

Learning To Retrieve Prompts for In-Context Learning

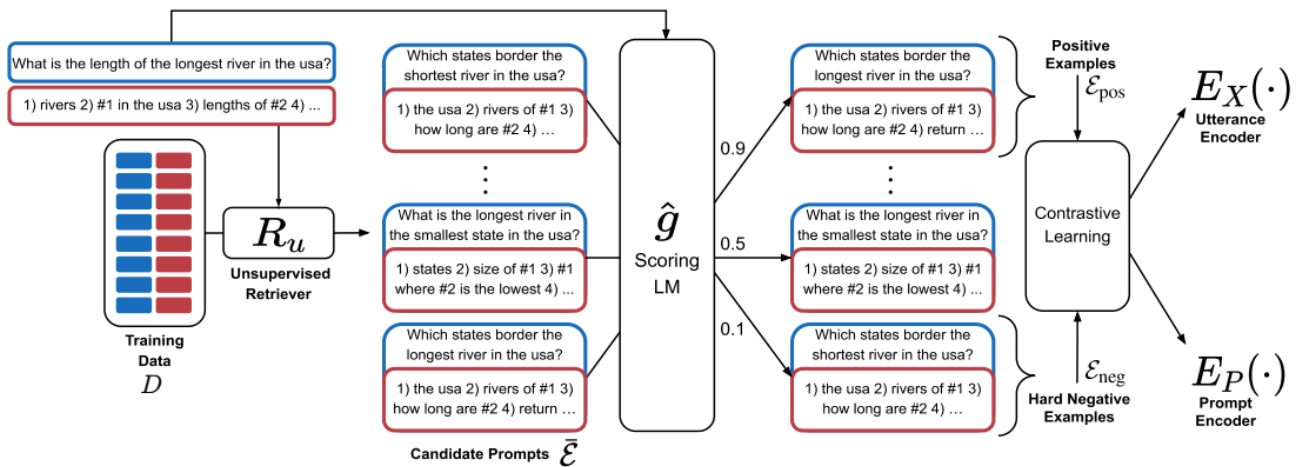
内容介绍

文章已经收录在NAACL2022，是最早提出用一个检索器根据大模型反馈选取数据作为prompt，而非根据数据本身的特性来选取的文章，即“一个示例是好是坏不应该由人来判定，而应该由模型来判定”。在这之前prompt的选取大概分为以下几类：

1. 使用无监督的编码器，对所有训练数据编码，然后为每个测试问题检索最近邻；（没有反馈示例的优劣）
2. 使用有监督的检索器，检索器是根据专门针对知识库查询定制的监督训练的，并依赖于正式查询之间的表面相似性。（有反馈示例的优劣，但并非大模型反馈）
3. 使用大模型反馈示例的优劣，但与训练一个检索器不同，做法是从训练集中随机抽取了大量话语-程序对，并根据GPT-3选择与目标实例问题相似的对。（有大模型的反馈，但直接使用大模型，故而每个测试问题要相应运行数百次GPT-3，而不是使用一个轻量的检索器来做这件事）

在具体实现上，*UPRISE: Universal Prompt Retrieval for Improving Zero-Shot Evaluation*与本篇文章有很多相似之处，因为前者就是在后者基础上的创新（使该方法扩展为可以跨模型、跨任务）。

方法框架：



方法

生成训练数据

对于一条数据 (x, y) ,

第一步先使用一个无监督检索器 $R_u(\cdot)$ 从训练集 D 中初步获得一个高质量候选集 $\bar{\mathcal{E}}$ ，无监督检索器可以是BM25或者SBERT；

第二步对于训练数据 (x, y) 的高质量候选集 $\bar{\mathcal{E}} = \bar{e}_1, \dots, \bar{e}_L$ ，使用语言模型 \hat{g} 为每个 e_i 打分， $s(\bar{e}_i) = \text{Prob}_{\hat{g}}(y | \bar{e}_i, x)$ ；

为所有候选集数据的得分排序，得到最高的k个与最低的k个各自组成 \mathcal{E}_{Pos} 与 \mathcal{E}_{Neg} 。

训练与推理

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，学习方法与目标均与UPRISE: Universal Prompt Retrieval for Improving Zero-Shot Evaluation类似：

$$L(x_i, e_i^+, e_{i,1}^-, \dots, e_{i,2B-1}^-) = -\log \frac{e^{\text{sim}(x_i, e_i^+)}}{e^{\text{sim}(x_i, e_i^+)} + \sum_{j=1}^{2B-1} e^{\text{sim}(x_i, e_{i,j}^-)}},$$

其中有一个本问题对应的正向示例，一个本问题对应的负向示例，同一批次其他问题的B-1个正向示例与B-1个负向示例，B就是批次大小。

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示例编码使用内积算相似度，从示例池中找 L' 个最相似的prompt，最后将 L' 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。

实验

实验的baseline包括随机选取与使用无监督检索器选取（SBERT、BM25），带检索器的baseline与文章所提出的方法区别在于打分函数的区别。

在两个维度评估了提出的检索器：语言模型作为服务（LM-as-aservice），指的是训练检索器的语言模型与用作最后推理的语言模型是同一个模型；语言模型作为代理（LM-as-a-proxy），指的是前后的语言模型不同。

LM-as-aservice的结果：

	Model	BREAK	MTop	SMCALFlow
Unsuper.	RANDOM	1.7	7.3	8.9
	SBERT	21.6	48.7	43.6
	BM25	26.0	52.9	46.1
	BRUTEFORCE	7.7	18.1	11.1
Super.	DR-BM25	23.6	50.2	43.1
	CBR	25.7	57.0	51.4
	EPR (ours)	31.9	64.2	54.3
Oracle	BM25-ORACLE	32.3	58.9	47.3
	LM-ORACLE	43.1	71.6	73.7

LM-as-a-proxy的结果：

	BREAK				MTop				SMCALFlow			
Method	RANDOM	BM25	CBR	EPR	RANDOM	BM25	CBR	EPR	RANDOM	BM25	CBR	EPR
GPT-3	4.2	20.1	21.3	25.3	7.6	52.5	54.8	62.6	5.8	35.3	41.6	46.5
CODEX	8.9	24.5	24.2	29.5	10.8	60.6	59.4	66.1	7.2	45.1	48.7	50.3
GPT-J	3.3	26.7	26.7	31.5	8.8	56.6	58.0	65.4	10.6	50.4	50.9	57.4
GPT-NEO	1.0	22.8	25.8	29.9	7.6	52.8	55.4	63.6	8.0	46.1	50.1	53.5

结论

本文提出的EPR方法，与先前方法最大的不同在于：先前的方法是通过人来评判一个示例是好是坏，而EPR则是通过模型来判断一个示例是好是坏。由于神经网络的解释性是较差的，作者提出的方法可以在一定程度上拉近人类与模型之间的隔阂，进而提升模型的效果