融合（直接相加）到learning-embedding中。图中的meta-net其实就是两层全连接层映射。添加meta-net之后，CoCoOp的在陌生类别上的效果比CoOp更好一些，但是基类上的效果比CoOp要差一些。



Dense Prediction: DenseCLIP[^5]将prompt的方法用于pixel-level的密集预测任务上。CLIP是使用contrastive learning的方法进行训练的，并且之前的CoOp是task-level,不能直接应用到pixel level的任务上。DenseCLIP通过finetinue一个Image-Decoder，并将视觉的特征经过一个transformer注入到文本特征中，并将视觉特征和文本特征进行比对，得到一个pixel-text score maps。后续将这个score map输入到Image-Decoder做下游pixel-level的任务，并使用对应的loss损失来进行训练。



DenseCLIP还提出了一个pre-prompt 的方式，直接将视觉特征融合到prompt中，然后通过text-encoder生成class-embedding.



unified Prompt:

参考文献

[^1]::《Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing》

[^2]:CPT: Colorful Prompt Tuning for Pre-trained Vision-Language Models

[^3]:CLIP: Learning Transferable Visual Models From Natural Language Supervision.

[^4]:CoOp: Learning to Prompt for vision-language models CVPR 2021.9

[^5]:DenseCLIP: Language-Guided Dense Prediction with Context-Aware Prompting (2021.12) [^6]:CoCoOp: Conditional Prompt Learning for Vision-Language Models CVPR 2022