

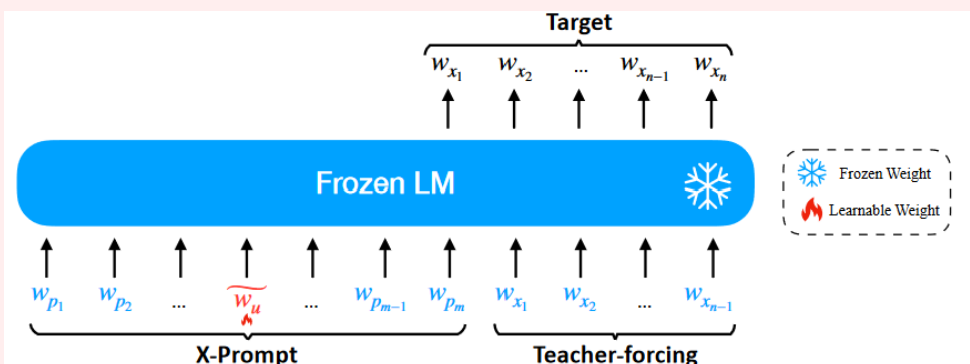
ICL+Prompt

利用ICL优化Prompt

Prompt Tuning/Prompt Engineering

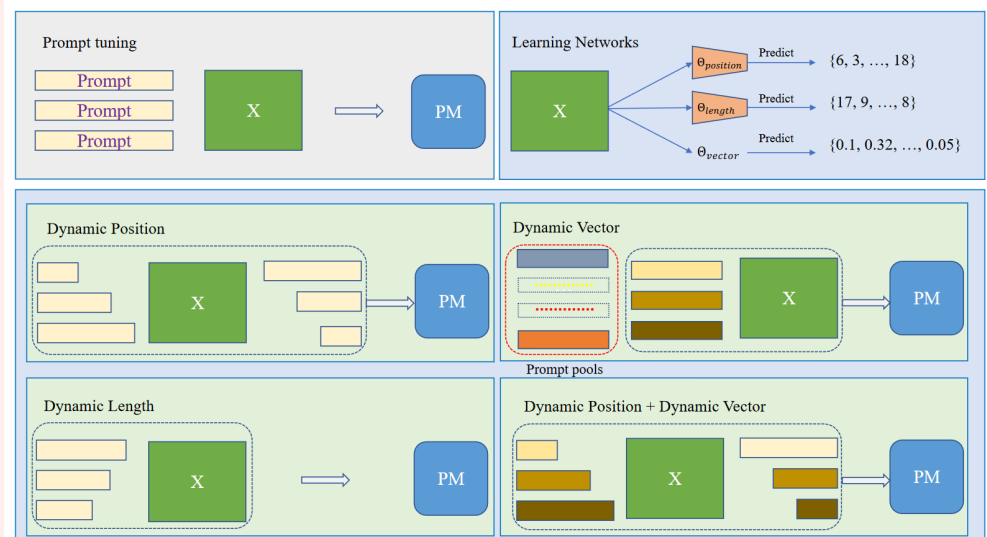
Paper: Extensible Prompts for Language Models on Zero-shot Language Style Customization

X-Prompt主要的思想是将imaginary words注入到natural language中构成prompt, 其中只有imaginary words是learnable. 文章还提出了Context-Augmented Learning(CAL)的概念帮助更好地学习imaginary words(增强在OOD上的表现)。采用的task类型为open-ended text generation和style transfer



Paper: Dynamic Prompting: A Unified Framework for Prompt Tuning

Dynamic Prompting主要思想是根据不同的task/instance的情况来调整相对应prompt的position、length、representation.



Paper: When Do Prompting and Prefix-Tuning work? A Theory of Capabilities and Limitations(粗读)

这篇paper比较理论, 公式也比较多, 所以没怎么看, 这里只记下结论。

- content-based fine-tuning只能激发LLM原有的skill而不能学一个全新的task
- a hierarchy: prompting<soft prompting<prefix-tuning, prefix-tuning的效果最好, prompting和in-context learning是prefix-tuning的特例
- 对于Transformer而言, 变化一个virtual token比变化一个hard token, 会产生更多的completion
- prefix-tuning only adds a bias to the attention block output, 这个bias能激发出LLM原有的skill
- 参数相同的情况下, LoRA可以学会全新的task, prefix-tuning不可以

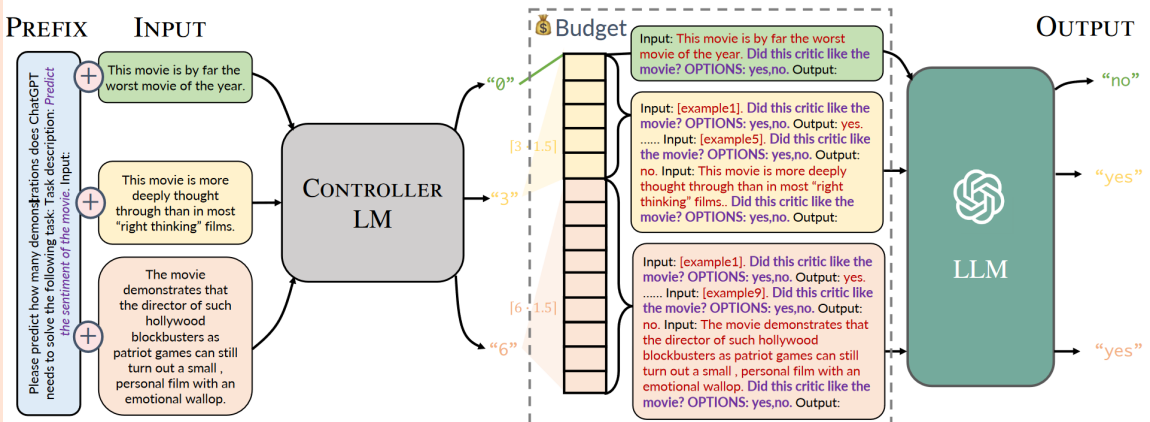
Paper: How Does In-Context Learning Help Prompt Tuning

这篇paper主要是对比了PT(Prompt Tuning)、ICL(In-Context Learning)、IPT(Instruction Prompt Tuning)三种方法的效果来探究ICL对prompt tuning的影响。task类型选用的是language generation task(data-to-text generation, logic-to-text generation、semantic parsing)

得到的一些结论: 1.ICL表现得比PT差 2.PT与IPT表现难分伯仲(取决于task类型/tunable parameter数目等) 3.当demonstration跟test input类似时, IPT可以很好work(别的论文有结论: in-context examples主要是帮助model学习output label space和distribution of input text) 4.IPT在有更多soft prompt tokens时表现得比PT更稳定 5.在有in-context demonstration的情况下, prompt embeddings对于新的task是transferable

Paper: Efficient Prompting via Dynamic In-Context Learning

DYNAICL(Dynamic In-Context Learning)的提出是为了有效解决performance-efficiency trade-off问题, 因为in-context learning使用的demonstration数目会影响prompt长度, 而prompt长度过大会导致低效, 因此动态分配demonstration的数目可以有效解决此问题。论文的核心是train一个meta controller可以动态分配in-context demonstration的数目, train分为两个阶段: 1.使得最后用的generalist model生成"good output"的同时利用最少的demonstrations 2.利用RL来微调



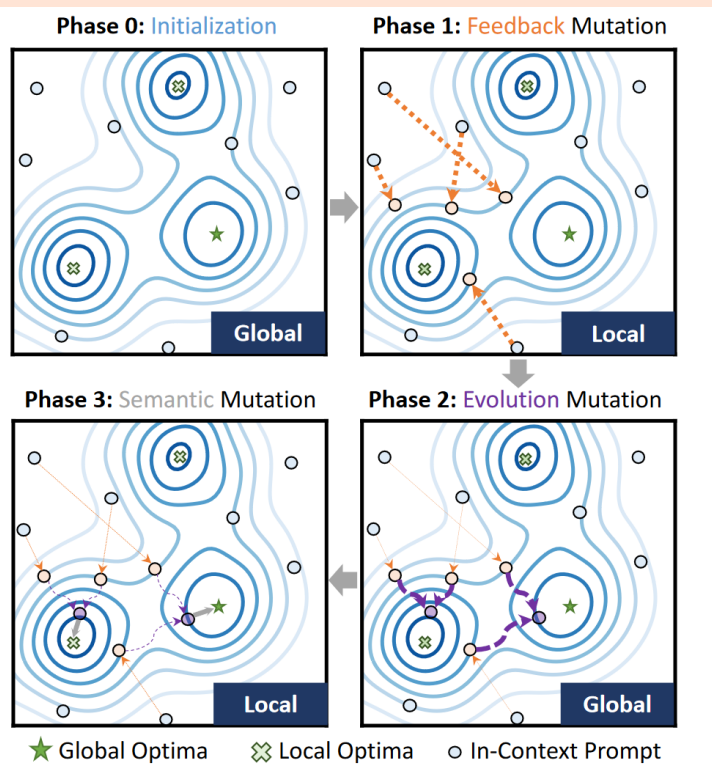
Paper: PhaseEvo: Towards Unified In-Context Prompt Optimization for Large Language Models(粗读)

Submitted on 17 Feb 2024

这篇paper主要提出了一个统一的in-context prompt优化的框架PhaseEvo, 核心算法是进化算法(Evolutionary Algorithms), 采用了Exploration、Exploitation两种不同的优化策略。整个框架分为四个阶段:

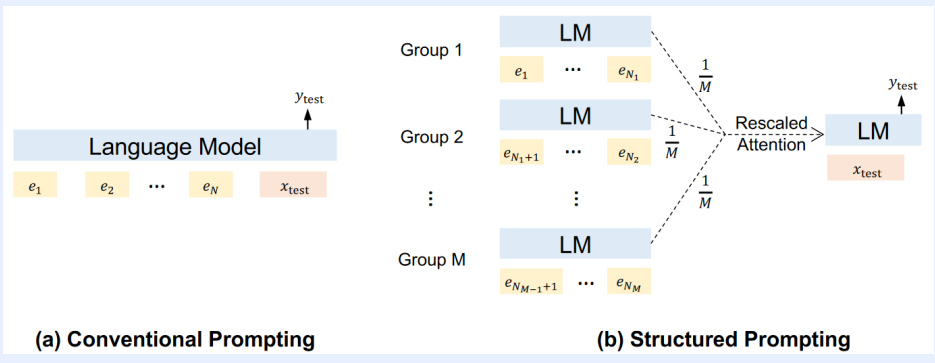
- global initialization
- local feedback mutation
- global evolution mutation
- local semantic mutation

local通常是为了加速收敛/逼近局部最优, global是为了防止stuck in局部最优



Paper: Structured Prompting: Scaling In-Context Learning to 1,000 Examples

该paper主要目的是解决传统Prompting增加demonstrations数目难度比较大的问题, 实现了使用较多数目的demonstrations进行train. Structured Prompting主要思路是把所有demonstrations进行分组, 然后不同group进行独立encode, 之后喂进Rescaled Attention对attention score进行正则化, 最终连同test input喂进LM中。



In Context Learning

Paper: Pre-Training to Learn in Context

该paper提出了Pre-training for In-Context Learning(PICL), 基于Corpus中很多paragraphs都包含intrinsic tasks的假设, 用Retriever将具有相同类型intrinsic task的paragraph检索出来, 并连接在一起构建一个meta-training dataset, 用contrastive learning的方法train一个Encoder使得具有相同类型intrinsic task的paragraph具有类似的Embedding

Paper: Active Example Selection for In-Context Learning(粗读)

这篇paper主要提出了为In-Context Learning挑选example的方法。方法提出的背景是Reorder/Calibration都不能很好解决instability/variance的问题, 基于此, 作者将example selection视为sequential decision问题。整个过程类似于active learning, 作者将active example selection视作MDP, 然后采用Q-learning算法解决

Paper: Large Language Models as Optimizers

这篇文章提出OPRO, 也就是用LLM作为优化器来生成使得task accuracy最高的hard prompt(自然语言形式)。整个工作是通过调用model api实现的, 适用于闭源大模型, 流程图如下图所示:

一个LLM负责优化, 称为optimizer LLM; 另一个LLM负责评估, 称为scorer LLM. optimizer LLM的输出称为an instruction, 用于连接到每个exemplar的question part然后prompt scorer LLM, 可以插入的位置是Q\_begin、Q\_end、A\_begin.

最后文章探究了Meta-prompt design(previous instructions的顺序、instruction score的表示方式、exemplar的数量)、generated instructions的数量、starting point、每步的diversity(LLM temperature设置)等对实际效果的影响。

Future work:

- exemplar的选取要充分利用ICL
- weaker starting point的收敛速度优化(个人感觉这篇paper的意义并不是那么大.....)

LLM as Prompt Optimizers

