# MIT发现语言模型内的事实知识可被修改？？

原创 小伟 夕小瑶科技说 2022-11-28 12:05 发表于北京

文 | 小伟

## 前言

众所周知，自回归语言模型(如GPT-2)里存储着大量的事实知识，比如语言模型可以正确的预测出埃菲尔铁塔所在的城市是巴黎市。

那么语言模型是在什么地方存储这些知识呢？我们是否可以修改存储在语言模型里的知识呢？

来自于MIT的这篇文章就对这些问题做出了解答。

它发现GPT中的事实知识对应于可以直接编辑的局部计算。通过对GPT的一小部分参数进行小的改变就可以修改其内部的知识，实现我们把埃菲尔铁塔搬到英国的小目标 :)

论文标题:

**Locating and Editing Factual Associations in GPT**

论文链接:

*https://arxiv.org/abs/2202.05262*

## 概览

首先，什么是语言模型里的知识呢？我们可以用三元组 (**s,r,o**) 来代表这些事实知识，其中 **s** 和 **o** 分别是主体和客体，**r** 代表它们之间的关系。例如：

> (s = Kevin Durant, r = plays sport professionally, o = basketball)

就表明了杜兰特是一名职业篮球运动员这一事实。

其次，为什么需要定位以及修改语言模型里的知识呢？显而易见，它可以帮助我们很容易地更新改正语言模型中存在的过时或者错误的知识。例如关于川普已经过时了的知识:

> (s = Donald Trump, r = is President of, o = the US)

可以看到对语言模型里的知识进行定位和修改还是蛮有用的，让语言模型可以与时俱进。

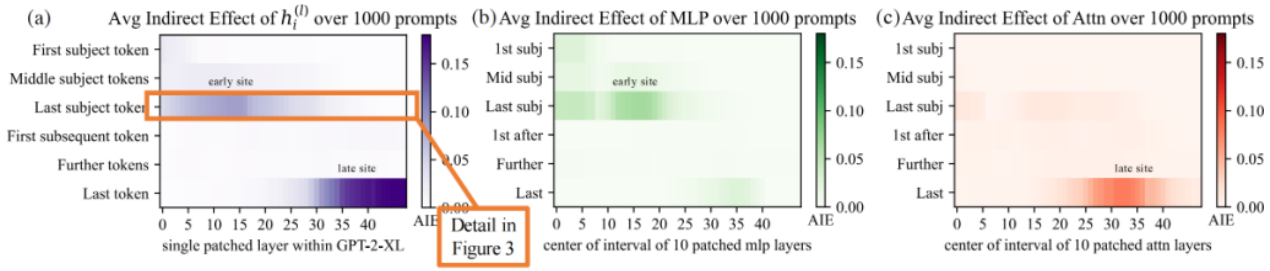
那么本文具体是用什么方法来实现目标的呢？下面让我们一起来一探究竟。

## 定位

为了定位语言模型中的知识，本文采用了因果追踪的方法来量化每个隐藏状态对模型预测的因果影响。为了计算每个隐藏状态对正确的事实知识预测的贡献，本文设计了3种不同的运行模式：

- 干净模式: 将输入正常喂给模型得到输出
- 干扰模式: 给输入的embedding加上高斯分布的噪声来得到被干扰的输出
- 干扰后恢复模式: 给输入的embedding加上高斯分布的噪声,同时调整模型在某一层的某个token index处的状态为对应的干净模式中的状态。直觉上来看，在许多其他状态被干扰的情况下，一些干净状态恢复正确事实的能力将表明它们在计算图中的因果重要性。

通过把干扰模式以及干扰后恢复模式的输出进行对比(在本文中定义为average indirect effect)，我们就可以知道模型的不同组成部分对最终模型预测的因果影响。

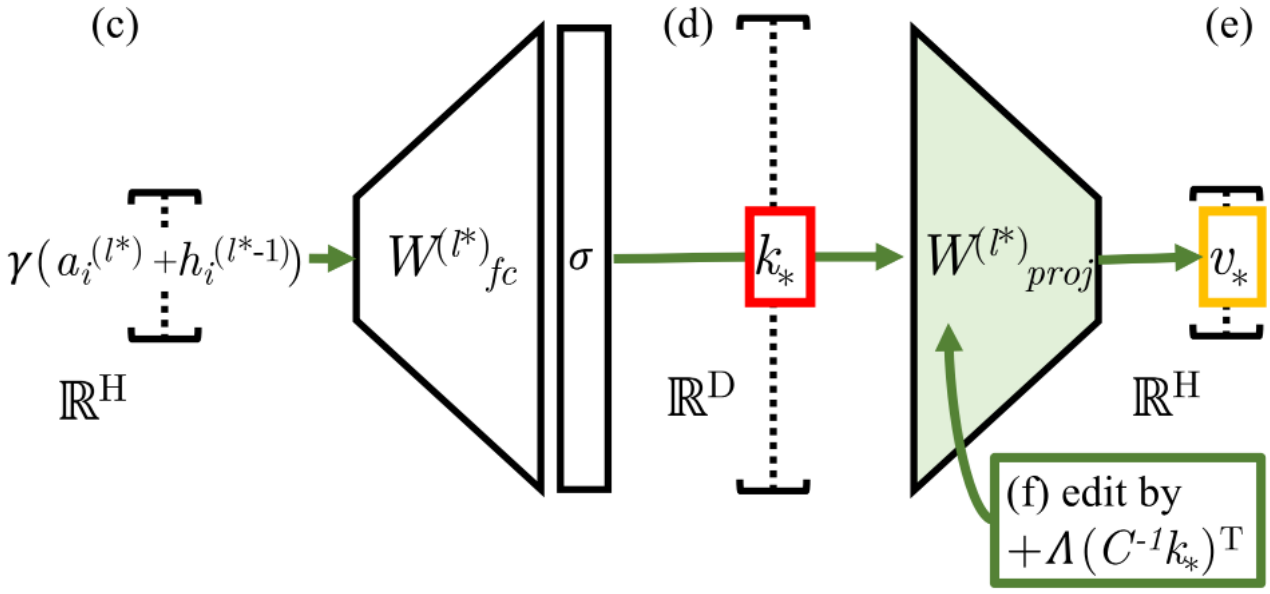



如上图所示，我们可以看到MLP模块在早期起到了决定性的作用 (MLP 6.6% AIE vs. attention 1.6% AIE)，而attention模块则是在最后一个token处比较重要。

基于因果追踪的结论以及过往的工作，作者提出了一种存储事实知识的特定机制: 早期的**MLP**模块进行知识检索，然后后期的注意力机制将累积的信息带到计算结束**(**最后一个**token)**处来预测输出。

## 修改

现在我们已经知道了事实知识主要存储在早期的MLP中，那我们应该怎么样来修改这些知识呢？本文引入了 **Rank-One Model Editing (**一阶模型编辑**)** 来修改模型里的知识。




具体来说，一阶模型编辑(ROME)把MLP视为简单的键值存储:如果键编码主体，值编码和主体相关的知识，MLP就可以通过检索与键对应的值来获取相应的知识。

在本文中，ROME使用针对**MLP**参数的一阶编辑来直接写入新的键值对，也就是为模型注入新的知识。如上图所示，(d) 处的向量$k_*$表示要插入的主体的键，而 (e) 处的输出$v_*$编码了有

关该主体的知识。ROME通过对键值间映射矩阵 $W_{proj}^{l^*}$进行一阶编辑 (f) 来插入新的键值对 $(k_*, v_*)$:

$$\hat{W}_{proj}^{l^*} = W_{proj}^{l^*} + \wedge (C^{-1}k_*)^T$$

其中$W_{proj}^{l^*}$ 是原始映射矩阵，$C = KK^T$是从维基百科文本中估计出的关于$k$的协方差常量，以及 $\wedge = \frac{v_* - W_{proj}^{l^*}k_*}{(C^{-1}k_*)^T k_*}$。

这样通过把 $W_{proj}^{l^*}$ 修改为 $\hat{W}_{proj}^{l^*}$, 我们就实现了在语言模型中插入新的事实知识。

## 实验

本文主要在两个基准上进行了实验，结果非常给力(**ROME**是本文方法)

Table 1: zsRE Editing Results on GPT-2 XL.

| Editor | Efficacy ↑ | Paraphrase ↑ | Specificity ↑ |
|---|---|---|---|
| GPT-2 XL | 22.2 (±0.5) | 21.3 (±0.5) | 24.2 (±0.5) |
| FT | 99.6 (±0.1) | 82.1 (±0.6) | 23.2 (±0.5) |
| FT+L | 92.3 (±0.4) | **47.2 (±0.7)** | 23.4 (±0.5) |
| KE | 65.5 (±0.6) | 61.4 (±0.6) | 24.9 (±0.5) |
| KE-zsRE | 92.4 (±0.3) | 90.0 (±0.3) | 23.8 (±0.5) |
| MEND | 75.9 (±0.5) | 65.3 (±0.6) | 24.1 (±0.5) |
| MEND-zsRE | 99.4 (±0.1) | **99.3 (±0.1)** | 24.1 (±0.5) |
| ROME | **99.8 (±0.0)** | 88.1 (±0.5) | **24.2 (±0.5)** |

Table 4: **Quantitative Editing Results**. 95% confidence intervals are in parentheses. **Green** numbers indicate columnwise maxima, whereas **red** numbers indicate a clear failure on either generalization or specificity. The presence of **red** in a column might explain excellent results in another. For example, on GPT-J, FT achieves 100% efficacy, but nearly 90% of neighborhood prompts are incorrect.

| Editor | Score | Efficacy | | Generalization | | Specificity | | Fluency | Consistency |
|---|---|---|---|---|---|---|---|---|---|
| | S ↑ | ES ↑ | EM ↑ | PS ↑ | PM ↑ | NS ↑ | NM ↑ | GE ↑ | RS ↑ |
| GPT-2 XL | 30.5 | 22.2 (0.9) | -4.8 (0.3) | 24.7 (0.8) | -5.0 (0.3) | 78.1 (0.6) | 5.0 (0.2) | 626.6 (0.3) | 31.9 (0.2) |
| FT | 65.1 | 100.0 (0.0) | 98.8 (0.1) | 87.9 (0.6) | 46.6 (0.8) | **40.4 (0.7)** | **-6.2 (0.4)** | 607.1 (1.1) | 40.5 (0.3) |
| FT+L | 66.9 | 99.1 (0.2) | 91.5 (0.5) | **48.7 (1.0)** | 28.9 (0.8) | 70.3 (0.7) | 3.5 (0.3) | 621.4 (1.0) | 37.4 (0.3) |
| KN | **35.6** | **28.7 (1.0)** | **-3.4 (0.3)** | **28.0 (0.9)** | **-3.3 (0.2)** | 72.9 (0.7) | 3.7 (0.2) | **570.4 (2.3)** | **30.3 (0.3)** |
| KE | 52.2 | 84.3 (0.8) | 33.9 (0.9) | 75.4 (0.8) | 14.6 (0.6) | **30.9 (0.7)** | **-11.0 (0.5)** | **586.6 (2.1)** | 31.2 (0.3) |
| KE-CF | **18.1** | 99.9 (0.1) | 97.0 (0.2) | 95.8 (0.4) | 59.2 (0.8) | **6.9 (0.3)** | **-63.2 (0.7)** | **383.0 (4.1)** | **24.5 (0.4)** |
| MEND | 57.9 | 99.1 (0.2) | 70.9 (0.8) | 65.4 (0.9) | 12.2 (0.6) | **37.9 (0.7)** | **-11.6 (0.5)** | **624.2 (0.4)** | 34.8 (0.3) |
| MEND-CF | **14.9** | **100.0 (0.0)** | **99.2 (0.1)** | **97.0 (0.3)** | **65.6 (0.7)** | **5.5 (0.3)** | **-69.9 (0.6)** | **570.0 (2.1)** | 33.2 (0.3) |
| ROME | **89.2** | 100.0 (0.1) | 97.9 (0.2) | 96.4 (0.3) | 62.7 (0.8) | **75.4 (0.7)** | **4.2 (0.2)** | 621.9 (0.5) | **41.9 (0.3)** |
| GPT-J | 23.6 | 16.3 (1.6) | -7.2 (0.7) | 18.6 (1.5) | -7.4 (0.6) | 83.0 (1.1) | 7.3 (0.5) | 621.8 (0.6) | 29.8 (0.5) |
| FT | **25.5** | **100.0 (0.0)** | **99.9 (0.0)** | 96.6 (0.6) | 71.0 (1.5) | **10.3 (0.8)** | **-50.7 (1.3)** | **387.8 (7.3)** | **24.6 (0.8)** |
| FT+L | 68.7 | 99.6 (0.3) | 95.0 (0.6) | **47.9 (1.9)** | 30.4 (1.5) | 78.6 (1.2) | **6.8 (0.5)** | **622.8 (0.6)** | 35.5 (0.5) |
| MEND | 63.2 | 97.4 (0.7) | 71.5 (1.6) | **53.6 (1.9)** | 11.0 (1.3) | 53.9 (1.4) | **-6.0 (0.9)** | 620.5 (0.7) | 32.6 (0.5) |
| ROME | **91.5** | 99.9 (0.1) | 99.4 (0.3) | **99.1 (0.3)** | **74.1 (1.3)** | **78.9 (1.2)** | 5.2 (0.5) | 620.1 (0.9) | **43.0 (0.6)** |

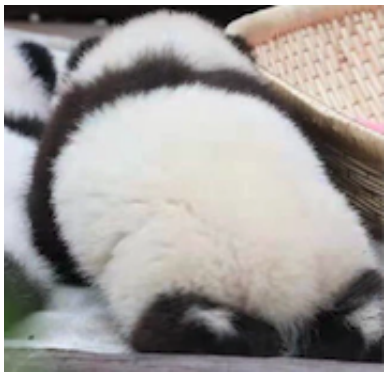
我们可以看到:

- 尽管ROME十分简单，它的性能和之前的方法相比还是很有竞争力
- 除了ROME之外，之前所有的方法都或多或少有以下两个问题: (1) 对反事实陈述过度拟合，无法泛化 (Generalization); 以及 (2) 对不相关的一些主体欠拟合并预测相同的输出 (Specificity)。而ROME则避免了这些问题，很好的实现了 **Generalization** 以及 **Specificity** 的目标。

最后本文还进行了人工评估来衡量ROME生成文本的质量，结果发现ROME生成的文本在一致性方面表现的更好，但在流畅性方面却略有不足，这说明ROME在流畅性方面引入了一些没有被相关指标 (表4中的Fluency) 度量到的损失。

## 总结

本文发现了在自回归语言模型中，事实知识是可以进行定位以及修改的，比如我们可以直接给语言模型注入知识: 埃菲尔铁塔在英国 (搬到大英博物馆 :)) 。与此同时，本文也存在着一些不足，比如它一次只能编辑一个事实知识，不能处理其他诸如逻辑、空间以及数学知识，以及会猜测出没有依据的似是而非的新知识等。但总体感觉这篇文章所做的内容还是很有趣的，相信未来也会有更多后续工作来解决这些问题 :)。



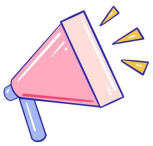
卖萌屋作者：小伟

NTU-NLP小萌新，目前对few-shot learning有着浓厚的兴趣，也是偶尔玩玩kaggle的竞赛爱好者。个人主页: https://qcwthu.github.io/

作品推荐

1.NYU & Google: 知识蒸馏无处不在，但它真的有用吗?

2.谷歌 | 多任务学习，如何挑选有效的辅助任务？只需一个公式!



后台回复关键词【入群】

加入卖萌屋NLP、CV、搜广推与求职讨论群

夕小瑶科技说

更快的AI前沿，更深的行业洞见。一线作者均来自清北、国外顶级AI实验室和互联网大厂...

528篇原创内容

公众号

喜欢此内容的人还喜欢

GPT-4打脸DeepMind：你的顶级排序优化算法，我两条提示就搞定了

夕小瑶科技说


GPT-4知道它是不是“胡说八道”吗？一篇关于大模型“自知之明”的研究

夕小瑶科技说

对大模型微调后竟能超越ChatGPT！只需要让模型模拟真实的人类交互！

夕小瑶科技说