规则组合实现大语言模型的数字理解能力

陈钶杰
学号: 22135030

12. März 2024

Inhaltsverzeichnis

1 摘要(Abstract)

在以往的研究中,通常大型语言模型在不借助外部工具的情况下,无法准确进行算术运算。因为现有的语言模型都是在进行记忆数学题,而不是理解数学题,而数学计算是无穷无尽的,而语言模型无法记忆无穷无尽的数学规则,自然无法进行精准的数学计算了.本文旨在挑战这种误解。本文提出的关于规则学习的语言模型,通过对不同规则的组合来实现数字的精准计算,以及其他相关的复杂任务.我们在本文中提出了规则组合学习的语言模型.首先我们设计了一套关于数学计算中用到的基本规则,复合规则和迭代规则数据集,然后将我们所制定的规则进行文本嵌入并进行预训练处理,最后得到的预训练模型便能够进行复杂的数字计算.模型实验中,我们的模型在数字计算上,特别针对一些高位数字计算上,我们的模型在各类数字计算问题上准确度都优于现有的SOTA模型;通过测试在随机生成的数值问题上计算的准确度实验证明了我们模型在解决问题能力的全面性.通过测试同时解决两种不同数值问题的实验证明了我们的模型在解决问题能力上具有一定的多样性.

关键词:Transformer;nlp;Arithmetic Caluculation;Meta Learning;
(英文版文献综述)

2 引言(introduction)

在自然语言处理领域的最新进展中，如GPT-4等大型语言模型已经取得了显著的成功，在许多任务中均展示了惊人的理解能力。然而，在处理数字问题等专业领域时，这些模型仍然面临巨大挑战。所谓的数学问题,涵盖非常广,包括加减乘,求导,积分和解方程等.然而语言模型却在最简单的数字加减问题上就遭遇巨大的困难.
研究人员探索了基于Transformer架构的语言模型解决数学问题的潜力,并在这一领域取得了许多的进展,compute-gpt,mathgpt是相对成功解决复杂数学问题的语言模型中的优秀变体:compute-gpt提出了一种语言模型和外部工具(如python工具)结合方式来解决数值计算问题;MathGPT借鉴了传统语言模型微调的方法,通过提高基本数据集的训练大小,在更大基数的数据集训练之下，相比传统的语言模型能解决更多数字计算问题。尽管上述模型都在一定程度上优化了语言模型解决数学问题的能力，但是这些模型部分是通过借助外部工具，部分是通过训练提高数据量的方式来提升数字计算能力，并没有真正的让语言模型自己掌握一种数字的计算。因此，我们引入了元学习中提到的规则组合能力，我们尝试令语言模型学规则的调用和组合,并通过对问题的多次迭代，来解决具有逻辑性的问题。实验结果显示，直接通过喂养大量数学问题的方式虽然在数据量喂养较大的时候，可以使得语言模型展现出一定的数字计算能力，但通常数字计算问题的数量是无限的或者巨大的，语言模型在数字问题这种具有严格逻辑的问题上所展现出来的泛化能力也非常的有限。相比之下，我们的模型通过规则层面的学习，可以用极少量的数字规则训练数据集，模型通过对问题进行迭代计算，可以解决任意的数字计算问题。此外，我们的简易语言模型也可同时接受多种不同的任务，我们的模型可以同时完成多种任务，比如不仅可以进行任意数值的加减任务，同时也可以完成向量计算中比较复杂的外积任务。并且在进行外积任务时可以利用已经学得的部分数字规则来进行外积的计算。后续的实验结果显示，我们提出的模型，在数字计算任务以及处理复杂逻辑问题上除了与chatgpt3.5,chatgpt4.0持平以外，超越了现有的SOTA(State-of-the-art model)模型。这些贡献总结如下:

1. 设计了一套适用于语言模型的规则数据集
2. 通过对规则的编码嵌入，来实现语言模型对规则的调用和组合.
3. 设计了一种全新的模型迭代方法，更加接近人类对问题逐步求解的思考过程。
4. 可以通过少量的规则学习，来解决大量的问题，远小于其他语言模型的训练数据集的数量，同时保证解决问题的准确度。

3 相关工作(relate work)

3.1 背景介绍(back ground)

大语言模型通过大规模数据学习语言规则，但在数学计算中，数字之间的运算是无穷无尽的，但是数字计算是有明确规则的,数字运算对于模型而言是一项具有挑战性的任务。当前的语言模型在这方面仍然存在一些限制，因此有必要进一步研究数字规则学习的方法，以增强其在数字运算中的适用性。

3.2 Large Language Models

大型语言模型(LLM)在自然领域展示了强大的功能,即语言处理任务，在大量语料库上进行训练多样化且未标记的数据，在不同任务中均展示出了良好的通用能力,显著改变了该领域的研究范式。语言模型在经过对广泛语料库的预训练，这些模型获得了强大的语言理解能力和生成能力，使其在各种基准测试中具有卓越的性能，例如情感分析,机器翻译等任务,均有良好的表现.
尽管如此，目前最优秀的大型语言模型ChatGPT和GPT-4在语言理解和生成中，对文本问题具有精确的理解且给出相当精确的回答,但是在解决数学问题时仍然遇到挑战,数学问题都是只有唯一答案的,这对于模式识别的语言模型而言无疑是困难的,语言模型本质上是一种概率模型,是将可能性最大的结果告诉我们,然而数字计算,数学推理是非常严谨的,每一步都需要严格遵循,这就导致语言模型以概率计算的方式去进行数字计算会出现巨大的偏差。这项工作致力于解决和提高语言模型在解决数学问题领域的表现，包括算术任务和以及严格的逻辑推理问题。

3.3 Meta Learning

元学习，是机器学习领域的一个重要分支。元学习的核心在于让机器学习算法能够利用过去的经验来提高其未来学习新任务的效率和效果。与传统的机器学习相比，元学习更加关注于如何在多个任务上学习通用的学习策略，从而在面对新任务时能够快速适应。人类语言和思维的力量源于系统的组合性，即从已知成分中理解和产生新组合的代数能力。Fodor 和Pylyshyn 曾提出了一个著名的论点:即人工神经网络缺乏这种能力，因此不是可行的思维模型,虽然从那以后的几年里,神经网络取得了巨大的进步，但此挑战仍然存在。Meta Learning [**?**] 成功地解决了Fodor 和Pylyshyn 的挑战，并提供了证据，证明神经网络在优化其组合技能时可以实现类似人类的系统性。为此，他们引入了组合性元学习(MLC)方法，用于通过动态的组合任务流来指导训练。MLC 实现了类人泛化所需的系统性和灵活性，还在几个系统的泛化基准中提高了机器学习系统的组合技能。

3.4 Arithmetic Calculation

早期已经有不少研究集中在使用大型语言模型解决数字计算问题。比如MetaMath[5],该模型通过喂养大量的算术和符号推理任务数据集,在经过大量数据训练后,来实现语言模型对数学问题的理解以及求解,该方法能够实现对简单数字问题的求解,一旦问题变长,比如数字长度变长,模型求解起来将会非常困难,究其本质是因为模型只是记忆了数学问题,但并没有深刻理解其中的数学规律和算法,同时这也是当前主流模型所面临的问题.因此,ComputeGPT[3],开始探索了语言模型和算术任务的外部工具的集成,这种方法的好处是计算能力强大,且计算的数值不受任何位数的影响,但在这种情境下语言模型只是起到了一个文本识别器,将数字,运算符号识别出来,真正起到运算的作用的是算术任务的外部工具,而语言模型本身并未进行数字计算的.同理,像程序辅助语言模型[1]也是类似的,这是一种新的自然语言推理方法，使用程序作为中间推理步骤。与现有的基于LLM的推理方法不同，主要思想是将求解和计算使用外部的Python解释器，而不是使用LLM本身来对问题进行理解和求解。这种方法相比ComputeGPT,语言模型的需要处理更多的问题,但是核心的计算依旧是使用外部工具.最新的模型MathGPT[4]旨在不使用计算器工具情况下执行算术运算，其20亿参数的语言模型可以相对准确地进行多位算术运算，此文的核心思路是通过训练较多的低位数的数值计算方法,以此来对高位数运算的准确性.该模型所用的方法与利用外部工具不同，其专注于探索如何在不依赖外部工具的情况下增强LLM的固有的算术能力,但是这个模型依旧没有让模型理解数字运算的基本规律.而是通过记忆大量基本计算规律来进行问题的求解.

4 研究方法(Methodology)

为了提高语言模型在数学推理中的有效性，我们提出了MathRuleGLM模型，其旨在提升语言模型在数学推理中的性能。首先，我们的模型MathRuleGLM受到了元学习的启发，更加注重学习通用的学习策略，致力于使用规则来执行准确的算术任务。它通过在训练过程中动态的整合数学计算的基本规则以及复合规则的学习，来提高模型组合规则的能力。MathRuleGLM模型在直接面对一个复杂的未见过的算术表达式时，通过将迭代策略整合到其架构之中，对每一次迭代过程中根据算式表达式选用合适的规则，经过有限次迭代后，即可得到相对精确的计算结果。与直接解决复杂算术表达式不同，MathRuleGLM采用这种策略，使用规则的进行计算更接近人类解决数学问题的方式。其次，MathRuleGLM用较少的训练数据来训练模型，但比当前主流大语言模型达到更高的计算准确度。通过利用这一策略，MathRuleGLM不仅可以处理单任务，同时学习多种任务，对不同任务中每个规则是会出现交叉的，我们的模型依旧可以通过动态的学习交叉的规则，可同时完成多种不同的任务，且相互不受影响。

4.1 算术任务上的学习

算术任务大体上可以分为高位数的加,减法运算。另一方面，算术任务还包括计算向量的外积等复杂任务，这需要具备管理多种不同算术运算和数字格式的组合的技能。MathRuleGLM的主要计算能力就如下表中展示的一样，

4.1.1 算数训练数据集

用于训练的算术数据集经过精心设计，其中包含了最简单的个位数算术任务以及各种算术规则。这个数据集被深思熟虑地设计以包含各种操作，其中包括对齐规则，进位规则，退位规则，简单计算规则，组合规则等。我们创建了一个小型数据集，大约是2万条记录不等。在这些数据集中，算术表达式仅仅包含了最最简单的个位数计算，例如九九加法表，九九减法表等，其他所涵盖的均是一系列的数学规则操作，比如数字的对齐规则，数字的进位规则，数字的退位规则，外积的规则，组合规则的学习等。为了与人类的计算习惯保持一致，我们采用逐步策略，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成一系列更简单的步骤，逐步生成答案。这种策略反映了人们通常在解决复杂算术任务时遵循的过程。通过在这样的数据集上训练，我们的模型在数值计算上具有良好的算术性能，因为它所学习的是底层的计算规则而不是单单依靠记忆得到的结果。图x中提供了一些从算术数据集中抽取的训练示例，展示了数据集中逐步策略的具体实现手段。
(将测试的结果数据都放上来*)

4.1.2 模型和训练程序

模型和训练程序。表中报告了所有不同模型参数的模型概览。我们对几个不同参数量的模型都使用相同规模的数据集进行训练，比较最终的结果。我们的训练主要包含了以下4种不同参数量的模型，其中最大的模型配备了10亿（1B）参数，使其在容量方面最为强大。在此之后，我们还有用5亿（500M）参数训练的第二个模型，以及用7000万（70M）参数训练最小的模型。值得注意的是，尽管模型参数大小存在差异，数据集大小规格一致，该数据集包含2万多条训练记录。(准备一个模型规格大小的表格)

4.2 多任务上的学习训练架构

对于多任务的数据集而言，算术表达式依旧只包含了个位数计算，但是对于其他规则的涵盖需要作出一些改变，我们的测试例子是学会做加减乘，以及向量的外积，这是两种完全不相同的任务，一个是数字的加减，一个是向量的外积。但是在规则上有一些共性，比如向量的外积具体计算时，也需要进行个位数的加减运算，也需要进行对齐等，因此我们找出了两种不同任务的特性和共性，分别设计出相应的公共规则和特殊规则，最后合并训练数据，得到相应的结果。同时我们的模型在针对不同的任务时会根据所习得的规则作出相应的回答，具体回答依旧采用逐步策略，逐步生成答案。图x中提供了一些从算术数据集中抽取的训o示例，展示了数据集中逐步策略的具体实现手段。

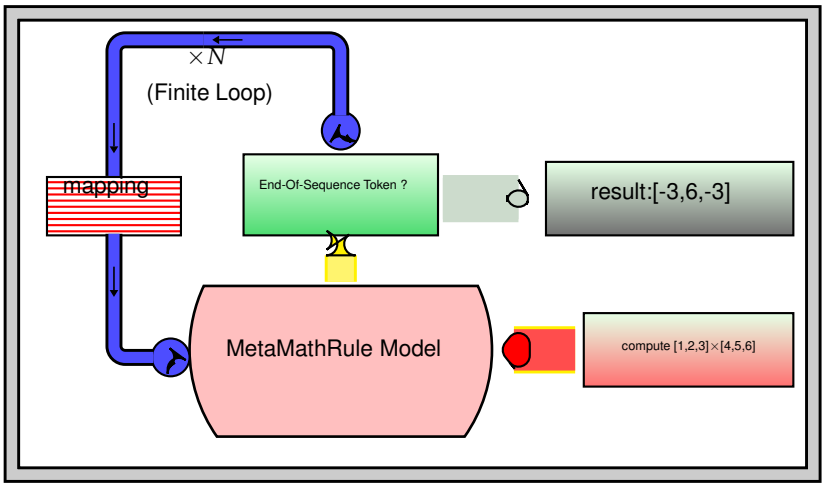
4.2.1 算数训练数据集

数据包含了最简单的个位数算术任务以及各种算术规则，同时还包括数字，向量的对齐规则，进位规则，退位规则，简单计算规则，组合规则等。由于我们主要测试方法的实用性，我们创建了一个小型数据集，大约是1千条记录左右.

4.2.2 模型和训练程序

由于我们训练的数据集比较小，所以我们只用了3000万(30M)的参数训练模型。已经能对数据集有较好的学习效果。

4.3 模型架构



4.3.1 模型训练结构

在关注算术任务的同时，我们训练了一系列基于Transformer的语言模型，称为通用语言模型，以解决数学问题。我们的模型架构分别如上图所示，将问题输入训练好的MetaMathRule模型，根据模型输出是否包含结束标识符，来判断是否需要进行循环。如果输出内容无标识符，重新输入到模型中，但是中间结果得经过一次映射。

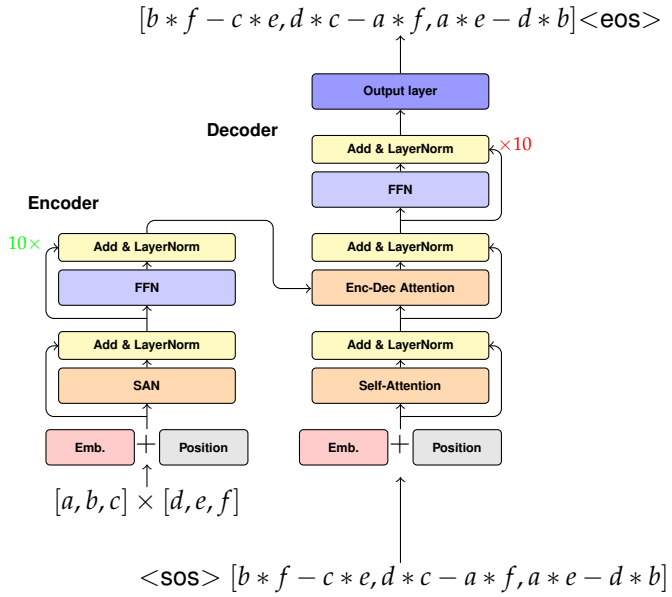


Abbildung 1: Transformer训练架构图

4.3.2 模型训练策略

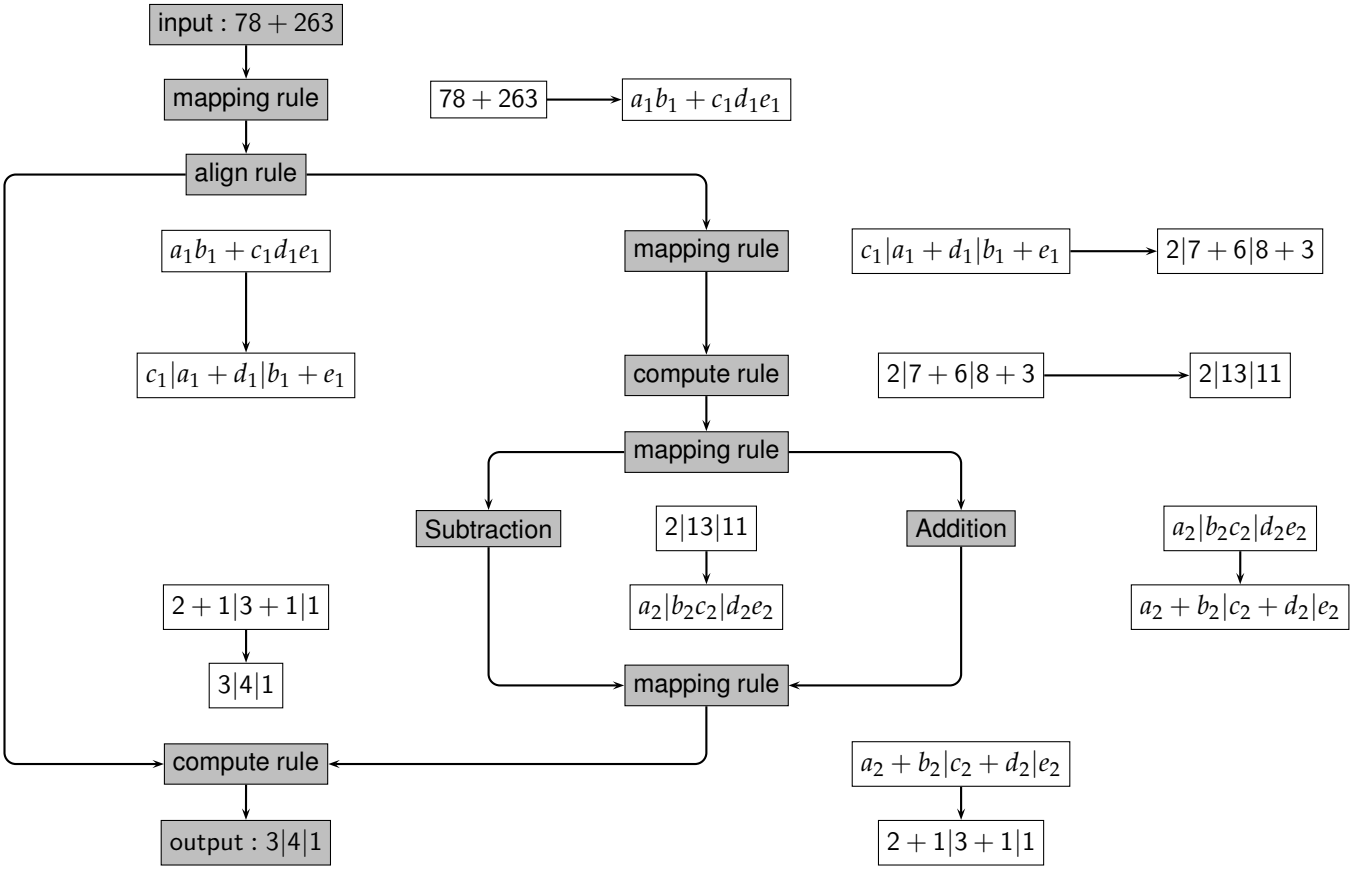


Abbildung 2: 如何调用规则求解算式

4.3.3 模型计算示例一(加法)

(详细说明ing)

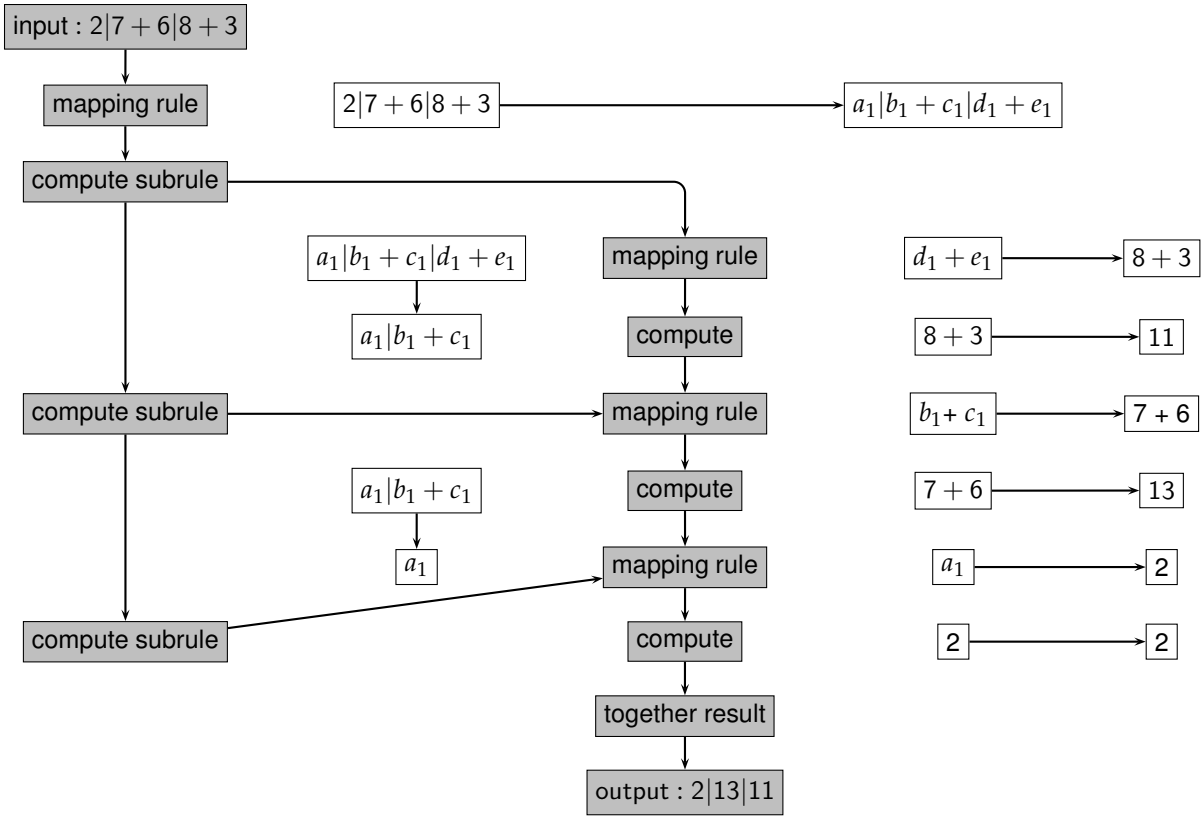


Abbildung 3: compute rule

4.3.4 模型计算中间过程

(详细说明ing)

5 实验设定(Experimental settings)

5.1 数据集

5.2 实验过程

5.3 评估指标

6 实验(experiments)

6.1 实验结果

Task	Interger
Addition	x + y
Subtraction	x - y
Multiplication	x * y
Division	x / y

Tabelle 1: 不同参数量的模型结果对比

Size	$0-10K$	$10K-1M$	$1M-1B$	$1B-10B$	$10B-70B$	$70B-\infty$
GPT-3.5	-	-	-	-	-	95.29%
GPT-4	-	-	-	-	-	98.32%
llama2-7b	-	-	-	2.3%	-	-
llama2-13b	-	-	-	-	21.89%	-
llama2-70b	-	-	-	-	76.09%	-
Google-PaLM	-	-	-	-	-	95.96%
Qwen-72b-Chat	-	-	-	-	-	93.94%
MathRuleGLM	100%	100%	100%	100%	100%	100%

6.2 结果分析

MathRuleGLM的总体目标围绕展示语言模型在数学推理领域的能力。为了验证这一点，我们设计了一种复杂算术计算实验，将普通算术问题进行分类,专门针对先有语言模型计算不准确的问题进行测试比较,我们的模型能全面覆盖当前模型在部分计算问题上的不足，为评估模型在数学推理中的熟练程度提供了强有力的评估。

6.3 学习算术

6.3.1 评估指标

为了衡量MathRuleGLM在算术任务上的能力，我们采用以下指标来评估输出结果。准确度通常通过比较MathRuleGLM的输出和实际答案来衡量。相对误差是用来评估MathGLM有效性的另一个重要指标，它量化了MathRuleGLM生成的输出与正确答案之间的差异。

6.3.2 结果与分析

对于算术任务，我们预先训练了一个名为MathRuleGLM的基于Transformer的模型，其在预训练和推理阶段都具有()个模型参数。为了准确评估MathGLM的有效性，我们将其性能与领先的大型语言模型（LLMs）如GPT-4和ChatGPT进行对比。MathRuleGLM在处理算术任务的各个方面上都表现优越，胜过所有其他模型。而且MathRuleGLM是一个非常小的模型变体，即只有()万参数。尽管其紧凑的参数大小，我们的模型在特定算术任务上表现依旧优于GPT-4和ChatGPT。这个惊人的结果展示了规则调用的有效性，该方法涉及将复杂的算术表达式使用复合规则来解决问题，赋予了它识别和理解算术任务中微妙之处的能力。它有效地学习了算术操作的基本规则和原理，使其能够生成准确而精确的解决方案。对于模型在不同参数规模下的表现，我们观察到MathGLM的算术性能与其参数数量的增加直接相关。这一发现表明随着模型大小的增加，其性能呈现相应的增强。总的来说，在复杂的算术任务评估结果中，MathRuleGLM的表现异常出色。通过子规则和复合规则的学习,这些模型很大程度上弥补了GPT-4和ChatGPT的一些数学计算不足之处. 去比对当前的llama模型等模型的计算能力

- 3种特定数据集类型
  - 进位加法:
    1. 48551+1449=
    2. 7223+32777=

- 负差:

  1. 1140-26787=
  2. 234-15579=

- 逐位差分

  1. 41085-80976=
  2. 65570-73618=

Tabelle 2: 3种特定的数据集实验结果

Task	进位加法	负差	逐位差分
GPT-3.5	97.2%	99.4	95.29%
GPT-4	100%	97.8%	98.32%
llama2-70b	19.4%	50.2%	76.09%
llama2-13b	5%	20.8%	21.89%
llama2-7b	3.6%	4%	2.3%
Google-PaLM	52.6%	43.6%	95.96%
Qwen-72b-Chat	85.8%	86.4%	93.94%
MathRuleGLM	100%	100%	100%

6.4 使用规则学习去处理其他逻辑推理问题

7 总结与展望(Summary and Outlook)

Literatur

[1] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*, 2022.

[2] Brenden M Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023.

[3] Ryan Hardesty Lewis and Junfeng Jiao. Computegpt: A computational chat model for numerical problems. *arXiv preprint arXiv:2305.06223*, 2023.

[4] Zhen Yang, Ming Ding, Qingsong Lv, Zhihuan Jiang, Zehai He, Yuyi Guo, Jinfeng Bai, and Jie Tang. Gpt can solve mathematical problems without a calculator. *arXiv preprint arXiv:2309.03241*, 2023.

[5] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.