成　绩	
评阅人	

复 旦 大 学

研 究 生 课 程 论 文

论文题目：　增强大模型指令遵循能力的数据挖掘

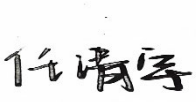
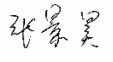
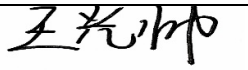
修读课程：　大数据挖掘（COMP620051）

选课学期：　2024-2025 学年第一学期

选课学生：　任清宇（24210240282），张景昊（24210240404），
王光帅（24110240080），吴师师（24112030038）

完成日期：　2024 年 12 月 25 日

分工

学号	姓名	分工	签名
24210240282	任清宇	**技术方案，运行实验（34%）**	
24210240404	**张景昊**	**运行实验，分析结果（22%）**	
24110240080	王光帅	**相关工作调研（22%）**	
24112030038	吴师师	**背景调研（22%）**	

第一章 背景与意义

1. 背景

近年来，基于深度学习的自然语言处理技术在多个领域取得了显著的进展，其中**大语言模型**是其中最具代表性的技术之一 [1]。大语言模型是指那些基于海量数据进行预训练的超大型深度学习模型。这些模型通过对海量语料进行学习，能够捕捉语言的复杂规律和语境关系，处理多种复杂的语言任务 [2]。经典的大语言模型包括 OpenAI 的 GPT 系列等，这些 LLMs 在文本生成、代码生成等多项自然语言处理任务中均展示了强大的性能。

LLMs 准确理解指令并传达期望输出的能力，称为指令遵循能力 [3]。对于 LLMs 来说，准确遵循具有详细要求的指令尤为重要 [4][5]

在研究中，指令遵循能力的任务复杂度通常是通过约束的数量和种类来定义的。单一约束指令通常要求模型遵循一个明确的要求，例如“回答中必须包含 mutations 这个单词”。这些任务一般都非常简单，任务目标清晰明确，因此，语言模型通常能够高效且准确地遵循这些简单指令，并生成符合预期的输出。而多约束指令则包含多个约束条件，这些约束可以是格式上的（如字数、大小写等）、内容上的（如必须包含特定关键词）、结构上的（如必须遵循某种语法规则）等。例如，一个多约束指令可以是：“请生成一个包含 mutations 这个单词的 200 字以内的文本，要求所有字母均为大写，且文本末尾必须包含 END。”。在这种情况下，LLMs 不仅要遵循多个约束，还要确保这些约束在生成的文本中协调一致，这使得任务变得更加复杂和困难，对模型指令遵循能力的提出了更高的挑战 [6][7]。尽管目前 LLMs 在遵循单一约束指令方面取得了显著进展，但在面对复杂的多约束指令时，LLMs 的表现仍然存在明显的不足，事实上许多模型往往会产生不符合所有硬性约束条件的输出，导致输出不符合预期 [8][9]。

目前，复杂指令的研究主要集中在评估，缺乏针对如何提高 LLMs 在多约束指令任务中的能力的有效方法 [10][11]。现有的改进方法主要关注少量约束指令的任务，未能有效展示实际应用中的复杂性 [12][13]。虽然已有一些研究通过构造包含多约束的复杂指令并对 LLMs 进行微调 [14]，但仍然有一个关键问题未得到充分探讨，即什么样的训练数据能够有效提模型的多约束指令的遵循能

力？而这引出了两个后续问题：如何获取有效的训练数据以及如何有效地利用这些训练数据。

2. 意义

本研究发现，比起单一约束指令，多约束训练数据集能够帮助 LLMs 更好地理解和处理涉及多个约束条件的复杂指令，进而提升其执行任务的准确性和鲁棒性。基于这一发现，我们成功构建了一个高质量的多约束训练数据集，并通过一系列训练方法（学生-教师模型修正策略和 DPO 训练）进一步增强了模型的指令遵循能力。试验证明，我们的高质量多约束训练数据集和训练方法在保持大语言模型通用能力的同时，能够显著提升其在不同环境下遵循复杂指令的能力。特别是在面对涉及多种约束（如格式要求、语法规则、内容限制等）的实际应用时，LLMs 的表现得到了显著优化。这为如何提高大语言模型在多约束指令下的执行能力提供了见解，并为大语言模型在更广泛的实际应用中提供了更强的支持。

3. 研究内容

我们围绕上述三个关键问题进行了探索。我们通过试验研究探讨了什么样的训练数据更能增强模型对于复杂指令的理解。试验结果表明，包含多个约束的指令比单一约束的指令能够更有效地提升 LLMs 的指令遵循能力。基于这一发现，为了获得高质量的训练数据集，我们首先通过学生模型（LLaMA2）生成初始输出。接着，我们通过测试程序识别出模型未能遵循的约束，并使用教师模型（ChatGPT）对学生模型的输出进行逐一修正。最终，我们利用教师模型生成的最终输出构建正样本集合，利用教师模型修正的中间结果构建负样本集合，并使用这些数据构建的偏好数据集进行 DPO 训练挖掘数据。总体而言，我们构建了高质量的多约束训练数据集，在保持 LLMs 的通用能力的前提下，进一步提高了 LLMs 的指令遵循能力。

第二章 相关工作

与传统以样本为驱动的监督学习相比，指令跟随有着截然不同的特性。传统监督学习主要依靠大量带标注的样本，让模型学习固定的输入输出模式。而指令跟随的本质是训练大语言模型理解不同的指令并且产生相对应的回复，比如让它写一篇包含限定单词和数量的短文，模型理解指令的意思，然后输出对应的内容。由于它在下游的 unseen 任务中很强的能力，指令跟随已经成为解决 few/zero-shot 任务的范式[15,16]。现在有许多的工作去验证 LLM 在指令跟随任务上的表现，有些工作关注当干扰答案空间时模型是否还能理解指令[17]，另一些工作在指令中包含可验证的约束（如词汇、数字和格式）。指令跟随效果的能力高度依赖于模型和任务的规模：一个更大的模型或者说包含更多 token 的模型在更复杂的任务上训练可以在下游的 few/zero-shot 任务上取得显著的效果[18]。但对于大多数人来说，放大模型的规模是不现实的，因为这需要大量的计算资源，因此现在许多的研究是通过人力收集或者通过蒸馏大模型的知识来构建多种任务的高质量的数据。

指令遵循的本质在于通过遵循各种任务指令并以相应的期望输出进行回应，来训练模型。指令微调数据集（高质量的指令输出对）就成为了至关重要的部分。按照不同的标注类别，当前的指令微调数据集可分为两类：1）人工标注数据集；2）由大型语言模型（LLM）生成的合成数据集：人工创建的数据集大多质量较高（标注错误极少），但需要耗费大量人力，且耗时良久。更为重要的是，人类存在多样性受限的问题——要想出多种多样且新颖的任务对人类来说着实颇具挑战性；因此，人工标注数据集的任务规模通常会受到人工标注员的限制（例如，人类标注员的专业水平以及协作方式等因素）。由于大型语言模型（LLM）已在各类自然语言处理任务中展现出了卓越的标注质量，近期大量的研究工作试图在指令微调数据集的整理工作中使用大型语言模型（例如 ChatGPT 和 GPT-4）来取代人工。尽管这些由大型语言模型生成的合成数据集包含大量噪声（例如，指令不连贯以及输出存在幻觉），但其多样的任务分布和模型偏好的输出模式仍然有利于较小模型的指令遵循，与人工标注数据集相比，甚至能实现相当或更好的泛化性能。总之，在人工标注数据集和由大型语言模型生成的合成数据集之间进行选择，也可以看作是在数据质量和多样性之间进行权衡。先前的研究已经得出结论，这两个因素都会影响最终模型的性能混合使用人工和机器生成的数据可能会带来更好的结果，然而，关于哪个因素更为重要并没有具体定论，这在很大程度上取决于下游任务和应用场景。

复杂指令可以指涉及更多推理步骤、复杂输入或多重约束的指令。复杂问题的处理能力决定着这一代 AI 技术的上限，模型能够更好地理解和执行复杂指令，

才能在各个方面提升其价值和应用潜力。许多研究已经表明，使用复杂指令进行微调能够提升诸如指令遵循、推理或代码生成等任务的表现，但这种复杂性的任务给 LLM 的能力提出了巨大的挑战。

在这样的背景下，一个名为 ComplexBench[19]的新基准应运而生。这个基准旨在全面评估 LLMs 遵循复杂指令的能力，特别是那些包含多重约束组合的指令。ComplexBench 的提出源于研究者们的一个重要发现：现有的评估基准主要关注人类指令中不同类型的约束建模，却忽视了不同约束之间的组合，而这恰恰是复杂指令中不可或缺的组成部分。ComplexBench 的创新之处在于，它不仅提出了一个分层的复杂指令分类体系，还手动构建了一个高质量的数据集，并设计了一种新颖的自动评估方法。这种方法巧妙地整合了基于规则和基于 LLM 的评估策略，从而提高了评估复杂指令跟随能力的准确性。通过对多个代表性 LLMs 的实验，ComplexBench 揭示了当前模型在处理复杂指令时存在的显著不足，特别是在面对复杂组合类型时的表现。

微软的研究人员提出了一种名为 Evol-Instruct[20]的方法，通过使用 LLM 自动大规模生成不同难度水平的开放域指令。使用生成的指令数据微调的 7B LLaMA 模型，在 29 项技能中有 17 项达到了 ChatGPT 的 90%以上性能；甚至在复杂问题的测试中，优于 ChatGPT。它从人工编写的一些简单指令开始，随机选择基于深度来升级简单指令为更复杂的指令；或基于广度来创建同领域新指令：深度(In-depth Evolving) 包括五种操作：添加约束条件、加深理解、具体化、增加推理步骤、复杂化输入；广度(In-breadth Evolving)：基于给定指令生成一个同领域的新指令。通过一个迭代过程重复多轮，以获得足够包含各种复杂性的指令数据。最后对生成指令的多样性和有效性进行过滤，得到最终复杂指令数据集

AUTOIF[21]的作者指出大型语言模型（LLMs）的一个核心能力是遵循自然语言指令，但目前尚未解决如何自动构建高质量的训练数据以增强 LLMs 复杂指令遵循能力的问题。作者提出了 AUTOIF，这是第一个可扩展和可靠的方法，用于自动生成指令遵循训练数据。AUTOIF 将指令遵循数据质量的验证转化为代码验证，要求 LLMs 生成指令、相应的代码以检查指令响应的正确性，以及单元测试样本以验证代码的正确性。利用执行反馈的拒绝采样可以为监督式微调（SFT）和基于人类反馈的强化学习（RLHF）训练生成数据。AUTOIF 在三种训练算法（SFT、离线 DPO 和在线 DPO）上取得了显著改进，并且已经应用于顶级开源 LLMs，Qwen2 和 LLaMA3，在自我对齐和强到弱的蒸馏设置中。

第三章 实验方法

1. 数据集

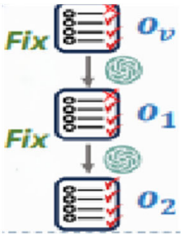
1) 数据获取

为了确保种子指令的覆盖面和多样性，我们从以下三个数据集收集 1500 条指令：

- Open Assistant：Open Assistant 数据集包含了人类与聊天机器人互动时编写的指令。这些数据被用来训练和优化聊天机器人的性能，使其更自然地与人类进行交流。只收集了排名 0 的指令，即被标注为最高质量的指令，以及对话的第一回合。
- Self-Instruct：Self-Instruct 数据集覆盖了各种不同的主题。这些指令被设计用来辅助新任务的指令生成，帮助构建更加全面和多样化的指令集。这个数据集有助于在缺乏足够用户指令的情况下，为新任务生成有效的指令。
- Super-Natural：Super-Natural 数据集包含了格式化为人类指令的自然语言处理任务。排除简单的任务，如分类和标记。这个数据集通过提供具体的人类指令，帮助研究者理解如何将 NLP 任务转化为可执行的指令，以及如何通过这些指令来训练和评估模型。

2) 数据预处理

- 从多个来源收集初始指令。
- 使用学生模型（LLaMA2）生成初始输出。
- 使用测试程序来识别模型未能遵循的约束。
- 使用教师模型（ChatGPT）逐一纠正这些错误。
- 修改过程中的数据用于后续训练。



具体例子如下：

- seed_data.jsonl：种子指令数据集， 1500 条

{"**instruction**": "In this task, you will be given a sentence. You need to

reconize the name of the disorder or disease. Disease is a disorder of structure or function in a human, animal, or plant, especially one that produces specific symptoms or that affects a specific location and is not simply a direct result of physical injury. Although there might be several correct answers, you need to write one of them. \n", "**input**": "input : In individuals with mutations in either region 2 or region 3 , the average number of adenomas tended to be lower than those in individuals with mutations in region 1 , although age at diagnosis was similar . ", "**output**" :"adenomas"}

- data.jsonl: 根据种子数据集构造的多约束数据集 1500 条 用于推理

  {"prompt": "In this task, you will be given a sentence. You need to ….. “," input " : " In individuals with……. **end it with a post script starting with P.S.**. **The response must contain at least 1 placeholders (i.e., [restaurant])**. **Make sure to include the words mutations**. **response without any commas**. **Make sure to only use capital letters**","instruction_id_list":["detectable_content:postscript","detectable_content:number_placeholders","keywords:existence","punctuation:no_comma","change_case:english_capital"], "kwargs": [{"postscript_marker": "P.S."}, {"num_placeholders": 1}, {"keywords": ["mutations"]}, {}, {}],"**constraints**": ["end it with a post script starting with P.S.", "The response must contain at least 1 placeholders (i.e., [restaurant])", "Make sure to include the words mutations", "response without any commas", "Make sure to only use capital letters in your entire response"]}

- checked.jsonl: 检查模型的回复是否遵循约束 1500 条

  {"constraints": ["end it with a post script starting with P.S.", "The response must contain at least 1 placeholders (i.e., [restaurant])", ……..], "**follow_all_instructions**": false, "**follow_instruction_list**": [true, false, false,false,false],"instruction_id_list":["detectable_content:postscript","detectable_content:number_placeholders","keywords:existence","punctuation:no_comma","change_case:english_capital"],"kwargs":[{"postscript_marker": "P.S."}, {"num_placeholders": 1}, {"keywords": ["mutations"]}, {}, {}],"prompt": "In this task, you will be given a sentence. You need to……….. \n", “input" : "In individuals with mutations in either region 2 or region 3 , ……","response": " Sure, I'd be happy to help! Based on the information

provided, the name of the disorder or disease being described is likely COLON POLYPS.\n\nP.S. ···"}

- ift.jsonl: 利用 ChatGPT 对多约束指令产生回答，形成 sft 数据集。 1500 条

  {"prompt": "In this task, you will be given a sentence. ·····..input : In individuals with mutations in either region 2 or region 3 , ·····, "output": "SURE I'D BE HAPPY TO HELP! BASED ON THE INFORMATION PROVIDED THE NAME OF THE DISORDER OR DISEASE BEING DESCRIBED IS LIKELY COLON POLYPS WITH MUTATIONS.\n\nP.S. IT'S IMPORTANT TO NOTE THAT COLON POLYPS ARE GROWTHS ON ···GOOD [RESTAURANT] FOR A CELEBRATORY DINNER AFTER MY SCREENING?"}

- dpo.jsonl: 利用 GPT 对未遵循的约束逐一修改，形成 dpo 数据集。2781 条+10000 条通用数据.

  {"prompt": "In this task, you will be given a sentence. ·····..input : In individuals with mutations in either region 2 or region 3 , ·····, "constraint": "The response must contain at least 1 placeholders (i.e., [restaurant])", **"rejected"**: " Sure, I'd be happy to help! Based on the information provided, the name of the disorder or disease being described is likely COLON POLYPS.\n\nP.S. It's important to note that colon polyps are growths on the lining of the colon that can develop into colon cancer if left untreated. Regular screening for colon polyps and colon cancer is recommended for individuals with a family history of the condition or those with other risk factors. If you have any concerns or symptoms, it's important to speak with a healthcare professional for proper evaluation and treatment.", **"chosen"**: "SURE I'D BE HAPPY TO HELP! BASED ON THE INFORMATION PROVIDED THE NAME OF THE DISORDER OR DISEASE BEING DESCRIBED IS LIKELY COLON POLYPS WITH MUTATIONS.\n\nP.S. ··· FOR PROPER EVALUATION AND TREATMENT. CAN YOU RECOMMEND A GOOD [RESTAURANT] FOR A CELEBRATORY DINNER AFTER MY SCREENING?"}

3) 数据存储

以 jsonl 格式存储，详细信息如下表：

文件名	元素数目
seed_data.jsonl	1500

data.jsonl	1500
checked.jsonl	1500
ift.jsonl	1500
dpo.jsonl	12781

2. 数据挖掘算法

1) 深度学习-监督微调

- 算法思想：直接利用多约束指令和 ChatGPT 产生的输出对模型进行监督微调，使模型学习到在多约束条件下如何生成符合要求的输出，从而提升模型对复杂指令的遵循能力。
- 算法步骤
  a) 从多约束数据集中（data.jsonl）获取指令（prompt+input）和对应的 GPT - 3.5 输出（output）。
  b) 加载预训练模型（LLaMA2 等）。
  c) 将指令输入模型，得到模型的预测输出。
  d) 根据预测输出和 GPT - 3.5 的输出计算损失（例如使用交叉熵损失函数）。
  e) 通过反向传播算法计算损失对模型参数的梯度。
  f) 使用 Adam 优化器根据梯度更新模型参数。
  g) 重复步骤 c - f，直到达到预定的训练轮数。
- 伪代码

```
model = load_pretrained_model('llama2')
# 设置优化器和损失函数
optimizer = set_optimizer()
loss_function = set_loss_function()
# 遍历多约束数据集
for data in multi_constraint_data:
    instruction data['prompt']+ data['input']
    gpt35_output = data['output']
    # 将指令转换为模型可接受的格式
    model_input = convert_to_model_input(instruction)
    # 模型预测
    model_prediction = model.predict(model_input)
    # 计算损失
```

```
            loss = loss_function(model_prediction, gpt35_output)
            # 反向传播
            loss.backward()
            # 更新模型参数
            optimizer.step()
        optimizer.zero_grad()
```

2) 强化学习-直接偏好优化

- 算法思想：通过为每个复杂指令生成正样本集合（教师模型最后一次修改的输出）和负样本集合（教师模型修改的中间输出），利用这些数据构造偏好数据集，从而训练模型使其更符合人类偏好。
- 算法步骤
  a) 对于每个复杂指令，获取教师模型修改过程中的所有输出。
  b) 将教师模型最后一次修改的输出作为正样本，其他中间输出作为负样本，构建正样本集合和负样本集合。这个正样本代表了教师模型经过一系列调整后认为最符合要求的回复，可作为模型学习的目标。这些负样本也提供了有用的监督信号。
  c) 利用正样本集合和负样本集合构造偏好数据集。
  d) 使用偏好数据集进行 DPO 训练，优化模型参数。
- 伪代码

```
    # 加载预训练模型
    model = load_pretrained_model('llama2')
    # 设置优化器（例如 Adam 优化器）和损失函数（例如基于 KL 散度的偏好损失函数）
    optimizer = set_optimizer()
    loss_function = set_loss_function()

    for i, instruction in enumerate(complex_instructions):
        # 构建正样本集合，取教师模型最后一次修改的输出作为正样本
        positive_samples = [teacher_outputs[i][-1]]
        # 构建负样本集合，取教师模型除最后一次修改之外的中间输出作为负样
        negative_samples = teacher_outputs[i][:-1]
        # 构造偏好数据集
```

```
preference_dataset = []
for pos_sample in positive_samples:
    for neg_sample in negative_samples:
        preference_dataset.append((pos_sample, neg_sample))
# 遍历偏好数据集进行 DPO 训练
for pos, neg in preference_dataset:
    pos_input = convert_to_model_input(pos)
    neg_input = convert_to_model_input(neg)
    # 模型对正样本和负样本的预测
    pos_prediction = model.predict(pos_input)
    neg_prediction = model.predict(neg_input)
    # 计算偏好损失
    loss = loss_function(pos_prediction, neg_prediction)
    # 反向传播计算梯度
    loss.backward()
    # 更新模型参数
    optimizer.step()
    optimizer.zero_grad()
```

第四章 实验结果

为了验证我们的方法是否有效，我们进行了一系列的实验，使用我们构造的数据集对大模型进行 DPO 训练，比较训练前后的模型在特定数据集上的性能。

为了便于展示结果，我们采用了在指令遵循能力性能较优的开源模型 LLaMa2 作为基座模型，同时选取指令遵循能力更强的 WizardLM[1]和 OpenChat[2] 作为比较的 Baseline。我们将从以下两个方面探究本次数据挖掘的有效性：

一．使用我们的方法构造出的 DPO 数据是否能够有效提升大模型的复杂指令遵循能力

二．挖掘出的数据是否会导致微调后的大模型在其他通用任务上的能力出现显著下降

我们首先在 IFEVAL 数据集上测试模型的指令遵循能力，实验结果如表 1 所示。结果表明，经过 DPO 训练后的模型在指令遵循能力上有显著的提升，微调后的 LLaMa 在指令级（Instruction level， I-level）准确率和约束级（Constraint level, C-level）准确率上都有显著的提升，且优于原本表现更好的 WizardLM 和 OpenChat

表 1 指令遵循能力的测试结果

model	I-level	C-level
LLaMA2	9.50	42.27
WizardLM	14.00	47.20
OpenChat	16.50	49.07
Our-DPO	**19.00**	**55.73**

接下来，为了测试模型的通用能力，我们使用了以下四个评测大模型知识能力的数据集：MMLU(Hendrycks et al., 2020),、TruthfulQA (Lin et al., 2021)、ARC (Clark et al., 2018)和 HellaSwag (Zellers et al.,2019)

基于这些数据集，我们对微调前后的 LLaMa2 和 OpenChat 进行了性能比较，实验结果如表 2 所示。经过 DPO 训练的模型在这四个数据集上的综合表现与微调前非常接近，这表明，我们构造的数据集在提升模型指令遵循能力的同时，有

效保留了模型原有的通用能力，不会对其泛化性造成严重损害。

表 2 大模型知识能力的测试结果

model	MMLU	TruthfulQA	ARC	HellaSwag	Avg
LLaMA2	54.64	44.12	59.04	81.94	59.94
OpenChat	56.68	44.49	59.64	82.68	60.87
Our-DPO	53.79	48.15	57.76	79.95	59.91

第五章 总结与不足

综上所述，我们的报告面向大型语言模型（LLMs）复杂指令遵循这一方向，探究了如何有效挖掘指令数据，以提升大语言模型在处理复杂指令时的表现能力。我们主要聚焦于有效训练数据的选择，通过实验来验证了获取高质量数据的方法。我们基于互联网上的开放指令数据集构造了种子指令集和多约束指令集，又基于教师模型和学生模型的协作推理，有效挖掘了大模型表现较差的多约束复杂指令数据。在此基础上，我们结合了正样本与负样本，构建了DPO 指令数据集，并通过微调开源大模型，验证了挖掘出的指令数据的有效性，即我们挖掘出的数据集有效提升了大语言模型处理复杂多约束的能力，同时保持了其原有的泛化性，能够适用于不同的任务和场景。

与此同时，我们的实验设计与内容还存在一些不足和可能的改进之处。一方面，在评估大模型遵循复杂指令的能力时，我们的重点在于模型在多重约束条件下的表现。但实际上，由于多约束指令数据构建的随机性，即便模型能够同时满足所有约束条件，这并不代表着它能够完全遵循类型相同或相似的复杂指令。另一方面，在训练数据的构建过程中，我们主要采用了 IFEval 中的“硬性约束”作为数据来源，即可以直接通过 python 脚本来进行判别的约束，如包含某个关键词或符合某种格式等。相比之下，真实世界中的许多任务更倾向于涉及“软约束”，例如语义层面、写作风格等语义上的要求，对于这些约束类型，我们的方法还无法进行有效的判别。因此，未来的研究中，可以进一步扩展至软性约束的处理，如使用大模型来判别其语义内容，以更贴近实际应用场景的需求。

参考文献

[1] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin John-son, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. arXiv preprint arXiv:2305.10403.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 30.

[3] Renze Lou, Kai Zhang, and Wenpeng Yin. 2024. A com-prehensive survey on instruction following. Preprint, arXiv:2303.10475.

[4] Wenpeng Yin, Qinyuan Ye, Pengfei Liu, Xiang Ren, and Hinrich Schütze. 2023. Llm-driven instruction following: Progresses and concerns. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts, pages 19–25.

[5] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244.

[6] Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2023. Follow-bench: A multi-level fine-grained constraints following benchmark for large language models. arXiv preprint arXiv:2310.20410.

[7] Qianyu He, Jie Zeng, Wenhao Huang, Lina Chen, Jin Xiao, Qianxi He, Xunzhe Zhou, Jiaqing Liang, and Yanghua Xiao. 2024. Can large language models understand real-world complex instructions? In Proceedings of the AAAI Conference on Artificial Intelli- gence, volume 38, pages 18188–18196.

[8] Renze Lou, Kai Zhang, and Wenpeng Yin. 2024. A comprehensive survey on instruction following. Preprint, arXiv:2303.10475.

[9] Stiennon, N., Edunov, S., & Raffel, C. (2020). Learning to summarize with human feedback. Advances in Neural Information Processing Systems (Vol. 33, pp. 3005-3015).

[10] Yihan Chen, Benfeng Xu, Quan Wang, Yi Liu, and Zhendong Mao. 2024. Benchmarking large language models on controllable generation under diversified instructions. arXiv preprint arXiv:2401.00690.

[11] Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. Fofo: A benchmark to evaluate llms' format-following capability. arXiv preprint arXiv:2402.18667

[12] Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. 2023. Tell your model where to attend: Post-hoc attention steering for llms. arXiv preprint

arXiv:2311.02262.

[13] Fei Wang, Chao Shang, Sarthak Jain, Shuai Wang, Qiang Ning, Bonan Min, Vittorio Castelli, Yassine Benajiba, and Dan Roth. 2024. From instructions to constraints: Language model alignment with automatic constraint verification. arXiv preprint arXiv:2403.06326.

[14] Haoran Sun, Lixin Liu, Junjie Li, Fengyu Wang, Bao hua Dong, Ran Lin, and Ruohui Huang. 2024. Conifer: Improving complex constrained instruction-following ability of large language models. arXiv preprint arXiv:2404.02823.

[15] Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 505525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[16]Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024a. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In The Twelfth International Conference on Learning Representations.

[17]Qianyu He, Jie Zeng, Wenhao Huang, Lina Chen, Jin Xiao, Qianxi He, Xunzhe Zhou, Jiaqing Liang, and Yanghua Xiao. 2024. Can large language models understand real-world complex instructions? In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 18188–18196.

[18]Pei Wang and Ben Goertzel. 2007. Introduction: Aspects of Artificial General Intelligence. In Proceedings of the 2007 conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006, pages 1–16.

[19] Wen B, Ke P, Gu X, et al. Benchmarking complex instruction-following with multiple constraints composition[J]. arXiv preprint arXiv:2407.03978, 2024.

[20]Xu, Can, et al. Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244 (2023).

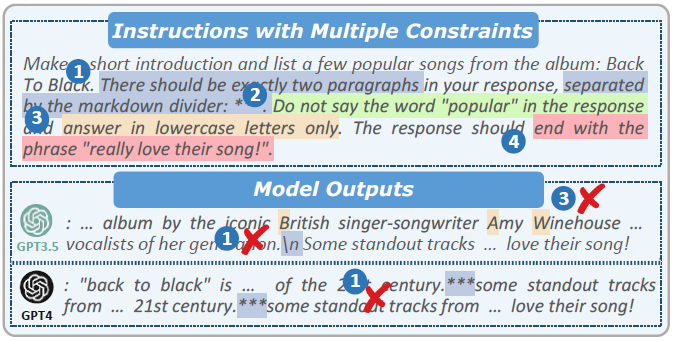
[21]Dong, Guanting, et al. Self-play with Execution Feedback: Improving Instruction-following Capabilities of Large Language Models. arXiv preprint arXiv:2406.13542 (2024).

Finding Data to Enhance Instruction Following Ability of Large Language Models

13组：任清宇 王光帅 吴师师 张景昊

Introduction

大语言模型（LLM)成为现实中很多应用的基础。大语言模型具有准确理解指令并产生期望输出的能力，也就是指令遵循能力。对于大语言模型来说，遵循带有多要求的指令是至关重要的。当前LLM在遵循简单要求指令方面已取得显著进展，但在处理含有多个约束的复杂指令时仍面临挑战，而复杂指令在现实世界的应用中非常普遍，因此提升LLM遵循复杂多约束指令的能力至关重要。



在本次研究中，我们的主要贡献如下：

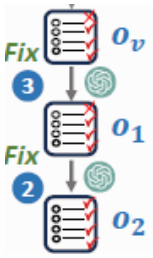
- 探索**什么样的数据**对于增加大预言模型的多约束遵循能力是有效的
- 如何**获得**这样的数据
- 如何**利用**这样的数据

Method

1. 确定有效训练数据的类型

- 将数据分为单一约束（原子数据）和复合约束（复合数据）。
- 分别使用这两类数据训练LLM。
- 评估并比较两种训练数据下的模型性能。
- 结论：复合约束指令更有效。

2. 生成高质量多约束指令数据



- 从多个来源收集初始指令。
- 使用学生模型（LLaMA2）生成初始输出。
- 使用测试程序来识别模型未能遵循的约束。
- 使用教师模型（ChatGPT）逐一纠正这些错误。
- 修改过程中的数据用于后续训练。

3. 利用构造的数据进行DPO训练

- 对于每个复杂指令，生成正样本集合（教师模型最后一次修改的输出）和负样本集合（教师模型修改的中间输出）。
- 利用以上数据构造偏好数据集，进行DPO训练。

Result

1. 指令遵循能力测试

在IFEval上测试了模型指令遵循能力。结果表明，我们的方法在有效提升模型指令遵循能力。

Model	I-level	C-level
LLaMA2	9.50	42.27
WizardLM	14.00	47.20
OpenChat	16.50	49.07
Ours	**19.00**	**55.73**

2. 通用能力测试

在MMLU、TruthfulQA、ARC、HellaSwag四个数据集上测试了模型的通用能力。结果表明，我们的方法在提升模型指令遵循能力的同时，能够保持模型的通用能力。

Model	MMLU	TruthfulQA	ARC	HellaSwag	Avg
LLaMA2	54.64	44.12	59.04	81.94	59.94
OpenChat	56.68	44.49	59.64	82.68	60.87
Ours	53.79	48.15	57.76	79.95	**59.91**