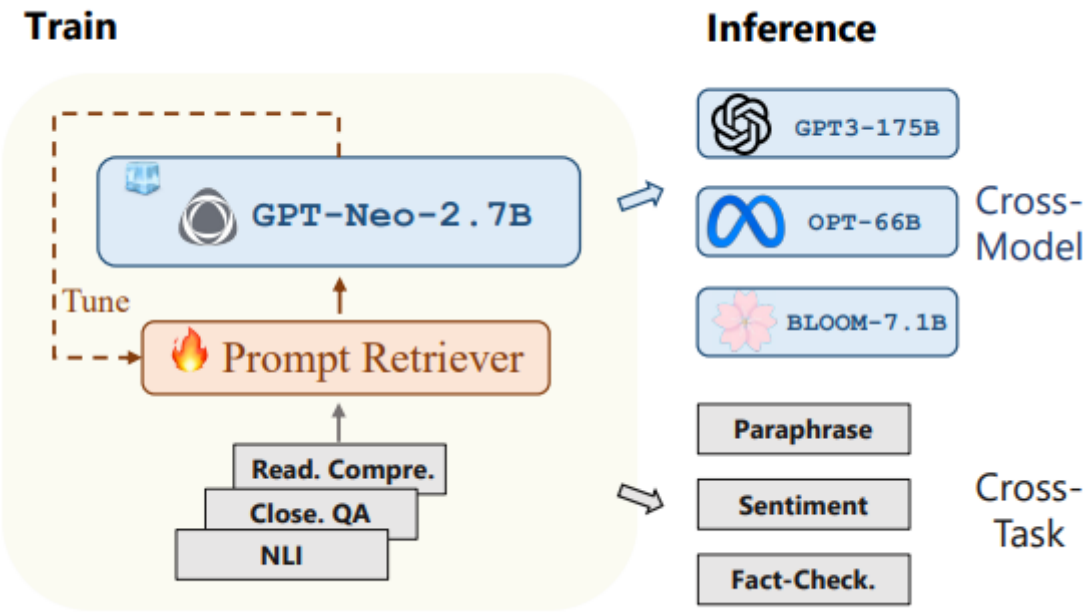
UPRISE: Universal Prompt Retrieval for Improving Zero-Shot Evaluation

工作原理

这篇也是微软研究院发的文章，在23年3月份挂上arxiv，主要探究了训练一个普适的prompt检索器，适用于zero-shot learning，另一篇文章*Learning to Retrieve In-Context Examples for Large Language Models*可以认为是在这篇文章的基础上做出创新，使本篇文章的工作原理可以被使用在few-shot learning（in-context learning）上。

文章的"universal"，指的是两个方面的"普适的"，一是指跨任务，即检索器在一些任务上训练，之后在另一些未见过的任务上检索可用prompt，性能依旧达到要求；二是指跨平台，即检索器在一个较小的模型的"监督"下训练，之后应用在大模型的任务上。具体可见下图：



这里的"监督"，指的是需要一个模型来评断选择的prompt的优劣，给定输入问题与prompt，通过模型的输出结果计算prompt的评分。

问题定义

给定了一个问题输入x，我们需要使用检索器$R(\cdot)$在一个已定义好的prompt池P中检索出一组正面的prompt P^+，即：

$$P^+ = R(x, P),$$

之后将选出的prompt与问题x连接，送至固定参数的大模型中，获得预测输出：

$$y^{P^+} = LLM(y^{P^+} \mid P^+ \oplus x),$$

研究的目标就是通过调整检索器R使用更好的prompt，从而使模型的输出y^{P^+}尽可能地契合正确答案y。

与此同时，由于检索器需要普适性，所以微调好的检索器应该可以直接被使用在其他模型上用于检索未曾见过的任务的prompt，而不需要再行调整。

方法

数据构建

1. 关于任务数据：这篇文章研究的是zero-shot下的prompt检索，所以prompt一般都是任务描述（例如，“根据我提供给你的电影梗概，告诉我这部电影的类型”或”根据人物对话，判断对话是积极还是消极的”等），如下图。本文根据FLAN的方法，将数据集数据转换为自然语言的指导作为prompt，每个数据集使用了七种不同的template，每条数据随机使用其中的一种template，组成prompt。

TESTING CLUSTER: TASK
Reading Comprehension: SQuADv1 (Rajpurkar et al., 2016)
INPUT INSTRUCTION
Here is a question about this article: As of August 2010, Victoria had 1,548 public schools, 489 Catholic schools and 214 independent schools. Just under 540,800 students were enrolled in public schools, and just over 311,800 in private schools. Over 61 per cent of private students attend Catholic schools. More than 462,000 students were enrolled in primary schools and more than 390,000 in secondary schools. Retention rates for the final two years of secondary school were 77 per cent for public school students and 90 per cent for private school students. Victoria has about 63,519 full-time teachers. What is the answer to this question: What percentage of private school students go to Catholic schools?
LABEL COMPLETION
61

PROMPT CLUSTER: TASK
Closed-book QA: Natural Questions (Kwiatkowski et al., 2019)
DEMONSTRATION INPUT
What is the answer to this question? what is the official poverty rate in the us?
DEMONSTRATION ANSWER
In 2015, 13.5%

---

2. 关于promp池：对于每一个数据集群（一个或多个类似的数据集转换成的任务数据），它的prompt池就是其他数据集群，这些数据集群由其他任务的数据集转化组成。这与in-context learning的思路类似，即使测试任务与训练任务的任务类型完全不同，测试输入仍能得益于类似的话题、提问方式、思考步骤等；这里必须使用其他数据集群，原因就是所谓的”普适性“的第一条，跨任务有效性。

promt打分

这里指明了prompt判断优劣的方法。对于训练数据集群（training cluster）中的每条数据(x_i, y_i)，都要从prompt池（prompt pool）中找到一个正向（positive）的prompt和N个负面（Negative）的prompt，感性的评价正向/负面的标准自然是LLM是否能根据该prompt达到更好的预测结果。详细而言，对于**文字补全**与**选择**两大类问题，prompt的好坏有各自的评价标准：

1. 文字补全，这类问题一般是自由输出结果，所以评价标准是

$$score(p_j, x_i) = metric(y_i, y_i^{p_j}),$$

也就是将测试问题的答案和模型根据prompt而做出的输出的答案作比较，这里的metric可以是F1、Exact Match或其他方法。

2. 选择，这类问题需要在给定的可选范围内选择一个或多个选项作为输出，对于可选的范围$\{o_m\}_{m=1}^M$，假设o_{y_i}是正确选项，同时模型输出的M个选项各自的概率是$LH(o_m)$，选最高的作为模型最终输出。但是这里仅使用$acc(y_i, y_i^{p_j})$的方差太大（正确则为1，错误则为0），因而改为使用这个公式作为最终的衡量标准：

$$score(p_j, x_i) = acc(y_i, y_i^{p_j}) \cdot \frac{LH(o_{y_i})}{\sum_{m=1}^{M} LH(o_m)},$$

最后，需要考虑一个问题：如果对于任意一个训练问题，我们都要得到其一个正向prompt和N个负面prompt，那工作量是巨大的，因为得给每个池子中的prompt都打分。所以需要一个过滤机制，减少需要打分的prompt数量。该机制源于一个发现：即使是随机抽取的单一训练演示，也可以改善测试样本的评估性能，因为训练演示和测试样本属于相同的任务，它们可能共享共同的模式或结构。因此随机从prompt池中抽取一个与测试问题相同任务的小子集L作为待检索的prompt池，如果该小子集中所有的prompt得分都为0（没有正向prompt），就重新抽一个小子集，直到抽出来为止。同时，也抽出B个不同任务的prompt和B个相同任务但是得分最低的prompt，组成负面prompt组合。

检索器微调

对于训练数据中的每一条数据，各自得到上节讲的一个正向样本与$2B$个负面样本后，将所有训练数据的90%作为训练用，10%作为验证用。检索器是一个双编码器模型，输入编码器$E_X(\cdot)$以问题输入x_i作为输入，prompt编码器$E_P(\cdot)$以prompt为输入，训练检索器的目标是最大化正向prompt与问题输入的相似度，最小化负面prompt与问题的相似度，对于一条训练数据(x_i, y_i)，其损失函数为：

$$L(x_i, p_i^+, p_{i,1}^-, \ldots, p_{i,2B}^-) = -log\frac{e^{sim(x_i, p_i^+)}}{e^{sim(x_i, p_i^+)} + \sum_{j=1}^{2B} e^{sim(x_i, p_i^-)}},$$

其中$sim()$函数是参数的嵌入内积，计算两者的相似度。

推理阶段

prompt编码器微调好后，将整个prompt池都编码（之前指抽了一些小子集）。推理阶段，对于一个未见过的任务x_{test}，使用输入编码器对其编码然后同样与prompt编码使用内积算相似度，从prompt池中找K个最相似的prompt，最后将K个prompt与测试问题连接起来，生成预测结果并检验预测结果的合理性，检验方法与训练时相同（也就是说，训练阶段与检验阶段都是可以看到正确答案的）。

实验

1. 训练数据集群（training cluster）前文已述是根据任务种类来集群一个或多个数据集的，包括阅读理解、闭卷问答、释义检测、自然语言推理、情感分析常识推理、文本摘要等；
2. 使用一个小模型（GPT-Neo-2.7B）去微调检索器，然后在大模型上检验检索器性能；
3. 前文讲的随机抽取小子集，小子集大小设为50，负样本数量B设为20，两个初始编码器都是$BERT_{BASE}$，正向样本一次没抽到重抽的最大次数为7，已被认为绝对可以抽到。

实验结果

该检索器在阅读理解、释义检测任务效果很好，闭卷问答、自然语言推理任务也有提高，但在常识推理上效果差。（其他工作已有结论，直接以语言建模问题的形式提出的任务使用指令prompt效果不好，应使用思维链更好）

还有证据表明该检索器可以减轻chat-GPT的幻觉问题。

一些检索器的训练细节：

Hyperparameter	Assignment
Computing Infrastructure	8 V100-32GB GPUs
Number of epochs	3
Batch size per GPU	16
Maximum sequence length	256
Maximum learning rate	1e-5
Optimizer	Adam
Adam epsilon	1e-8
Adam beta weights	0.9, 0.999
Learning rate scheduler	warmup linear
Weight decay	0.0
Warmup steps	1000
Learning rate decay	linear