组会汇报

陈钶杰
专业:计算数学

December 26, 2023

目录

ComputeGPT

1. 传统语言模型在解决数值问题时仍然不准确原因:因为它们的架构依赖于概率预测下一个单词，而不是执行计算.(语言模型不擅长计算的原因)
2. ComputeGPT 的工作原理是将自然语言问题转换为相关代码并执行代码来计算答案
3. 该过程的第一步是清理和微调输入提示，为代码生成模型提供清晰的指令。

compute在相关模型上的表现

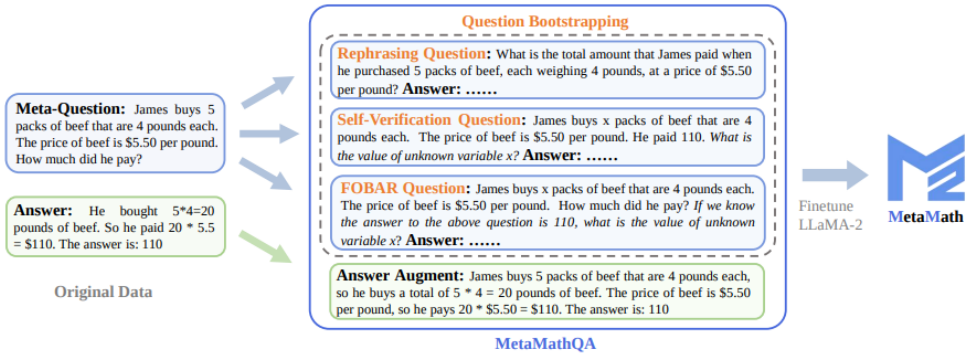
Table 1: Comparison of General Numerical Problem Solving Accuracy

Model	ComputeGPT	Wolfram Alpha	Davinci-003	ChatGPT	GPT-4
Overall Accuracy (%)	**98%**	56%	28%	48%	64%
Word Problems (%)	**95%**	15%	35%	50%	65%
Straightforward (%)	**100%**	83.3%	23.3%	46.6%	63.3%

- ComputeGPT模型的特点：
  1. ComputeGPT 可以提供代码执行过程的分步解释以及每一步背后的推理，帮助用户理解解决方案。
  2. 帮助用户理解模型所采用的底层数学概念和问题解决策略。
  3. ComputeGPT 结合了语言模型和代码执行的优势，为数值问题提供了更高效、更准确的解决方案。
  4. ComputeGPT 通过将法学硕士与封闭环境中的代码执行相集成，为数值问题提供更准确、更高效的解决方案，为这一不断增长的工作做出了贡献。

MetaMath

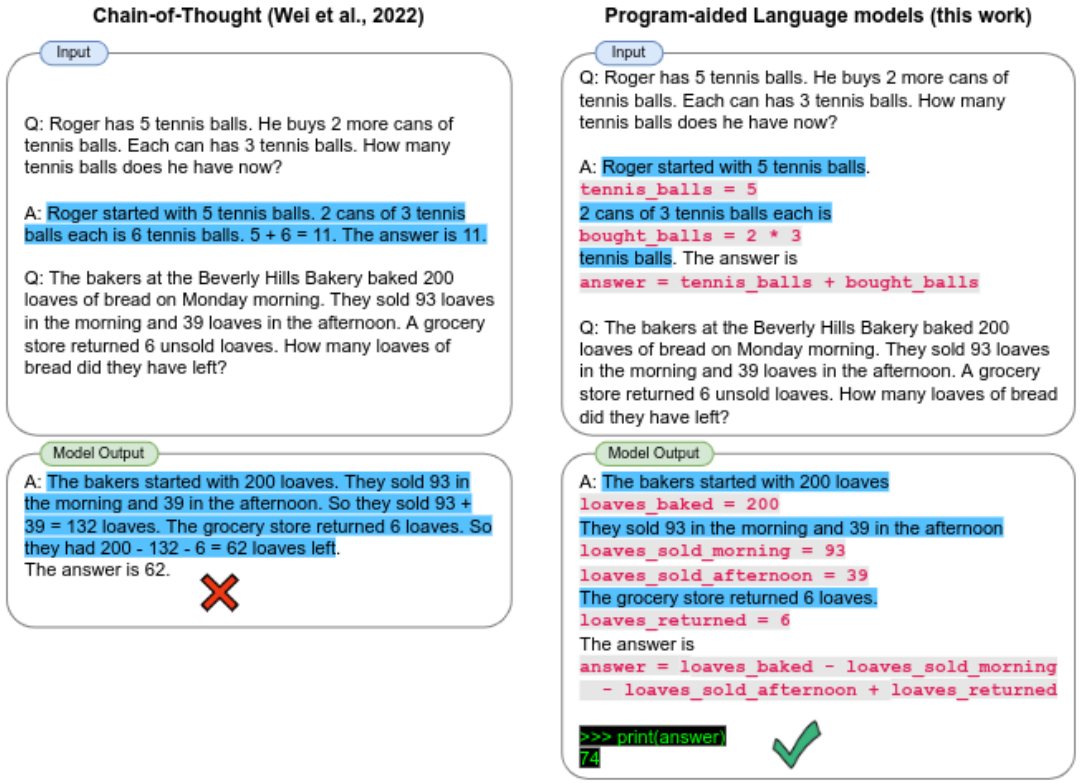
1. LLama2模型使用MetaMathQA数据集进行微调得到MetaMath模型
2. 该文章主要贡献:
   1. 我们提出了一种新颖的问题引导方法来增强训练数据集，从而产生MetaMathQA。问题引导通过前向和后向推理路径重写问题，并利用llm重新表述问题文本。



   具体解释：
   2. 在创建MetaMathQA 数据集时确定了一个重要因素——问题多样性。多样性在推理方向上尤为重要，逆向推理题对于LLM无需记忆就能理解数学知识非常有帮助。
   3. 我们的工作研究数据增强，以提高llm解决数学问题的能力。尽管很简单，但我们的方法明显优于许多复杂的方法。我们的结果强调了数据增强的重要性，也揭示了其他推理任务。

程序辅助语言模型(PAL)

1. 比思维链更精确的放法



程序辅助语言模型(PAL)

1. 两种方法的具体差别:

Wei et al. (2022) additionally augment each in-context example with *chain of thought* (CoT) intermediate steps. Specifically, each in-context example in the CoT setup is a triplet $\langle x_i, t_i, y_i \rangle$, where x_i and y_i are input-output pair as before, and t_i is a natural language description of the steps that are needed to arrive at the output y_i from the input x_i. See Figure 1 for an example. With the additional "thoughts" t_i, the prompt is set to $p \equiv \langle x_1 \cdot t_1 \cdot y_1 \rangle \| \langle x_2 \cdot t_2 \cdot y_2 \rangle \| \ldots \| \langle x_k \cdot t_k \cdot y_k \rangle$.

Figure: COT

In a Program-aided Language model, we propose to generate the thoughts t for a given natural language problem x as interleaved natural language (NL) and programming language (PL) statements. Since we delegate the solution step to an interpreter, we do not provide the final answers to the examples in our prompt. That is, every in-context example in PAL is a *pair* $\langle x_i, t_i \rangle$, where $t_j = [s_1, s_2, \ldots, s_N]$ with each $s_i \in \text{NL} \cup \text{PL}$, a sequence of tokens in either NL or PL. The complete prompt is thus $p \equiv \langle x_1 \cdot t_1 \rangle \| \langle x_2 \cdot t_2 \rangle \| \ldots \| \langle x_k \cdot t_k \rangle$.

Figure: PAL

MathGPT模型

1. MathGPT模型特征
   1. 研究背景:先前的研究通常假设大型语言模型在不使用计算器工具的情况下无法准确执行算术运算，特别是大于8位数字的乘法以及涉及小数和分数的运算。本文旨在挑战这种误解。在足够的训练数据下，20亿参数的语言模型可以准确地进行多位算术运算，几乎100%的准确率，且不会出现数据泄露，大幅超越GPT-4（其多位乘法准确率仅为4.3%）。
   2. 能力:MathGLM是一个精心设计的强大模型，可以完美地执行广泛的复杂算术运算
   3. MathGLM不仅突破了两个数字的基本算术领域，而且还解决了涉及多个数字和不同数据格式的复杂混合算术运算，与利用外部工具不同，我们专注于探索如何在不依赖外部工具的情况下增强llm固有的算术能力。

MathGPT构建模型策略

- MathGLM的模型构建策略:
  1. 我们采用分步策略构建算术数据集，作为MathGLM预训练的基础。该数据集旨在涵盖广泛的算术运算，从简单的一元运算到更复杂的九元运算。通过采用这种循序渐进的策略，MathGLM 学会了处理简单和复杂的算术表达式，这使得它能够准确地执行计算，即使是涉及大于8位的数字乘法以及小数和分数的乘法运算。
  2. 算术任务大致可以分为基本算术运算和复杂混合运算。基本算术运算包含基本的数学任务，这些任务围绕着进行涉及两个数字的简单计算。另一方面，算术任务还涵盖复杂的混合运算领域，这需要管理不同算术运算和数字格式的组合的技能。

Task	Integer	Decimal	Fraction	Percentage	Negative Numbers
Addition	nD+nD	nD.mD+nD.mD	(nD/mD)+(nD/mD)	nD%+nD%	-nD+-nD
Subtraction	nD-nD	nD.mD-nD.mD	(nD/mD)-(nD/mD)	nD%-nD%	-nD–nD
Multiplication	nD*nD	nD.mD*nD.mD	(nD/mD)*(nD/mD)	nD%*nD%	-nD*-nD
Division	nD/nD	nD.mD/nD.mD	(nD/mD)/(nD/mD)	nD%/nD%	-nD/-nD
Exponentiation	nD^nD	-	-	-	-nD^-nD
Mixed Computing	[(nD±nD.mD)*nD%]/-nD				

  3.

- 为了增强MathGLM 的算术能力,我们采用基于Transformer的仅解码器架构，并使用自回归目标在我们生成的算术数据集上从头开始训练它。
- 它还包含多种数字格式，例如整数、小数、百分比、分数和负数。这个综合数据集以各种规模创建，范围从100 万到5000 万条记录。在每个数据集中，单个算术表达式由2 到10 个运算步骤组成，涵盖一系列数学运算，例如加法(+)、减法(-)、乘法(×)、除法(/) 和求幂(^)。
- 为了符合人类的计算习惯，在算术数据集的构建中采用了分步策略。该策略不是直接计算每个复杂算术表达式的最终答案，而是将复杂表达式分解为一系列更简单的步骤，逐步生成答案。这种策略反映了人类在解决复杂算术任务时通常遵循的过程。通过在这样的数据集上进行训练，MathGLM 获得了出色的算术性能，因为它从详细的计算过程中学习了底层的计算规则。

模型相关参数

- 我们的训练工作包括4 种不同类型的模型，每种模型都有不同的参数大小。最大的模型被赋予2B参数，容量最强。接下来，我们训练具有500M 参数的第二个模型，具有100M 参数的第三个模型和具有10M 参数的最小模型。值得注意的是，尽管参数大小存在差异，但所有模型均使用由5000万条训练记录组成的相同数据集规模进行训练。MathGLM 的训练过程是使用包含5位数字范围内的数字的算术数据集启动的。

Model	Dimension	Heads	Layers	Parameters	Training Steps
MathGLM-10M	256	32	15	10M	120,000
MathGLM-100M	512	32	35	100M	155,000
MathGLM-500M	1024	32	40	500M	135,000
MathGLM-2B	2048	32	40	2B	155,000

Table 2: Model sizes and architectures of MathGLM.

MathGLM模型的训练集

- MathGLM 的训练过程是使用包含5 位数字范围内的数字的算术数据集启动的。在这个初始阶段之后，MathGLM 实现了稳定的训练收敛并在测试数据集上表现出了令人满意的性能，我们引入课程学习来增强其能力。具体来说，我们使用包含50,000 条记录的新数据集来扩充训练数据，其中包含5 到12 位数字的数字。

Basic arithmetic operations	*Complex mixing operations*
1+8/1*10+2=1+8*10+2=1+80+2=81+2=83	7826+855+4919/1050*1362-3673*7531/6726+5633=7826+855+4.6847619047619045*1362-3673*7531/6726+5633=7826+855+6380.645714285713-3673*7531/6726+5633=7826+855+6380.645714285713-27661363/6726+5633=7826+855+6380.645714285713-4112.602289622361+5633=8681+6380.645714285713-4112.602289622361+5633=15061.645714285714-4112.602289622361+5633=10949.043424663352+5633=16582.043424663352
7/5/8-3=1.4/8-3=0.175-3=-2.825	
5+5+8/1*2=5+5+8*2=5+5+16=10+16=26	
5/9=0.5555555555555556	
6*5*4=30*4=120	674+2939*2987*9430+6994/883-1642/521+2051=674+8778793*9430+6994/883-1642/521+2051=674+82784017990+6994/883-1642/521+2051=674+82784017990+7.920724801812004-1642/521+2051=674+82784017990+7.920724801812004-3.151631477927063 3+2051=82784018664+7.920724801812004-3.151631477927063 3+2051=82784018671.92073-3.151631477927063 3+2051=82784018668.7691+2051=82784020719.7691
5/8=0.625	
7*10+6=70+6=76	
4+9=13	
2+9-8=11-8=3	
2-1=1	[(-4453+9698.9284)*-4992.0]*3575/3238+-4722.991=[5245.928400000001*-4992.0]*3575/3238+-4722.991=(-5245.928400000001*4992.0)*3575/3238+-4722.991=(-26187674.572800003)*3575/3238+-4722.991=-26187674.572800003*3575/3238+-4722.991=-26187674.572800003*3575/3238-4722.991=-93620936597.76001/3238-4722.991=-28913198.455145154-4722.991=-28917921.446145155
3/2/7=1.5/7=0.21428571428571427	
7*6*4=42*4=168	
5*6/10-6*9=30/10-6*9=3-6*9=3-54=-51	(-8174.1-4561%)/-727.36226-8943=(-8174.1-45.61)/-727.36226-8943=-8219.71/-727.36226-8943=8219.71/727.36226-8943=11.300710047837788-8943=-8931.699289952163
2+8/1=2+8=10	
6/10=0.6	8689%*-5814*190-6470%/[-5900-(3540%/5945)]=86.89*-5814*190-64.7/[-5900-(35.4/5945)]=86.89*-5814*190-64.7/[-5900-0.005954583683767872]=86.89*-5814*190-64.7/-5900.005954583684=-86.89*5814*190+64.7/5900.005954583684=-505178.46*190+64.7/5900.005954583684=-95983907.4+64.7/5900.005954583684=-95983907.4+0.010966090627372149=-95983907.38903391
4+7-1+4-10=11-1+4-10=10+4-10=14-10=4	
1/5+9=0.2+9=9.2	
9*9=81	

- 模型特点
  1. 这种训练策略使MathGLM 能够首先解决更简单的示例，然后逐步应对更复杂的挑战。更重要的是，这种方法使MathGLM能够通过从相对较小的示例中学习来提高其能力，强调MathGLM处理日益复杂的任务或数据模式的效率。
  2. 为了获得更好的性能，我们对MathGLM 采用了两种训练策略。第一个是在单独的数学数据集上微调GLM 主干模型。此过程使MathGLM能够通过学习数学数据集的独特特征来专门理解和解决数学应用问题。然而，这种策略损害了MathGLM 的通用能力。为了规避这一限制，第二种策略是继续在结合了数学和文本内容的混合数据集上训练GLM 主干模型。这有助于平衡数学应用题的专业化与保留MathGLM 的通用能力。
- 模型评估指标:
  1. 准确性通常是通过比较MathGLM 的输出和真实答案来衡量的。在我们的实验中，我们遵守标准舍入规则，限制生成的答案精确到小数点后两位。当正确舍入的答案与MathGLM 生成的答案一致时，我们将此结果分类为正确答案。
  2. 相对误差是用于评估MathGLM 有效性的另一个重要指标，它量化了MathGLM 生成的输出与正确答案之间的差异。相对误差(RE) 使用以下公式进行量化：

1. 现存的一些问题:
   1. 模型可能会对“大于”或“小于”等不明确的短语进行不同的解释，从而导致解决方案不准确。此外，MathGLM-GLM-10B 往往难以解决涉及复杂数学运算的问题。因此,它可能会提供部分正确的算术解决方案，但无法得出最终的正确答案。
2. 模型结果:
   1. 在本文中，我们的主要重点是评估法学硕士的数学推理能力，包括算术运算和数学应用题。对于算术任务，我们结合分步解决方案和课程学习，从头开始训练基于Transformer 的语言模型。通过对大量数据的全面训练，我们建立了一个拥有20 亿个参数的语言模型，可以在多位数算术任务中实现出色的精度，远远超过GPT-4 的结果。这一发现令人信服地挑战了普遍的认知，即法学硕士在执行精确算术运算时面临限制，尤其是在不依赖外部计算辅助的情况下处理多位数、小数和分数时。当转向数学应用题时，我们重建了一个富含多步算术运算的数据集。在对源自GLM-10B 的改进数据集上的MathGLM 进行微调后，它在5,000 个样本的中国数学问题测试集上实现了与GPT-4 类似的性能，展示了其强大的实力。

模型结果:

-

Model	ACC	RE
GPT-4	22.22%	-
ChatGPT	13.25%	-
text-davinci-003	9.79%	-
text-davinci-002	4.08%	-
Galactica-120b	7.97%	-
Galactica-30b	7.02%	-
LLaMA-65b	5.02%	-
OPT-175B	3.83%	-
BLOOM-176B	3.96%	-
GLM-130B	3.06%	-
MathGLM-10M	64.29%	97.96%
MathGLM-100M	73.47%	98.23%
MathGLM-500M	89.80%	98.82%
MathGLM-2B	94.90%	98.98%

下一步计划

1. 学习MathGLM-10M学习其详细流程，能否为我们的实现提供思路。

谢谢老师和同学们的聆听!