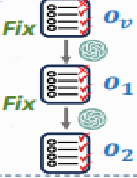
1. 数据集
2. **数据获取**

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。

1. **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?"}

1. 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 |

1. 数据挖掘算法
2. 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

1. 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）。
2. 加载预训练模型（LLaMA2 等）。
3. 将指令输入模型，得到模型的预测输出。
4. 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）。
5. 通过反向传播算法计算损失对模型参数的梯度。
6. 使用Adam优化器根据梯度更新模型参数。
7. 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()

1. 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

1. 对于每个复杂指令，获取教师模型修改过程中的所有输出。
2. 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。
3. 利用正样本集合和负样本集合构造偏好数据集。
4. 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()

# 遍历复杂指令及其对应的教师模型输出

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:

# 每个正样本与每个负样本组成一对偏好数据，(正样本, 负样本)

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()