

文章编号: 1003-0077(2025)07-0102-12

基于正负例思维链的表格-文本混合金融数据自动问答方法

李 希^{1,2}, 刘喜平¹, 舒 晴¹, 谭 钊¹, 万常选¹, 刘德喜¹

(1. 江西财经大学 计算机与人工智能学院, 江西 南昌 330013;

2. 江西科技师范大学 智能制造学院, 江西 南昌 330038)

摘 要: 金融领域表格-文本混合数据的自动问答面临复杂数值推理等挑战。针对这一挑战, 该文提出了正负例思维链方法。思维链技术通过选取演示样本, 搭配提示指令, 能有效提升大语言模型的多步骤推理能力。但思维链的演示样本多为人工制作, 费时费力, 且影响大模型推理。受正例和负例对学习效果影响的启发, 该文从大模型的推理结果中抽取样本, 构建正例样本池和负例样本池。采用静态和动态相结合的策略选取不同类型的演示样本: 选择最佳正例能够保障大语言模型输出的准确率, 选择相似负例能够指导大语言模型规避错误推理。实验结果显示, 该方法在 FinQA 数据集上的准确率提高了 3.6%, 在 FinQA-fix 数据集上的准确率提高了 12.73%, 显著提升了大模型的数值推理能力。

关键词: 思维链; 正负例; 表文混合问答; 大语言模型; 数值推理

中图分类号: TP181

文献标识码: A

Positive and Negative Chain-of-Thoughts Based Question Answering Method for Table-Text Financial Data

LI Xi^{1,2}, LIU Xiping¹, SHU Qing¹, TAN Zhao¹, WAN Changxuan¹, LIU Dexi¹

(1. School of Computing and Artificial Intelligence, Jiangxi University of Finance and Economics,

Nanchang, Jiangxi 330013, China;

2. School of Intelligent Manufacturing, Jiangxi Science and Technology Normal University,

Nanchang, Jiangxi 330038, China)

Abstract: A question answering for table-text hybrid data in the financial domain is a challenging task. This paper proposes a positive-negative chain-of-thoughts method. Inspired by the influence of positive and negative examples on modeling, chain of thought (COT) samples are extracted from the reasoning results of large language models to construct positive and negative sample pools. A combination of static and dynamic strategies is employed to select different types of demonstration samples; choosing the best positive examples ensures the accuracy of the large language models output, while selecting similar negative examples guides the model in avoiding incorrect reasoning. The experimental results show that the method improves the accuracy on the FinQA dataset by 3.6%, and on the FinQA-fix dataset by 12.73%.

Keywords: chain-of-thoughts; positive and negative examples; table-text hybrid question answering; large language models; numerical reasoning

0 引言

在现实世界中, 结合表格和文本的混合数据

(以下简称表文混合数据) 非常普遍, 尤其是在金融领域。例如, 财务报告中既包含资产负债表、利润表、现金流量表等表格数据, 又包含财务情况说明书这样的文本数据。然而, 在现有的研究中, 对这种表

收稿日期: 2023-11-26 定稿日期: 2024-03-13

基金项目: 国家自然科学基金(62076112, 61972184, 62462034); 江西省自然科学基金(20192BAB207017, 20232ACB202008); 江西省研究生创新专项资金项目(YC2022-B157)

文混合数据的问答研究还不够充分。与一般领域自动问答任务相比,金融领域表文混合数据自动问答需要理解专业知识和复杂的数值推理。要理解和回答这种混合语境中的数值推理问题,需要依靠表格和文本之间的密切关系,从表格和文本中提取相关数值信息,推理出所需计算公式,得出答案。

为了解决表文混合数据问答任务中的信息提取与生成推理问题,Chen 等人提出了一个基于检索与生成的问答架构,先检索相关支持事实,再生成可执行的推理程序,回答问题并输出答案^[1]。这种方法有效地利用了文本与表格数据之间的关联性和互补性。Zhu 等人提出了 TAGOP 模型,该模型采用序列标记方法从表格中提取相关单元格,并从文本中提取相关跨度,以推断其语义,使用一组聚合运算符对问题进行符号推理,得到最终答案^[2]。该方法将表文混合问答任务转化为序列标记任务。RegHNT 是一种关系图增强的表文混合数值推理模型,它将表文混合的数值推理建模为表达式树生成任务,其中的关系图建模方法是对问题、表和段落进行对齐操作,有效地利用了图结构来表示和处理表文混合数据^[3]。

Wei 等人^[4]发现,大语言模型^[5-7] (Large Language Model, LLM)可以在没有任何训练或微调的情况下,通过一些输入输出示例来解决复杂推理任务。这种提示方法被称为“思维链”(Chain-of-thoughts, CoT),它能够在广泛的文本和数值推理数据集上展现优异性能。CoT 使用 LLM 进行推理和计算,即语言模型不仅需要生成数学表达式,还需要在每个步骤中执行数学运算。然而,大模型求解这些数学表达式并不理想,非常容易出现算术计算错误,尤其是在处理较大数值时。为了解决计算问题,Chen 等人^[8]提出了“思维程序”(Program of Thoughts, PoT),它使用语言模型(主要是 Codex)来生成文本和程序代码,并最终给出答案。在 PoT 中,计算委托给程序解释器,从而将复杂的计算与推理和语言理解分开处理。PoT 方法在几个数学问题数据集上(如 GSM8K, AQuA, SVAMP, TabMWP)效果显著,但是在金融领域表文混合问答数据集上(FinQA)的准确率只有 64.5%,落后于其他数值计算数据集 10~20 个百分点。通过实验发现:①演示样例的内容、数量及顺序,都会影响实验结果的变化;②大模型在编写 Python 代码的过程中,变量名称会出现随机缺字现象,从而造成结果错误;③FinQA 数据集质量不高,有近 20% 的样本存在答案错误,对最终的实验结果影响很大。

针对上述问题,本文提出了正负例思维链,即一种基于正负例的方法来自动筛选有效演示样本,基于高质量的数据集,该方法可以显著提高大语言模型在数值推理任务中的性能。本文的主要贡献如下:

(1) 用 LLM 构建正例样本池和负例样本池,作为目标演示样本的来源。

(2) 对样本池按问题聚类,取质心样本,选取最佳正例确保 LLM 输出结果的准确率,动态分配相似负例以指导 LLM 规避错误推理,比原始 PoT 方法(选取 4 个演示样本)准确率提高了 3.6%。

(3) 在提示(Prompt)部分的样例中,增加代码注释语句,对变量及公式的意义进行了较为详细的说明,指导 LLM 生成相对简洁的代码,从而避免变量名缺字现象,提高了 LLM 代码生成效率。

(4) 修订 FinQA 数据集(FinQA-fix 数据集),比较了数据集质量对于推理的影响,发现准确率提升了 12.73%。

1 相关工作

1.1 思维链 CoT

上下文学习(In-Context Learning)是指大模型通过一些输入输出的例子,就能学会一个新的任务,如分类、翻译、填空等。GPT-3^[5]展示了大模型运用少量样本执行预测的强大能力。Xie 等人^[9]解释了上下文学习是怎么发生的,它认为大模型在预训练的时候就已经学会了一些潜在的概念。当大模型看到提示中的例子时,它就会用这些潜在的概念来理解它们之间的关系,从而学习新的任务。Xie 等人用了一个人造的数据集来证明这一点。另外,还发现了一些有趣的现象,比如模型越大越聪明,例子的顺序很重要,有时候没有例子比有少量例子更好。Min 等人^[10]探究了是什么让上下文学习更有效,它发现提示中的例子并不需要真实或准确,只要能告诉模型一些基本的信息,如有哪些标签、输入文本是什么样的、输出格式是什么样的等。Min 等人用了很多不同的模型和任务来验证这一点,还提出了一些新的问题,比如模型到底能从例子中学到多少东西,还是只是在做推理。

尽管大模型在一系列自然语言处理任务(Natural Language Processing, NLP)中取得了显著的成功,但普遍认为它们的推理能力仍具有局限性,直到思

思维链 CoT^[4] 被提出。CoT 可以分为两大范式。其中一个是在问题后添加一个提示,如“让我们一步一步地思考”,以促进大模型的推理链。Kojima 等人^[11]给出的这种提示范式不需要输入输出演示,因此称为 Zero-Shot CoT。使用 Zero-Shot CoT,大模型已被证明是不错的零样本推理器。另一种范式是通过手工推理演示样本进行 Few-Shot 提示。每个提示都有一个问题和一个推理链。推理链由一个基本原理(一系列中间推理步骤)和一个预期答案组成。之后,其他几项工作 Drozdov 等人^[12]; Zhou 等人^[13]也提出利用大模型,通过中间步骤来解决推理任务。

数学应用题(Math Word Problems, MWP)是一种求解数学问题任务,其用文字描述了一个实际情境,要求根据文字信息,生成合适的数学表达式或方程式,然后计算出答案。CoT 使用大模型在 MWP 任务上进行推理和计算,展现出优异性能。但是,大模型在做 MWP 任务时经常出现以下错误:算术计算错误;不擅长求解复杂的数学表达式;遗漏步骤;在表达迭代时效率很低,尤其是在迭代步骤数很大的情况下。为了解决这些问题, PoT 与 PAL^[14]都主张使用大模型读取自然语言问题,并生成程序作为中间推理步骤,将解决计算步骤委托给解释器,如 Python 解释器。为了解决遗漏步骤问题, Plan-and-Solve^[15]首先设计一个计划(Plan),将整个任务划分为多个更小的子任务,然后根据计划执行子任务。Faithful CoT^[16]将推理任务分解为两个阶段:翻译(自然语言查询→符号推理链)和问题解决(推理链→答案)。思维链 CoT 与自洽性(Self-Consistency)^[17]相结合是提升推理能力的有效方法。然而,这些方法没有充分利用大模型生成的答案来指导后续的回答。PHP^[18]通过使用之前生成的答案作为提示(Hints),来实现用户和大模型之间的自动多轮交互,从而逐步引导大模型生成正确的答案。

1.2 表文混合问答

2020 年, Chen 等人^[19]首次提出表文混合数据形式的开放域问答任务,给出了第一个结构化与半结构化混合的数据集 HybridQA。在该任务中,模型需要同时处理表格和文本信息,即模型兼具表格逻辑理解能力和文本语义理解能力,问题需要多跳才能回答。TAT-QA^[2]从真实的财务报告中提取样本,构建问答数据集,包含表格和文本数据,其中

问题通常需要数字推理来推断答案。同时提出 TAGOP 模型,能够结合表格和文本进行推理。它使用序列标记方法从表格中找出相关的单元格,从文本中找出相关的跨度,然后根据它们的语义,用一组聚合运算符进行符号推理,从而得到答案。FinQA^[1]是一个财经领域的数据集,它包含了 8 281 个由金融专家编写的问题和答案,以及详细的数值推理过程。该论文还设计了一个检索-生成的问答框架,它可以先检索出支撑事实,再生成推理过程来回答问题。

在表格事实验证方面, Ye 等人^[20]提出利用大语言模型作为表格推理的分解器,将大表格分解为小表格,将复杂的问题分解为更简单的子问题。利用分解的子表格和子问题,通过一些上下文提示的例子来获得最终答案。为了改进 CoT 推理的忠实度, Radhakrishnan 等人^[21]通过强制让大模型在不同的上下文中回答更简单的子问题,提高模型生成推理的忠实度。如何让大语言模型实现对表格的统一理解和操作? Zha 等人^[22]提出 TableGPT 框架,通过对大量文本和表格数据进行联合训练,并通过指令链的方式实现对表格的复杂操作。

PoT 选取演示样本的做法是:为 10~20 个样例编写提示,然后在一个小的验证集上人工调整样本,最后选择最佳的 4~8 个演示样本进行整体评估。这种方法需要人工手动选择并制作演示样本,费时费力,且选出的演示样本没有很好的可解释性。本文旨在探究演示样本对大模型在数值推理任务上的影响。本文重点考察了构建演示样本池的方法和演示样本的选取策略,增加了财经领域的注释语句,以提高推理的可读性,使所选样本具有很好的可解释性。同时,修订了数据集以提高数据集质量,从而更好地激发大模型在财经领域的应用潜力。

2 任务描述与研究动机

在表文混合问答任务中,定义上下文文本为 E , 结构化表格为 T , 给出问题为 Q , 目的是生成推理过程 $G = \{w_0, w_1, \dots, w_n\}$, 其中, w_i 是中间过程的 tokens, 执行 G 得到最终答案 A , 如式(1)所示。

$$P(A | T, E, Q) = \sum P(G_i | T, E, Q) \quad (1)$$

这里的 $\{G_i\}$ 是指所有中间推理过程。

近年来,基于 LLM 的少样本学习技术已被证明具有强大的生成和泛化能力^[23],广泛应用于各类自然语言处理任务。思维链技术则进一步提升了

LLM 的推理能力,能显著提升复杂任务上的生成效果。本文将少样本学习和思维链技术用于表文混合问答任务。采用少样本学习的表文混合问答任务可以公式化为:

$$P_{LM}(A | T, E, Q) = \sum_i P(G_i | \text{prompt}(\{T, E, Q\}, \{T_j, E_j, Q_j, A_j\}_{j \in m})) \quad (2)$$

其中, $\{T_j, E_j, Q_j, A_j\}_{j \in m}$ 是演示样例(Demonstrations),和问题样本 $\{T, E, Q\}$ 一起送入大模型。

已有的少样本学习方法一般在演示样例中给出正面样例,即包含正确的答案,通过正面案例可以引导 LLM 理解问题并进行正确的推理。但是,很多情况下,负面样例也具有较大价值。例如,在实际生活中的教学中,老师除了会讲解一些例题外(相当于

正面样例),还会展示一些典型的错误答案,让学生进行对比。通过正负两方面的例子让学生更深入地理解知识点。受此启发,本文在少样本学习中,将正面样例和负面样例相结合,提出正负例思维链,为每个样本选择最有效的正负样例,让 LLM 更好地与用户意图对齐,以产生更好的结果。

图 1 展示了一个正负例思维链的例子。图 1 顶部为正负例思维链,其中上部左边部分为选出的正例(正例思维链),上部中间和右边部分(带虚线框)为选出的负例(负例思维链),正负例思维链均包括问题 Q 、表格 T 、文本 E 与推理链 w_i (Python 代码形式)。中间上半部分为问题样本,包含问题 Q 和上下文(包含文本 E 和表格 T),下半部分为 LLM 推理的过程 G ,底部是结果 A 。

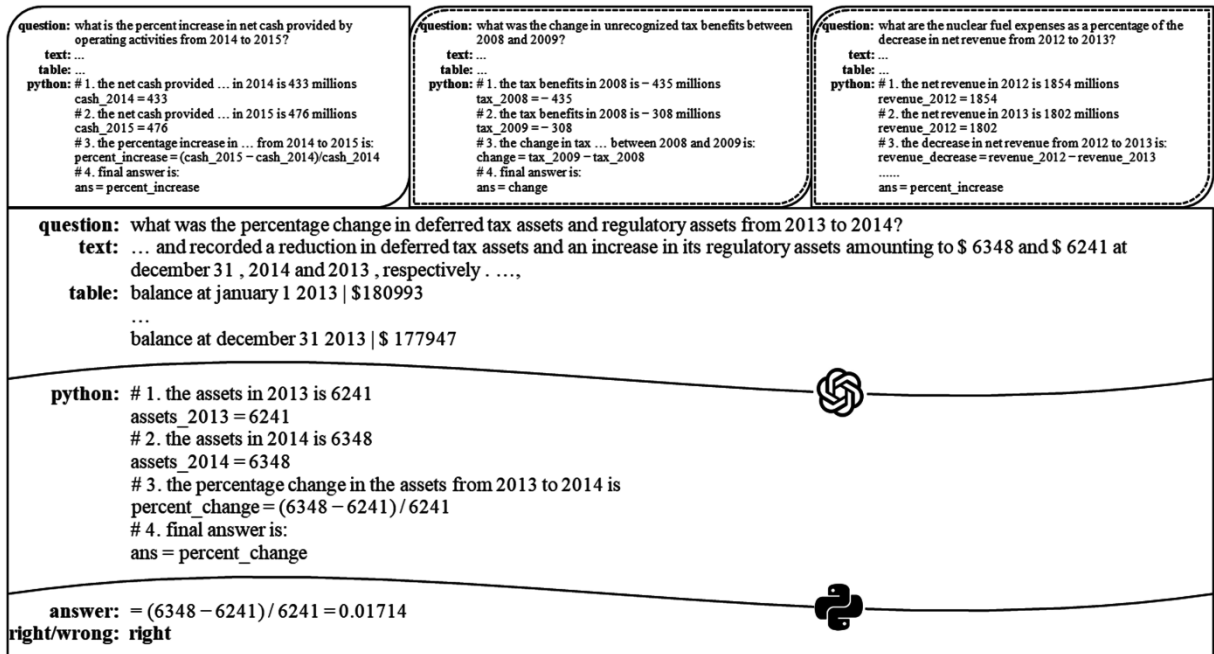


图 1 正负例思维链的一个例子

3 正负例思维链技术

图 2 显示了本文提出的正负例思维链技术的框架。首先根据 LLM 对数据集中已有样本的预测结果构造正例集和负例集,分别进行聚类,得到正例样本池和负例样本池。对于当前问题,从正例样本池中选取最佳正例,构建正例思维链,从负例样本池中选取两个相似负例,构建负例思维链,正负例思维链组成提示(Prompt),提交到 LLM 产生回答。下面分别叙述这一过程。

3.1 正例集和负例集的构造

延续 PoT 方法,在 FinQA 数据集上,将训练集送入大模型(GPT-3.5-Turbo),Zero-Shot 生成问题答案代码,再运行出结果。对结果数据进行分析,LLM 答对的样本归为正例集 C ,LLM 答错的样本归为负例集 F 。用这种方式将样本先粗略分为两大类。

3.2 正负例样本池的构造

从数据集的特点来看,FinQA 中有一些问题是

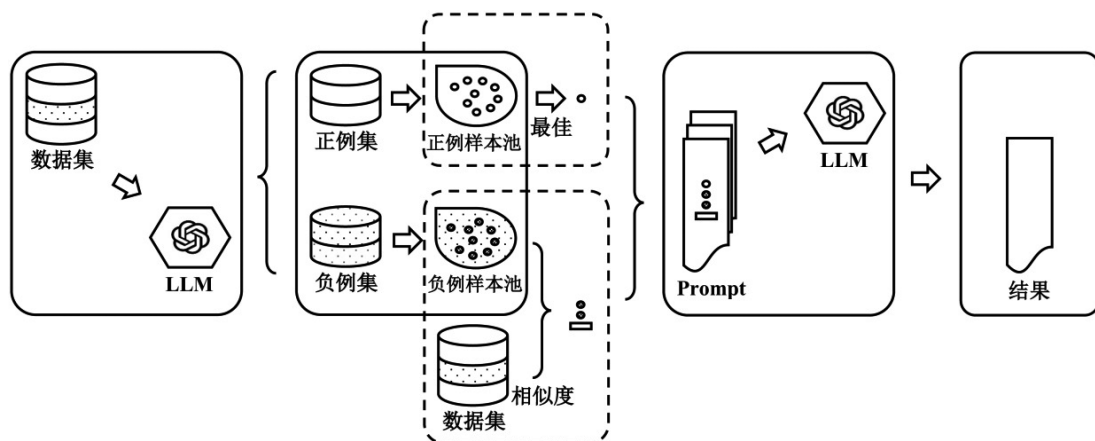


图2 正负例思维链框架

比较难或者比较特殊的,例如,涉及到复杂的计算公式或者特定的金融术语等。这些问题可能导致大模型生成错误或者不完整的答案代码。因此,本文实验设计了通过聚类方法^[24]来区分不同难度或者不同类型的问题。将上面正负例集合,通过 Sentence-BERT^[25]计算问题 Q 中每个问题的向量表示 v_i ,采用 k 均值聚类算法,计算问题之间的余弦相似度,使用轮廓系数来评估聚类效果。轮廓系数是一个介于 $-1 \sim 1$ 之间的值,它反映了一个问题是否被分配到了合适的簇中。轮廓系数越高,表示聚类效果越好。本文实验尝试不同的聚类个数 k 值,计算不同 k 值下的轮廓系数,然后选择轮廓系数最大的那个 k 值作为最佳簇数。

(1) 正例样本池

正例集 C 中,对问题 Q 进行聚类。通过尝试不同的 k 值,实验发现当 $k=9$ 时,轮廓系数达到了最大值,因此选出的最佳簇数是 9。生成 9 个问题类 $C_1, C_2, \dots, C_9$ 。得到 9 类问题,取质心样本 c_i ,这样就得到了 9 个样本的正例样本池 $P_c = \{c_1, c_2, \dots, c_9\}$ 。

(2) 负例样本池

负例集 F 中,同样按照上面的方式,进行聚类。实验发现当 $k=10$ 时,轮廓系数达到了最大值,因此得到 10 类问题 $F_1, F_2, \dots, F_{10}$,取质心样本 f_i ,这样就得到了含 10 个样本的负例样本池 $P_f = \{f_1, f_2, \dots, f_{10}\}$ 。

通过上述聚类方法,将同一类型问题聚为一类,取质心样本为代表,构建样本池。演示样本就从这些样本池中选取,缩小了选取范围,也使选取的样本更具有目的性、代表性和可解释性。

为了考察不同聚类数目对于实验的影响,分别

在正例集上聚 3 类~聚 11 类,图 3(a) 是 3 个~11 个质心样例集的选择率(即不同聚类样例在 1 147 个测试集问题中被选择的比例)和对错比(即每个样例答对数量与答错数量的比值),对错比越大,答题的准确率越高。综合图 3(b)和图 3(c)可以看出,当正例集聚 9 类时,7# 样例选择率是 14%,在 9 个样例中排名第 3,对错比排名第 1,接近排名第 2 的 2 倍,表现最好。因此最终设置正例集聚类数目为 9,选择 7# 样例为基础样例。

同样,在负例集上分别聚 3 类~聚 11 类,分别构建 9 个负例样本池,每个样本池内的元素为质心样本。每次实验取 7# 正例和两个相似负例作为正负例思维链。如图 3(d)所示,当负例集聚 10 类时,效果最佳。因此负例集设置聚 10 类。

给测试集中的每个样本分派两个负例,即用检索的方式,在负例样本池里检索与当前问题最近似的两个样例。负例思维链由这两个负例的表格、文本、问题、推理链和答案组合而成。

测试集中的每个样本都分配正例思维链和负例思维链,即正负例思维链。这样做的目的是:选取答对率最高的样例作为基础样例(最佳样例),构建正例思维链,能够指导大模型答对尽量多的问题;选取与样本问题近似的两个负例,构建负例思维链,是为了提供相似问题的解题思路,规避错误推理。

3.3 提示构造

为了指导大模型生成简洁的代码,并避免变量名缺字或不规范的现象,提高代码生成效率,在正负例思维链的 Prompt 部分,我们在样例 Python 代码的每行之前添加了一句注释语句,对变量和公式的含义进行了详细说明。即 Prompt 由四部分组成:

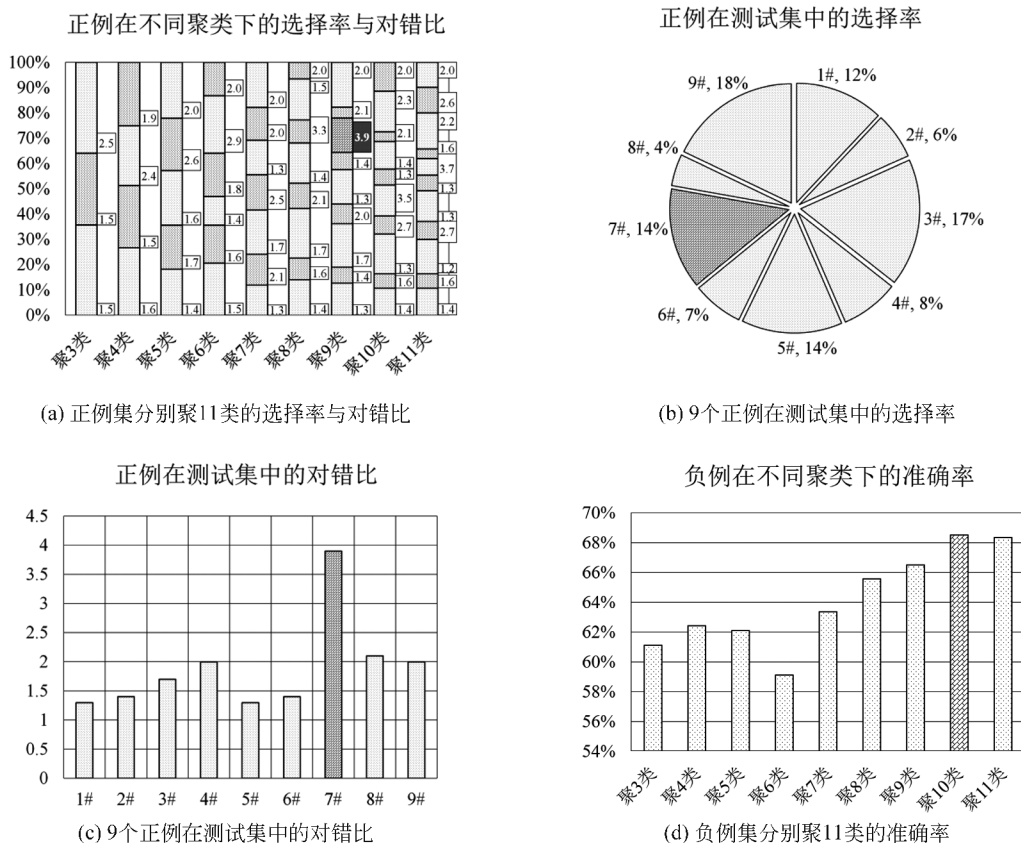


图3 聚类数目对于实验的影响

表格、文本、问题、代码和代码注解。按这种 Prompt 模式输入,大模型能够输出带有注解的 Python 代码。这种自动化的注释生成方法旨在提高代码的可

读性和准确性。

图4的示例是以提示构造方法解决基线实验代码生成错误的情形。

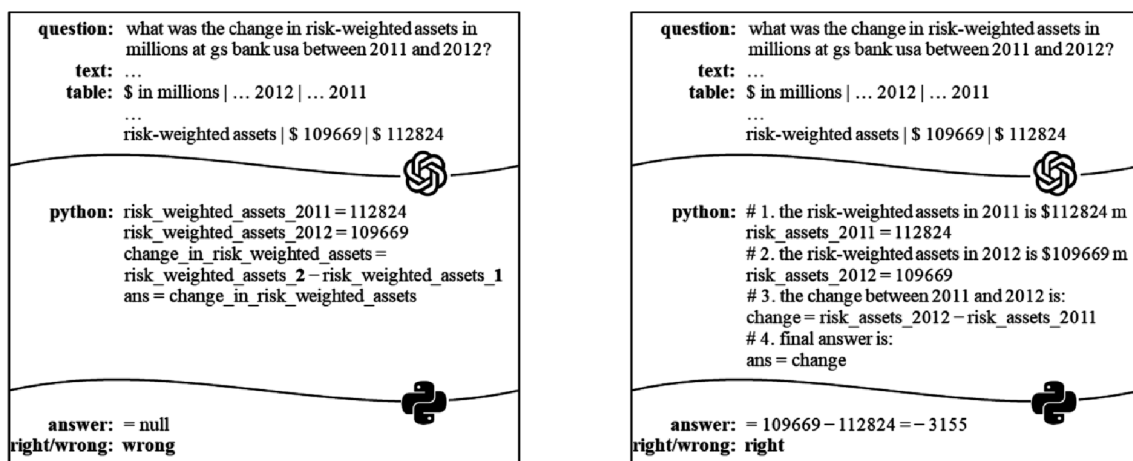


图4 提示构造代码方法解决基线实验代码生成错误

示例中,变量 risk_weighted_assets_2012 与 risk_weighted_assets_2011 被定义之后,在接下来的公式中被写成 risk_weighted_assets_2 与 risk_weighted_assets_1,从而出现变量未定义的情况,致使该段代码无法运行,答案输出为“null”。

将样例中给出代码提示注释后,代码构造生成错误消失,使得该问题回答由错改正。用这样的代码编写样例修改了基线实验(PoT实验),实验效果较原来有少量提高。详细实验结果见文章第4节实验[图7(a)]。

3.4 数据集修订

在对 FinQA 数据集实验后的数据进行分析时,发现数据集本身存在一些原始错误,从而使得部分实验结果本应是正确的,但由于数据集的原始错误而被判为错误。

这些数据集错误情况包括:

(1) 数值计算选择错误公式,解答问题时数值推理未能给出正确公式。

(2) 数值答案的语态理解错误,在解答问题时对“增长”“减少”等词汇的理解不能正确给出答案的正负号。

(3) 数据选择错误,问题所需数据未能从表格正确位置中选出。

(4) 数值单位错误,数值未能正确区分千、百万等单位。

(5) 问题语句错误,给出的问题未能和文本表格内容匹配。

(6) 文本与表格数据不一致,问题所需数据在文本与表格中存在两个不同的数值,从而造成两个不同的答案。

(7) 文本表格噪声,数据集中有大量文本噪声,是 Unicode 字符丢失“\u”,如表格中许多“2014”应是“\u2014”(意为“——”,数据为空),文本中许多“2019s”应该是“\u2019”(意为“/s”),这些噪声影响大模型对文本与表格的阅读。

以上问题属于 FinQA 数据集本身的质量问题,为了让大模型能更好地对样本进行数值推理,对 FinQA 数据集中部分存在质量问题的样本进行了修订,得到修订后的 FinQA-fix 数据集,大模型在修订后的数据集上的数值推理能力有显著提升[图 7(c), 图 7(d)]。

4 实验

4.1 实验设置

模型

本文实验通过 OpenAI API 访问的模型是 GPT-3.5-Turbo。使用贪婪解码器时设置温度系数 $\tau=0.0$ 可保证输出结果的稳定性。

数据集

FinQA^[1] 金融表格文本问答数据集,包含了 8 281 个由金融专家编写的关于财务报告的问答

对。其中,训练集 6 251 个样本、验证集 883 个样本、测试集 1 147 个样本。这些问答对涉及到复杂的数值推理和对异构表示的理解,比如文本和表格。数据集还提供了每个问题的目标推理过程,以便于评估模型的解释能力和逻辑性。

TAT-QA^[2] 也是一个金融表格与文本混合数据问答数据集。其中训练集 13 215 个样本、验证集 1 668 个样本、测试集 1 669 个样本。TAT-QA 数据集样本包括一个半结构化的表格和至少两段相关的段落,这些段落描述、分析或补充了表格内容。数据集答案形式多样:包括单一跨度、多重跨度和自由形式的答案。部分问题需要使用数值推理来回答。

实验中对表格的输入,采用了与 PoT^[8] 相同的方式。将表格线性化为文本字符串。表格中的列用“|”分隔,行用“\n”分隔。如果表格单元格为空,则用“——”填充。对于文本+表格的混合输入,采用“\n”分隔表格和文本。提示由任务指令、文本、线性化表格和问题串联而成。

对 FinQA 数据集的实验数据进行了分析,发现部分原始答案是错误的,导致一些正确的实验结果被误判为错误。错误情况主要有七类:公式错误、语态错误、数据错误、数值单位错误、问题语句错误、文本表格数据不一致、表格错误及文本表格噪声。对有问题的样本进行了修正(详见附录修正细节),修正后的数据集为 FinQA-fix 数据集。大模型在修正后的 FinQA-fix 数据集上的数值推理能力有了显著提升。

评估指标

执行准确率 (Execution Accuracy, Exe-Acc) 是一个衡量模型执行结果与真实结果匹配程度的指标,通常用于评估机器学习或深度学习的模型性能。执行准确率越高,说明模型的预测结果越接近真实结果,模型的性能越好。执行准确率是一种简单直观的评价指标,适用于数据集平衡且各类别同等重要的情况。

基线模型

本文用于对比的基线模型分为基线小模型和基线大模型。其中基线小模型为: FinQANet^[1], 由一个检索器首先从输入的财务报告中检索支持事实,然后由一个生成器生成程序以获得答案;单元检索器和多生成器集成的数值推理问答系统^[26]。基线大模型为 PoT^[8], 使用大语言模型 (Code-Davinci-002) 来生成文本和编程语句,计算委托给程序解释器,并最终给出答案。

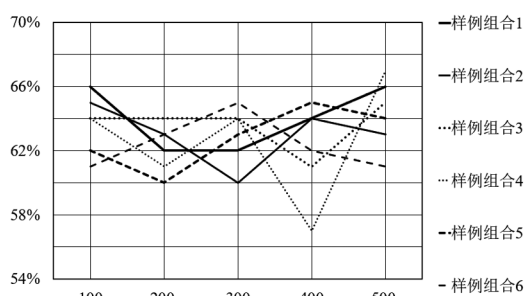
4.2 主要实验

本文复现了 PoT 在 FinQA 数据集上的结果, 使用 GPT-3.5-Turbo, 以此作为基线实验进行对比。FinQANet 的实验结果来自 FinQA 网站^①。单元检索器和多生成器集成的数值推理问答系统来自论文文献[26]。表 1 展示了本文提出的方法以及上面三个基线模型的对比结果。从表 1 中可以看出, 正负例思维链的实验结果最好, 大模型在 FinQA-fix 数据集的实验结果提升显著。这说明本文提出的基于正负例方法来筛选合适的演示样本, 能够提升大模型在表格文本问答数值计算任务中的性能。验证了为不同问题样本筛选合适演示样例, 能够提升大模型解决复杂推理问题能力这一假设。表 1 也显示了本文提出的方法能够在 GPT-4 上有更好的表现。

表 1 FinQA 实验结果

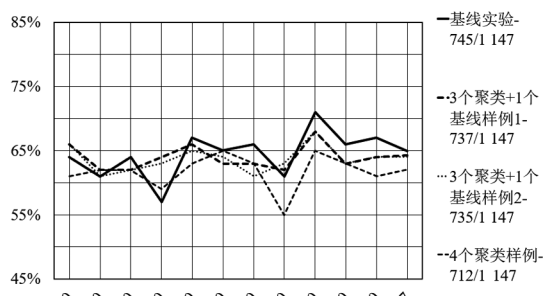
模 型	FinQA ExeAcc	FinQA-fix Exe-Acc
基线小模型: FinQA Net-Roberta-large	61.240	—
基线小模型: 单元检索器和多个生成器集成的数值推理 QA 系统	68.000	—
基线大模型: PoT 论文结果 (code-davinci-002)	64.500	—
基线大模型: PoT 复现论文结果 (GPT-3.5-Turbo)	64.952	76.635

不同样例组合局部实验比较



(a) 6种不同样例组合的局部实验比较

几种不同样例实验与基线实验比较



(b) 几种不同样例实验与基线实验比较

图 5 几种实验结果的比较

(1) 演示样例的内容、数量及顺序, 都能引起实验结果的变化, 因此演示样例的选择, 是影响实验结果的一个关键因素:

① 任何一个样例, 对问题解答都有一定的帮助; 而不同的样例组合, 对大模型解答问题的准确率会有不同的影响。图 5(a) 是选取 6 组不同样例组

4.3 实验分析

4.3.1 演示样例的选择对实验结果的影响

在这一部分的实验中, 尝试将聚类 (2.1 节所述聚类方法) 得到的样本与原始 PoT 实验中的四个演示样本排列组合, 考察各种演示样本组合对大模型输出效果的影响。在这一部分提到的基线实验数据是复现论文 PoT 实验 (GPT-3.5-Turbo) 的结果。用聚类方法先对训练集按问题聚为五类 (在该阶段的实验中, 由于 FinQA 数据集规模较大, GPT 输入受限, 5 个及 5 个以上的样例组合会超出 GPT 的 Token 限制, 所以该阶段的样例组合都控制在 4 个样例以内), 取质心样本, 得到 5 个样例, 在对样例进行一定的比较实验后, 选出 4 个样例进行实验, 得到的准确率是 62.075%。在对聚类实验数据进行分析时, 发现以下情况: 图 5 中, 横坐标是样本编号, 纵坐标是执行准确率, 图中记录了从 100 ~ 1100 每 100 个样本的执行准确率, 以及最终 1147 个样本的总准确率, 图例中的数字 737/1147 则表示 1147 个样本中, 执行准确的样本数量是 737 个。

合的局部实验比较。这 6 组样例组合, 是从聚类出来的 5 个样例和 PoT 论文中的 4 个样例中选取出来的。从图 5 能够看出, 这些不同组合的演示样例, 对大模型做数值推理任务的效果有较明显的影响。

① <https://github.com/czyssrs/FinQA>

② 相同样例的组合, 对大模型解答问题的准确率也会有较小概率的随机差异。图 5(b) 中, 有相同样例组合的两次实验比较。这两个实验采用了相同的样例组合 (3 个聚类样例和 1 个基线样例), 两次实验有少量的随机差异。

③ 存在一些优异的样例, 对大模型解答问题的准确率有较大帮助。

图 5(b) 中, 在 4 个聚类样例实验 (准确率为 62.075%) 的基础上, 替换进 1 个基线实验样例 (3 个聚类样例和 1 个基线样例), 准确率为 64.255%, 提升了 2.18 个百分点, 可以看出替换进的这个样例对实验结果的影响比较大。

将该实验和基线实验进行比较。这次替换 1 个基线样例后的实验结果已较为接近基线实验。因此, 寻找表现突出的样例, 作为本文的实验目标。出于这个目标, 在实验中, 从正例样本池中选出了最佳样例 (7# 样例)。

(2) 对某些大模型答错的题目, 如果能给出相应的样例, 在这类问题上使大模型更容易给出正确的解答; 但同时, 大模型也会将另外一些原本能回答正确的题目给出错误答案。

如图 6(a) 所示, 该样本无论是在 PoT 演示样例下还是在聚类演示样例下, 大模型回答都是错误

的。如图 6(b) 所示, 将该样本单独取出, 在大模型迭代环境下进行分析, 并不断提示修正, 最开始大模型未能在文本与表格中找到有关数据, 直接回答“所提供文本不包含……信息。因此, 无法确定……未租赁面积占总面积的百分比”; 在提示“相关数据在文本和表格中寻找”后, 大模型给出信息如下: “办公面积 254 000 平方英尺, 租赁面积未说明, 未租赁面积占总面积的 100%”; 直接提示: “除了……165 000 平方英尺外”, 大模型依然给出错误答案: “165 000 为租赁面积, 254 000 为总面积, $(254\,000 - 165\,000) / 254\,000 \times 100$ 为最终答案计算公式”。于是在训练集中搜索了 1 个与之相近似问题的样本, 制作成演示样例 [图 6(c) 中带虚线部分], 与该样本一道送入大模型后, 能够给出该样本的正确答案。但添加这个样例后, 总体实验准确度降低了。针对这一现象, 从负例样本池中, 为每个问题样本分配与之相似的两个负例。

(3) 大模型在模仿样例编写 Python 代码的过程中, 变量名称会出现随机缺字情况, 从而造成实验差错, Prompt 部分按照 3.4 节所述方式修改了基线实验, 实验结果较原来有少量提高。图 7(a) 在基线实验的基础上, 对样例的代码编写格式做了改进, 实验性能有少量提升, 准确率从 64.255% 提升到 65.562%。

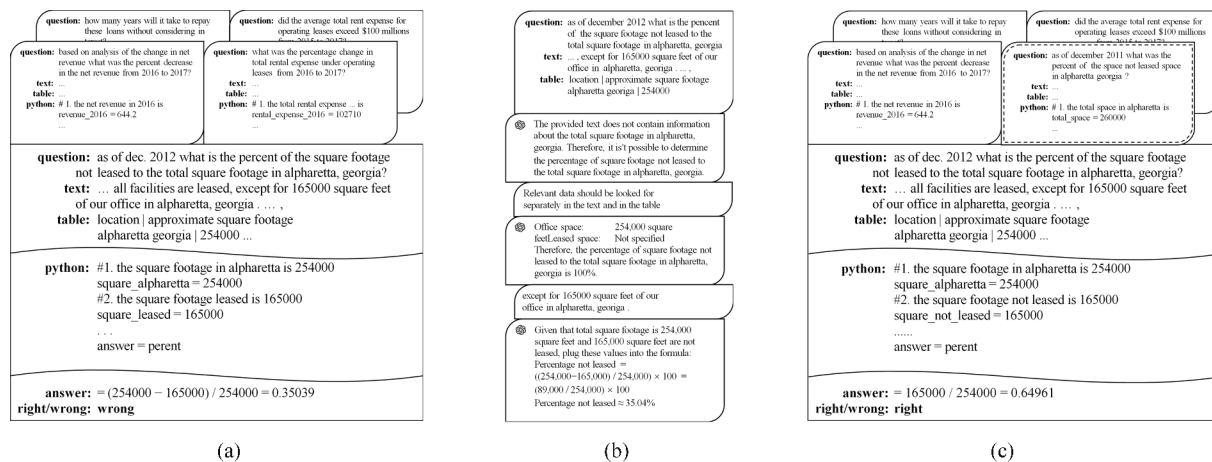


图 6 更换近似样例后, 纠正答题错误

4.3.2 正负例思维链实验结果分析

分别自动给测试集中的每个样本分派一个 7# 样例 (正例) 和两个相似负例。最终实验准确率是 68.527%; 比原始模型的 64.952% 要高出 3.58 个百分点。图 7(a) 中也给出了基线实验 (PoT in GPT-3.5-Turbo) 与本文实验结果比较, 基线实验用了四个演示样例, 而本文实验只用了三个演示样例, 相比之下更节省成本。在第 2 份 100 个样本集上, 本文

提出的方法比基线实验提高了 11 个百分点 (61%~72%), 在第 4 份 100 个样本集上, 本文提出的方法提高了 9 个百分点 (57%~66%), 在第 7 份 100 个样本集上, 本文提出的方法提高了 8 个百分点 (66%~74%), 在第 11 份 100 个样本集上, 本文提出的方法提高了 7 个百分点 (67%~74%)。

4.3.3 修订后数据集结果分析

针对数据集的问题, 对 FinQA 数据集进行修订, 修订后重新完成 PoT 基线实验和正负例思维链

实验,得到的实验结果是:76.635%和77.681%。基线实验结果提升了12.38个百分点,正负例思维链实验结果提升了9.15个百分点(图7(b))。数据

集修订后,本文实验仍高于基准实验1.05个百分点。由此可见,高质量的数据集对于大模型的推理能力的影响是非常显著的。

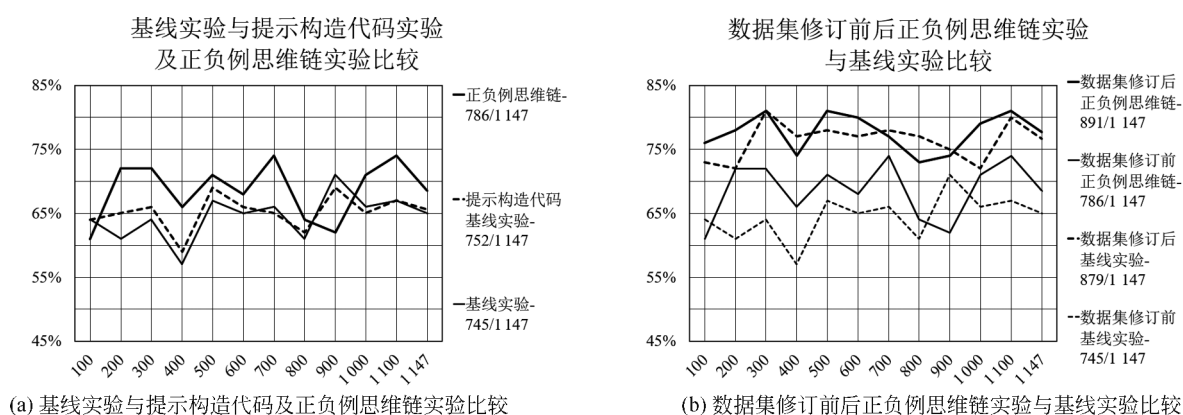


图7 正负例思维链方法实验与基线实验及数据集修订前后的实验结果比较

4.3.4 消融实验

为了对比不同演示样本对大模型数值推理效果的影响,本文实验进行了如表2所示的消融实验。从表2可以看出,只取一个演示样本时,无论是只取相似正例还是取最佳正例(7#样例),其准确率都要高于只取一个负例的情况,说明正例的选择能够保障大模型做对大部分的题目。选取两个演示样本时,取一个固定最佳正例(7#样例)和一个相似正例,效果比只取一个最佳正例要差,说明演示样本的加入也会降低大模型的推理效果,而加入相似负例时,效果明显提升,说明负例发挥了指导作用。由于FinQA数据集规模较大,GPT输入Tokens受限(16384个Tokens),超过4个样本就会被限制输入。而过多的演示样本也会给大模型带来误导,所以合适的样本内容以及合适的样本数量是影响大模型推理性能的关键因素。

表2 消融实验结果

选择演示样本的策略	FinQA-fix Exe-Acc
只取一个相似正例	73.496
只取一个相似负例	72.624
只取一个固定最佳正例(7#样例)	74.106
一个固定最佳正例和一个相似正例	73.758
一个固定最佳正例和一个相似负例	76.199
一个固定最佳正例和两个相似负例	77.681

4.4 其他方法的实验效果

文本也在FinQA数据集上尝试过其他方法,但效果都不理想。以下为实验操作、实验结果与实

验分析:

(1) 表格转文本:将表格用模板转换为文本,再送入大模型,在FinQA数据集上得到的结果是61.552%,低于基线模型。说明大模型对原始表格的理解能力很强,传统的Table-to-Text方法割裂了表格与文本的关联,未来将尝试其他表格与文本的相互转换方式。

(2) 自我检验^[18]:为了考察大语言模型的自行检查代码能力,在实验中将生成的代码送回大语言模型,设计提示语句,让大语言模型自己判断对错,如果错误则输出新的代码。在原始FinQA数据集上的准确度是62.685%。观察发现,大模型在Zero-Shot情况下,经常把做对的题目改错,说明在判断对错方面,仍然需要寻找一些典型的Few-Shots作为参照依据。

(3) Plan-and-Solve方法^[15]:先用大模型生成Plan,检索支持事实,再将Plan与支持事实送入大模型得到结果,在原始数据集上得到的准确率是50.828%,准确率低的原因是大模型无法将Plan与支持事实的语义相关联,从而无法输出正确结果。

(4) 子问题分解法^[21]:先用大模型将原始问题分解为2~3个子问题,子问题解答并回填,根据子问题答案生成最后答案,其准确率是54.141%,观察发现生成Plan的错误率也很高,说明这种方法不适合表文混合的数值计算问题。

这一部分实验中的自我检验、Plan-and-Solve方法和子问题分解方法分别来自MWP和表格实验领域,这些方法在这些领域的数据集上均获得了显著性能提升,但在金融领域表文混合问答数值推理任务上效果不佳。究其原因是一些方法都没有充

分考虑到表格和文本之间的关联性和互补性,没有利用表格和文本的结构化信息和语义信息,尤其是金融领域的专业知识。因此,这些方法都不适合金融领域表文问答的数值推理任务。本文提出的正负例思维链方法,在正例思维链和负例思维链的 Prompt 部分,向大模型展示了样本的表格、文本、问题、推理链和答案等信息,并用注释语句对变量和公式进行说明。这些信息都是出自同一份财务报告,按照一定的逻辑和规则排列组合在一起,保留了表格的结构特性,避免将表格与文本割裂,从而提高了大模型对表格和文本之间关联性和互补性的理解能力,提升了大语言模型在金融领域自动问答任务中的多步骤推理能力。

4.5 TAT-QA 数据集结果

为验证本文方法的有效性,在 TAT-QA 数据集上做了正负例思维链实验。TAT-QA 数据集的问题类型比较特殊,具体分为:跨度(Span)、多跨度(Multi-spans)、运算(Arithmetic)、计数(Count)。实验发现,Span 类型的准确率很低,这是因为大模型回答这类问题时,随机性很高,有时截取字符串,有时截取整句回答;而数据集中这类问题的答案也同样如此,因此答案判错较多。

和 FinQA 一样,将 TAT-QA 的训练集送入大模型(GPT-3.5-Turbo),Zero-Shot 生成问题答案代码,再运行出结果。对结果数据进行分析,将 LLM 答对的样本归为正例集,将 LLM 答错的样本归为负例集。由于 TAT-QA 数据集问题类型已经分为 Span, Multi-spans, Arithmetic 和 Count 这四类,这四类分别占比 42%,13%,43%和 2%。将正例集按上述实验方法聚为四类,每类取质心样本,这样就得到了 4 个正例样本,用以构建正例样本池(正例思维链)。分析负例集发现,Span 类答错率为 44%,Multi-spans 类为 35%,Arithmetic 类为 25%,Count 类为 16%。Count 类占总题数的 2%,答错率也最低,只有 5 道答错,因此将负例集分为三类,按照比例选取质心样本:Span 类选 7 个,Multi-Spans 类选 4 个,Arithmetic 类选 4 个,这样得到负例样本池(共计 15 个样本),验证集样本在这个负例样本池内选取两个最相似负例样本作为负例思维链。实验结果如表 3 所示。在该实验环节,PoT 实验 Few-Shots 的样本个数为 8 个,本文实验 Few-Shots 的样本个数为 6 个(4 个固定正例思维链和两个最相似负例思维链)。

表 3 TAT-QA 实验结果

模型	TAT-QA Exe-Acc
基线小模型: TAGOP	55.20
基线大模型: PoT 复现结果(GPT-3.5-Turbo)	65.71
基线大模型: 正负例思维链(GPT-3.5-Turbo)	66.61
基线大模型: 正负例思维链(GPT-4.0)	72.12

5 总结

本文研究了思维链(CoT)在金融领域自动问答任务中的应用。通过实验发现,演示样例的内容、数量和顺序都会造成实验结果的变化,因此构建有效的演示样例成为影响实验结果的一个关键因素。本文提出的正负例思维链,借鉴学生做对题目和做错题目对学习效果的不同影响,从大模型的推理结果中筛选出对题和错题样本,形成正例样本池和负例样本池。采用静态和动态相结合的策略,选取不同类型的演示样本,构建正负例思维链,提高大模型的推理能力。本文还修订了 FinQA 数据集,提高了数据集质量。实验结果表明,本文提出的方法在 FinQA 数据集上准确率提高了 3.6%,在 FinQA-fix 数据集上准确率提高了 12.73%,显著提高了大模型的数值推理能力。本文的研究为大模型在金融领域数值推理任务提供了一种有效的思维链构建方法,为未来的 Few-Shots 研究提供了新的思路。

参考文献

- [1] CHEN Z, CHEN W, SMILEY C, et al. FinQA: A dataset of numerical reasoning over financial data[C]//Proceedings of EMNLP, 2021: 3697-3711.
- [2] ZHU F, LEI W, HUANG Y, et al. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance [C]//Proceedings of ACL, 2021: 3277-3287.
- [3] LEI F, HE S, LI X, et al. Answering numerical reasoning questions in table-text hybrid contents with graph-based encoder and tree-based decoder [C]//Proceedings of COLING, 2022: 1379-1390.
- [4] WEI J, WANG X, SCHUURANS D, et al. Chain-of-Thought prompting elicits reasoning in large language models[C]//Proceedings of NeurIPS, 2022: 24824-24837.
- [5] BROWN T B, MANN B, RYDER N, et al. Language models are Few-Shot learners [C]//Proceedings of NeurIPS, 2020: 1877-1901.
- [6] CHEN M, TWOREK J, JUN H, et al. Evaluating

- large language models trained on code [J/OL]. arXiv preprint. arXiv: 2107.03374, 2021.
- [7] CHOWDHURY A, NARANG S, DEVLIN J, et al. PaLM: Scaling language modeling with pathways [J]. Journal of Machine Learning Research, 2023, 24 (240): 1-113.
- [8] CHEN W, MA X, WANG X, et al. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks [J/OL]. arXiv preprint. arXiv: 2211.12588, 2022.
- [9] XIE S M, RAGHUNATHAN A, LIANG P, et al. An explanation of in-context learning as implicit bayesian inference [J/OL]. arXiv preprint. arXiv: 2111.02080, 2021.
- [10] MIN S, LYU X, HOKTZMAN A, et al. Rethinking the role of demonstrations: What makes in-context learning work? [C]//Proceedings of EMNLP, 2022: 11048-11064.
- [11] KOJIMA T, GU S S, REID M, et al. Large language models are Zero-Shot reasoners [J/OL]. arXiv preprint. arXiv: 2205.11916, 2022.
- [12] DROZDOV A, SCHÄRLI N, AKYÜRE E, et al. Compositional semantic parsing with large language models[J/OL]. arXiv preprint. arXiv: 2209.15003, 2022.
- [13] ZHOU D, SCHÄRLI N, HOU L, et al. Least-to-Most prompting enables complex reasoning in large language models[J/OL]. arXiv preprint. arXiv: 2205.10625, 2022.
- [14] GAO L, MADAAN A, ZHOU S, et al. PAL: Program-aided language models[C]//Proceedings of ICML, 2023: 10764-10799.
- [15] WANG L, XU W, LAN Y, et al. Plan-and-solve prompting: improving zero-shot chain-of-thought reasoning by large language models[C]//Proceedings of ACL, 2023: 2609-2634.
- [16] LYU Q, HAVALDAR S, STEIN A, et al. Faithful chain- of-thought reasoning [J/OL]. arXiv preprint. arXiv: 2301.13379, 2023
- [17] WANG X, WEI J, SCHUURMANS D, et al. Self-consistency improves chain of thought reasoning in language models[J/OL]. arXiv preprint. arXiv: 2203.11171, 2022.
- [18] ZHENG C, LIU Z, XIE, et al. Progressive-Hint prompting improves reasoning in large language models[J/OL]. arXiv preprint. arXiv: 2304.09797, 2023.
- [19] CHEN W, ZHA H, CHEN Z, et al. HybridQA: A dataset of multi-hop question answering over tabular and textual data[C]//Proceedings of EMNLP, 2020: 1026-1036.
- [20] YE Y, HUI B, YANG M, et al. Large language models are versatile decomposers: Decompose evidence and questions for table-based reasoning [C]//Proceedings of SIGIR, 2023: 174-184.
- [21] RADHAKRISHNAN A, NGUYEN K, CHEN A, et al. Question decomposition improves the faithfulness of model-generated reasoning [J/OL]. arXiv preprint. arXiv: 2307.11768, 2023.
- [22] ZHA L, ZHOU J, LI L, et al. TableGPT: Towards unifying tables, nature language and commands into one GPT[J/OL]. arXiv preprint. arXiv: 2307.08674, 2023.
- [23] WEI J, WEI J, TAY Y, et al. Larger language models do In-Context learning differently [J/OL]. arXiv preprint. arXiv: 2303.03846, 2023.
- [24] ZHANG Z, ZHANG A, LI M, et al. Automatic chain of thought prompting in large language models [J/OL]. arXiv preprint. arXiv: 2210.03493, 2022.
- [25] REIMERS N, GUREVYCH I. Sentence-BERT: Sentence embeddings using siamese BERT-networks [C]//Proceedings of EMNLP, 2019: 3980-3990.
- [26] WANG B, JU J, MAO Y, et al. A numerical reasoning question answering system with fine-grained retriever and the ensemble of multiple generators for FinQA[J/OL]. arXiv preprint. arXiv: 2206.08506, 2022.



李希 (1982—), 博士生在读, 主要研究领域为文本挖掘。

E-mail: yunjuyan2021@163.com



舒晴 (1990—), 博士生在读, 主要研究领域为机器学习、数据挖掘。

E-mail: sq0124@gmail.com



刘喜平 (1980—), 通信作者, 博士, 教授, 博士生导师, 主要研究领域为大数据分析、文本挖掘。

E-mail: liuxiping@jxufe.edu.cn