第六课

基础作业

使用 OpenCompass 评测 internlm2-chat-1_8b 模型在 C-Eval 数据集上的性能

基础概念

评测方法

OpenCompass 采取客观评测与主观评测相结合的方法。针对具有确定性答案的能力维度和场景，通过构造丰富完善的评测集，对模型能力进行综合评价。针对体现模型能力的开放式或半开放式的问题、模型安全问题等，采用主客观相结合的评测方式。

客观评测

针对具有标准答案的客观问题，我们可以我们可以通过使用定量指标比较模型的输出与标准答案的差异，并根据结果衡量模型的性能。同时，由于大语言模型输出自由度较高，在评测阶段，我们需要对其输入和输出作一定的规范和设计，尽可能减少噪声输出在评测阶段的影响，才能对模型的能力有更加完整和客观的评价。 为了更好地激发出模型在题目测试领域的能力，并引导模型按照一定的模板输出答案，OpenCompass 采用提示词工程 （prompt engineering）和语境学习（in-context learning）进行客观评测。 在客观评测的具体实践中，我们通常采用下列两种方式进行模型输出结果的评测：

- 判别式评测：该评测方式基于将问题与候选答案组合在一起，计算模型在所有组合上的困惑度（perplexity），并选择困惑度最小的答案作为模型的最终输出。例如，若模型在 问题? 答案1 上的困惑度为 0.1，在 问题? 答案2 上的困惑度为 0.2，最终我们会选择 答案1 作为模型的输出。
- 生成式评测：该评测方式主要用于生成类任务，如语言翻译、程序生成、逻辑分析题等。具体实践时，使用问题作为模型的原始输入，并留白答案区域待模型进行后续补全。我们通常还需要对其输出进行后处理，以保证输出满足数据集的要求。

主观评测

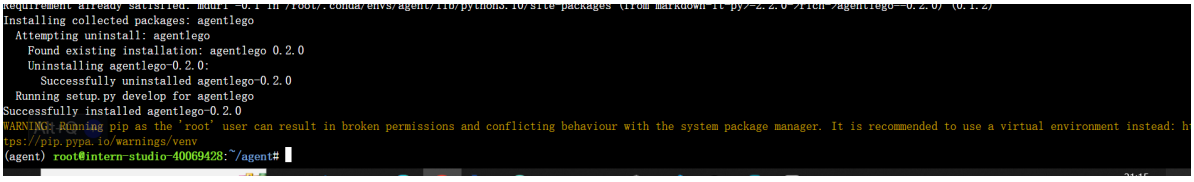
语言表达生动精彩，变化丰富，大量的场景和能力无法凭借客观指标进行评测。针对如模型安全和模型语言能力的评测，以人的主观感受为主的评测更能体现模型的真实能力，并更符合大模型的实际使用场景。 OpenCompass 采取的主观评测方案是指借助受试者的主观判断对具有对话能力的大语言模型进行能力评测。在具体实践中，我们提前基于模型的能力维度构建主观测试问题集合，并将不同模型对于同一问题的不同回复展现给受试者，收集受试者基于主观感受的评分。由于主观测试成本高昂，本方案同时也采用使用性能优异的大语言模拟人类进行主观打分。在实际评测中，本文将采用真实人类专家的主观评测与基于模型打分的主观评测相结合的方式开展模型能力评估。 在具体开展主观评测时，OpenComapss 采用单模型回复满意度统计和多模型满意度比较两种方式开展具体的评测工作。

Lagent 和 AgentLego 都提供了两种安装方法，一种是通过 pip 直接进行安装，另一种则是从源码进行安装。为了方便使用 Lagent 的 Web Demo 以及 AgentLego 的 WebUI，我们选择直接从源码进行安装。 此处附上源码安装的相关帮助文档：

- Lagent：https://lagent.readthedocs.io/zh-cn/latest/get_started/install.html
- AgentLego：https://agentlego.readthedocs.io/zh-cn/latest/get_started.html

可以执行如下命令进行安装：

```
cd /root/agent
conda activate agent
git clone https://gitee.com/internlm/lagent.git
cd lagent && git checkout 581d9fb && pip install -e . && cd ..
git clone https://gitee.com/internlm/agentlego.git
cd agentlego && git checkout 7769e0d && pip install -e . && cd ..
```



OpenCompass 评估过程

在 OpenCompass 中评估一个模型通常包括以下几个阶段：配置 -> 推理 -> 评估 -> 可视化。

- 配置：这是整个工作流的起点。您需要配置整个评估过程，选择要评估的模型和数据集。此外，还可以选择评估策略、计算后端等，并定义显示结果的方式。
- 推理与评估：在这个阶段，OpenCompass 将会开始对模型和数据集进行并行推理和评估。推理阶段主要是让模型从数据集产生输出，而评估阶段则是衡量这些输出与标准答案的匹配程度。这两个过程会被拆分为多个同时运行的“任务”以提高效率，但请注意，如果计算资源有限，这种策略可能会使评测变得更慢。如果需要了解该问题及解决方案，可以参考 FAQ: 效率。
- 可视化：评估完成后，OpenCompass 将结果整理成易读的表格，并将其保存为 CSV 和 TXT 文件。你也可以激活飞书状态上报功能，此后可以在飞书客户端中及时获得评测状态报告。 接下来，我们将展示 OpenCompass 的基础用法，展示书生浦语在 `C-Eval` 基准任务上的评估。它们的配置文件可以在 `configs/eval_demo.py` 中找到。

具体流程

环境配置

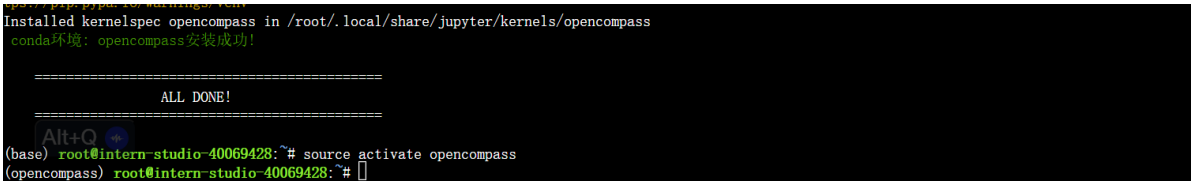
创建开发机和 conda 环境

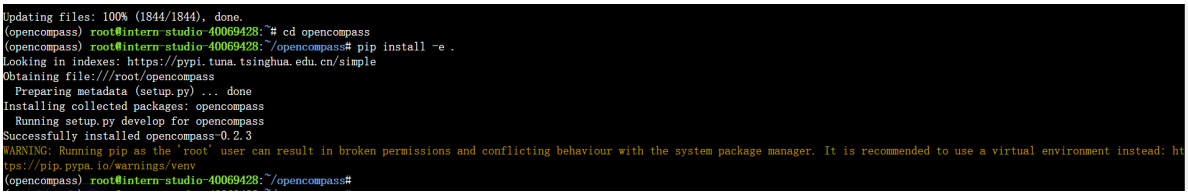
在创建开发机界面选择镜像为 Cuda11.7-conda，并选择 GPU 为10% A100。

面向GPU的环境安装

```
studio-conda -o internlm-base -t opencompass
source activate opencompass
git clone -b 0.2.4 https://github.com/open-compass/opencompass
cd opencompass
pip install -e .
```

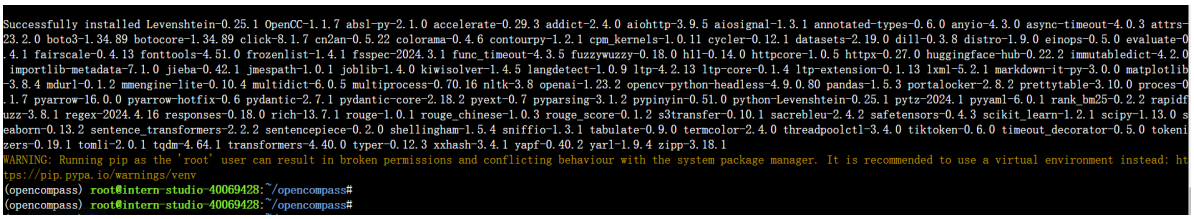
```
Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
|Executing transaction: done
\#
# To activate this environment, use
#
#     $ conda activate opencompass
#
# To deactivate an active environment, use
#
#     $ conda deactivate

[2/2] 同步当前conda环境至jupyterlab kernel
```





安装依赖（注意观察是否虚拟环境下安装依赖，否则影响后续module调用）

```
pip install -r requirements.txt
```

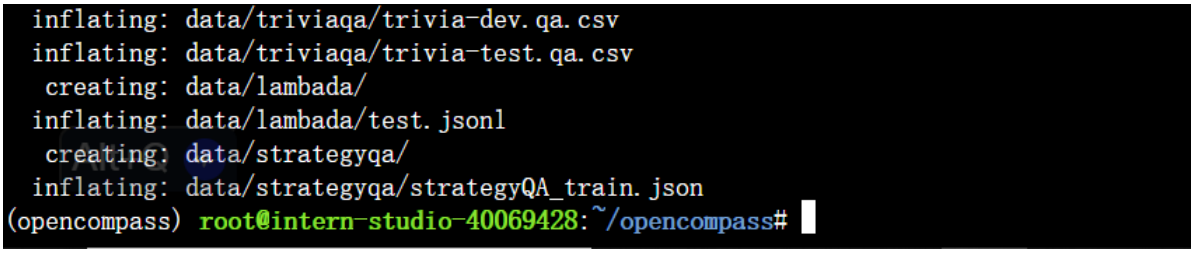


有部分第三方功能,如代码能力基准测试 HumanEval 以及 Llama 格式的模型评测,可能需要额外步骤才能正常运行,

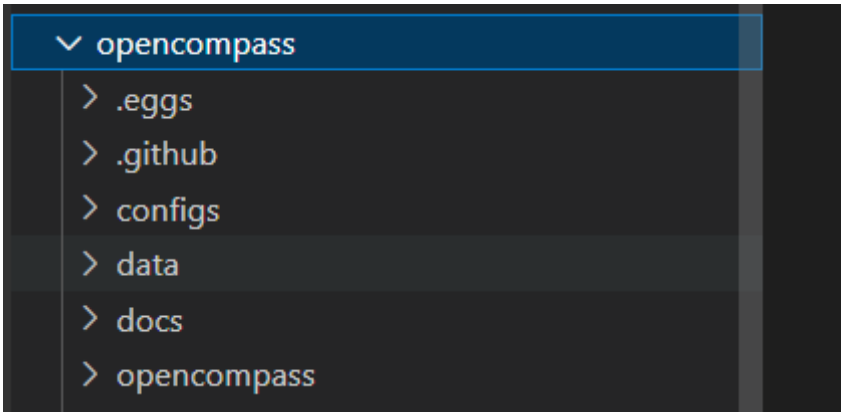
数据准备

解压评测数据集到 data/ 处

```
cp /share/temp/datasets/OpenCompassData-core-20231110.zip /root/opencompass/
unzip OpenCompassData-core-20231110.zip
```



将会在 OpenCompass 下看到data文件夹



查看支持的数据集和模型

列出所有跟 InternLM 及 C-Eval 相关的配置

```
source activate opencompass
cd opencompass
python tools/list_configs.py internlm ceval
```

可能出现的报错

```
(opencompass) root@intern-studio-40069428:~/opencompass# python tools/list_configs.py internlm ceval
Traceback (most recent call last):
  File "/root/opencompass/tools/list_configs.py", line 3, in <module>
    import tabulate
ModuleNotFoundError: No module named 'tabulate'
(opencompass) root@intern-studio-40069428:~/opencompass#
```

解决方案：

```
pip install tabulate
#或执行一遍
pip install -r requirements.txt
```

列出相关配置将会看到

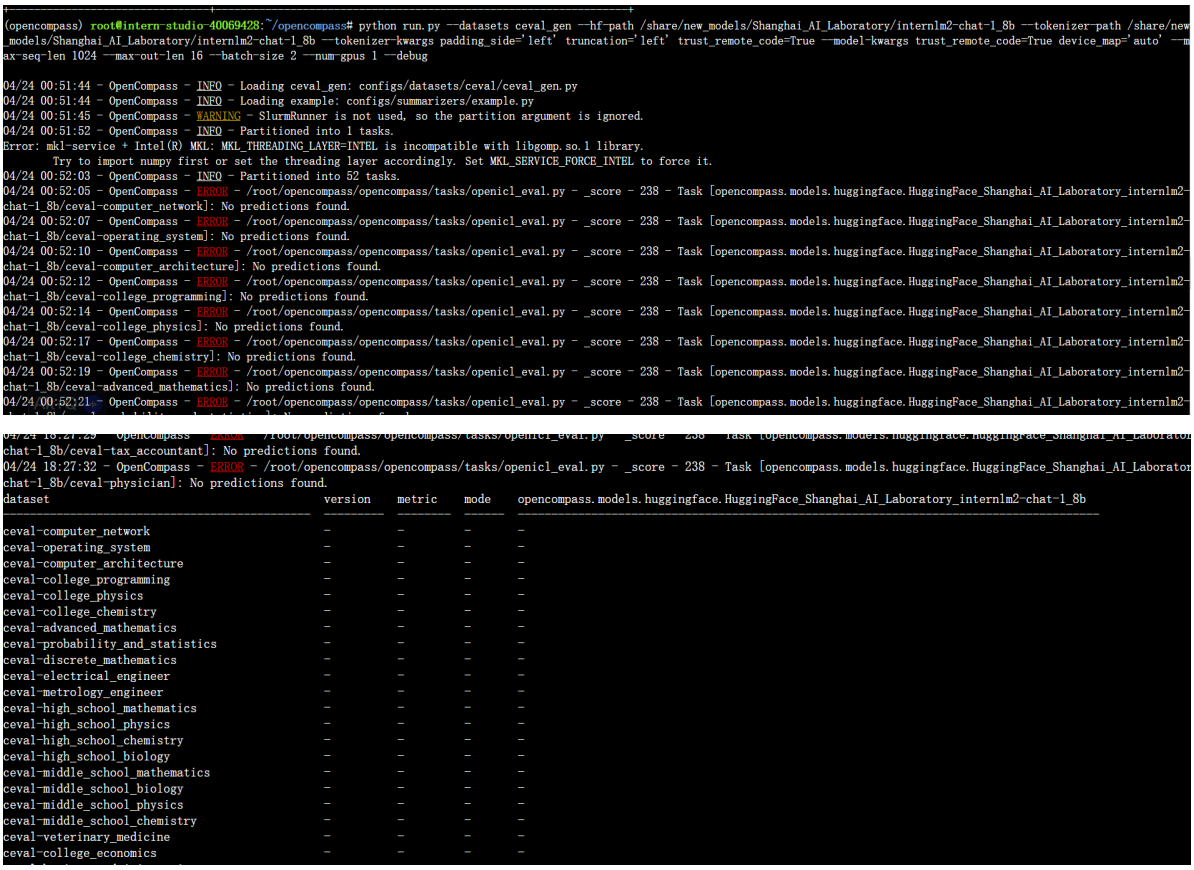
```
(opencompass) root@intern-studio-40069428:~/opencompass# python tools/list_configs.py internlm ceval
+-----------------------------------------+-----------------------------------------------------------------------------------+
| Model                                   | Config Path                                                                       |
|-----------------------------------------+-----------------------------------------------------------------------------------|
| hf_internlm2_1_8b                       | configs/models/hf_internlm/hf_internlm2_1_8b.py                                   |
| hf_internlm2_20b                        | configs/models/hf_internlm/hf_internlm2_20b.py                                    |
| hf_internlm2_7b                         | configs/models/hf_internlm/hf_internlm2_7b.py                                     |
| hf_internlm2_base_20b                   | configs/models/hf_internlm/hf_internlm2_base_20b.py                               |
| hf_internlm2_base_7b                    | configs/models/hf_internlm/hf_internlm2_base_7b.py                                |
| hf_internlm2_chat_1_8b                  | configs/models/hf_internlm/hf_internlm2_chat_1_8b.py                              |
| hf_internlm2_chat_1_8b_sft              | configs/models/hf_internlm/hf_internlm2_chat_1_8b_sft.py                          |
| hf_internlm2_chat_20b                   | configs/models/hf_internlm/hf_internlm2_chat_20b.py                               |
| hf_internlm2_chat_20b_sft               | configs/models/hf_internlm/hf_internlm2_chat_20b_sft.py                           |
| hf_internlm2_chat_20b_with_system       | configs/models/hf_internlm/hf_internlm2_chat_20b_with_system.py                   |
| hf_internlm2_chat_7b                    | configs/models/hf_internlm/hf_internlm2_chat_7b.py                                |
| hf_internlm2_chat_7b_sft                | configs/models/hf_internlm/hf_internlm2_chat_7b_sft.py                            |
| hf_internlm2_chat_7b_with_system        | configs/models/hf_internlm/hf_internlm2_chat_7b_with_system.py                    |
| hf_internlm2_chat_math_20b              | configs/models/hf_internlm/hf_internlm2_chat_math_20b.py                          |
| hf_internlm2_chat_math_20b_with_system  | configs/models/hf_internlm/hf_internlm2_chat_math_20b_with_system.py              |
| hf_internlm2_chat_math_7b               | configs/models/hf_internlm/hf_internlm2_chat_math_7b.py                           |
| hf_internlm2_chat_math_7b_with_system   | configs/models/hf_internlm/hf_internlm2_chat_math_7b_with_system.py               |
| hf_internlm_20b                         | configs/models/hf_internlm/hf_internlm_20b.py                                     |
| hf_internlm_7b                          | configs/models/hf_internlm/hf_internlm_7b.py                                      |
| hf_internlm_chat_20b                    | configs/models/hf_internlm/hf_internlm_chat_20b.py                                |
| hf_internlm_chat_7b                     | configs/models/hf_internlm/hf_internlm_chat_7b.py                                 |
| hf_internlm_chat_7b_8k                  | configs/models/hf_internlm/hf_internlm_chat_7b_8k.py                              |
| hf_internlm_chat_7b_v1_1                | configs/models/hf_internlm/hf_internlm_chat_7b_v1_1.py                            |
| internlm_7b                             | configs/models/internlm/internlm_7b.py                                            |
| lmdeploy_internlm2_chat_20b             | configs/models/hf_internlm/lmdeploy_internlm2_chat_20b.py                         |
| lmdeploy_internlm2_chat_7b              | configs/models/hf_internlm/lmdeploy_internlm2_chat_7b.py                          |
| ms_internlm_chat_7b_8k                  | configs/models/ms_internlm/ms_internlm_chat_7b_8k.py                              |
+-----------------------------------------+-----------------------------------------------------------------------------------+
```

```
+-------------------------------+-----------------------------------------------------------------+
| Dataset                       | Config Path                                                     |
|-------------------------------+-----------------------------------------------------------------|
| ceval_clean_ppl               | configs/datasets/ceval/ceval_clean_ppl.py                       |
| ceval_contamination_ppl_810ec6 | configs/datasets/contamination/ceval_contamination_ppl_810ec6.py |
| ceval_gen                     | configs/datasets/ceval/ceval_gen.py                             |
| ceval_gen_2daf24              | configs/datasets/ceval/ceval_gen_2daf24.py                      |
| ceval_gen_5f30c7              | configs/datasets/ceval/ceval_gen_5f30c7.py                      |
| ceval_internal_ppl_1cd8bf     | configs/datasets/ceval/ceval_internal_ppl_1cd8bf.py             |
| ceval_ppl                     | configs/datasets/ceval/ceval_ppl.py                             |
| ceval_ppl_1cd8bf              | configs/datasets/ceval/ceval_ppl_1cd8bf.py                      |
| ceval_ppl_578f8d              | configs/datasets/ceval/ceval_ppl_578f8d.py                      |
| ceval_ppl_93e5ce              | configs/datasets/ceval/ceval_ppl_93e5ce.py                      |
| ceval_zero_shot_gen_bd40ef    | configs/datasets/ceval/ceval_zero_shot_gen_bd40ef.py            |
+-------------------------------+-----------------------------------------------------------------+
(opencompass) root@intern-studio-40069428:~/opencompass#
```

启动评测 (10% A100 8GB 资源)

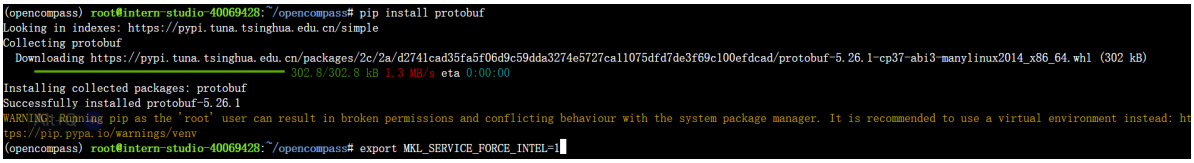
通过以下命令评测 InternLM2-Chat-1.8B 模型在 C-Eval 数据集上的性能。由于 OpenCompass 默认并行启动评估过程，我们可以在第一次运行时以 --debug 模式启动评估，并检查是否存在问题。在 --debug 模式下，任务将按顺序执行，并实时打印输出。

```
python run.py --datasets ceval_gen --hf-path
/share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b --tokenizer-path
/share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b --tokenizer-kwargs
padding_side='left' truncation='left' trust_remote_code=True --model-kwargs
trust_remote_code=True device_map='auto' --max-seq-len 1024 --max-out-len 16 --
batch-size 2 --num-gpus 1 --debug
```



遇到错误：安装缺少的包

```
pip install protobuf
```



命令解析

```
python run.py \python run.py
--datasets ceval_gen \
--hf-path /share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b \  #
HuggingFace 模型路径
--tokenizer-path /share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b \
# HuggingFace tokenizer 路径（如果与模型路径相同，可以省略）
--tokenizer-kwargs padding_side='left' truncation='left' trust_remote_code=True \
# 构建 tokenizer 的参数
--model-kwargs device_map='auto' trust_remote_code=True \  # 构建模型的参数
--max-seq-len 1024 \  # 模型可以接受的最大序列长度
--max-out-len 16 \  # 生成的最大 token 数
--batch-size 2  \  # 批量大小
--num-gpus 1  # 运行模型所需的 GPU 数量
--debug
```

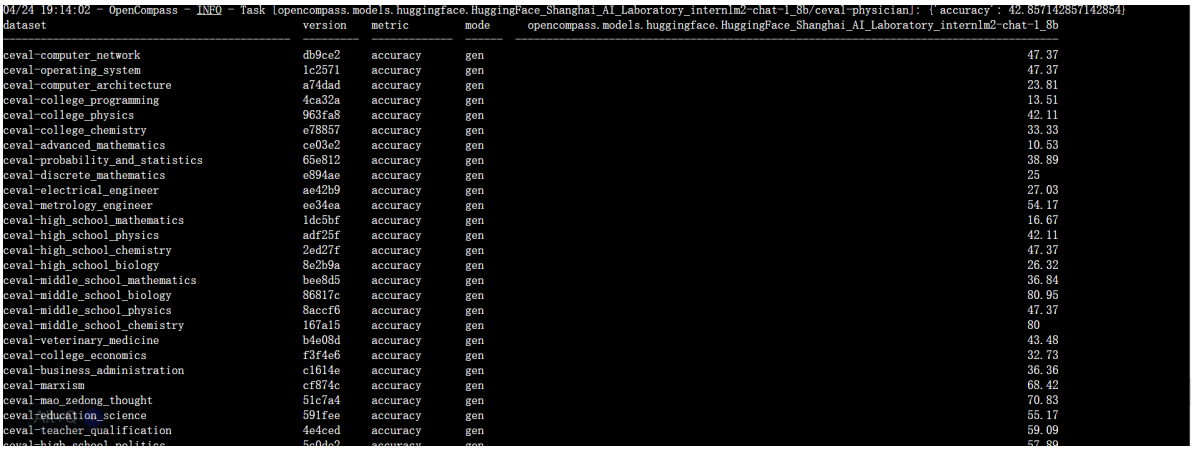
遇到报错

解决方案

```
export MKL_SERVICE_FORCE_INTEL=1
#或
export MKL_THREADING_LAYER=GNU
```

正常情况如下显示：

评测完成结果



评测结果（基础作业完成）

```
dataset                                          version    metric         mode
   opencompass.models.huggingface.HuggingFace_Shanghai_AI_Laboratory_internlm2-
chat-1_8b
-----------------------------------------------  ---------  -------------  ------
-------------------------------------------------------------------------------
------
```

```
ceval-computer_network                  db9ce2    accuracy    gen

47.37
ceval-operating_system                  1c2571    accuracy    gen

47.37
ceval-computer_architecture             a74dad    accuracy    gen

23.81
ceval-college_programming               4ca32a    accuracy    gen

13.51
ceval-college_physics                   963fa8    accuracy    gen

42.11
ceval-college_chemistry                 e78857    accuracy    gen

33.33
ceval-advanced_mathematics              ce03e2    accuracy    gen

10.53
ceval-probability_and_statistics        65e812    accuracy    gen

38.89
ceval-discrete_mathematics              e894ae    accuracy    gen

25
ceval-electrical_engineer               ae42b9    accuracy    gen

27.03
ceval-metrology_engineer                ee34ea    accuracy    gen

54.17
ceval-high_school_mathematics           1dc5bf    accuracy    gen

16.67
ceval-high_school_physics               adf25f    accuracy    gen

42.11
ceval-high_school_chemistry             2ed27f    accuracy    gen

47.37
ceval-high_school_biology               8e2b9a    accuracy    gen

26.32
ceval-middle_school_mathematics         bee8d5    accuracy    gen

36.84
ceval-middle_school_biology             86817c    accuracy    gen

80.95
ceval-middle_school_physics             8accf6    accuracy    gen

47.37
ceval-middle_school_chemistry           167a15    accuracy    gen

80
```

```
ceval-veterinary_medicine                       b4e08d     accuracy       gen

43.48
ceval-college_economics                         f3f4e6     accuracy       gen

32.73
ceval-business_administration                   c1614e     accuracy       gen

36.36
ceval-marxism                                   cf874c     accuracy       gen

68.42
ceval-mao_zedong_thought                        51c7a4     accuracy       gen

70.83
ceval-education_science                         591fee     accuracy       gen

55.17
ceval-teacher_qualification                     4e4ced     accuracy       gen

59.09
ceval-high_school_politics                      5c0de2     accuracy       gen

57.89
ceval-high_school_geography                     865461     accuracy       gen

47.37
ceval-middle_school_politics                    5be3e7     accuracy       gen

71.43
ceval-middle_school_geography                   8a63be     accuracy       gen

75
ceval-modern_chinese_history                    fc01af     accuracy       gen

52.17
ceval-ideological_and_moral_cultivation         a2aa4a     accuracy       gen

73.68
ceval-logic                                     f5b022     accuracy       gen

27.27
ceval-law                                       a110a1     accuracy       gen

29.17
ceval-chinese_language_and_literature           0f8b68     accuracy       gen

47.83
ceval-art_studies                               2a1300     accuracy       gen

42.42
ceval-professional_tour_guide                   4e673e     accuracy       gen

51.72
ceval-legal_professional                        ce8787     accuracy       gen

34.78
```

```
ceval-high_school_chinese                          315705     accuracy       gen

42.11
ceval-high_school_history                          7eb30a     accuracy       gen

65
ceval-middle_school_history                        48ab4a     accuracy       gen

86.36
ceval-civil_servant                                87d061     accuracy       gen

42.55
ceval-sports_science                               70f27b     accuracy       gen

52.63
ceval-plant_protection                             8941f9     accuracy       gen

40.91
ceval-basic_medicine                               c409d6     accuracy       gen

68.42
ceval-clinical_medicine                            49e82d     accuracy       gen

31.82
ceval-urban_and_rural_planner                      95b885     accuracy       gen

47.83
ceval-accountant                                   002837     accuracy       gen

36.73
ceval-fire_engineer                                bc23f5     accuracy       gen

38.71
ceval-environmental_impact_assessment_engineer     c64e2d     accuracy       gen

51.61
ceval-tax_accountant                               3a5e3c     accuracy       gen

36.73
ceval-physician                                    6e277d     accuracy       gen

42.86
ceval-stem                                         -          naive_average  gen

39.21
ceval-social-science                               -          naive_average  gen

57.43
ceval-humanities                                   -          naive_average  gen

50.23
ceval-other                                        -          naive_average  gen

44.62
ceval-hard                                         -          naive_average  gen

32
```

```
ceval                                           -          naive_average  gen

46.19
04/24 19:14:03 - OpenCompass - INFO - write summary to
/root/opencompass/outputs/default/20240424_183626/summary/summary_20240424_18362
6.txt
04/24 19:14:03 - OpenCompass - INFO - write csv to
/root/opencompass/outputs/default/20240424_183626/summary/summary_20240424_18362
6.csv
```

在上面的结果中发现，8b模型评测表现最好的前三个数据集，和表现最差的数据集

performance	dataset	version	opencompass……internlm2-chat-1_8b
high	ceval-middle_school_history	48ab4a	86.36
high	ceval-middle_school_biology	86817c	80.95
hight	ceval-mao_zedong_thought	51c7a4	70.83
low	ceval-advanced_mathematics	ce03e2	10.53

自定义数据集客主观评测

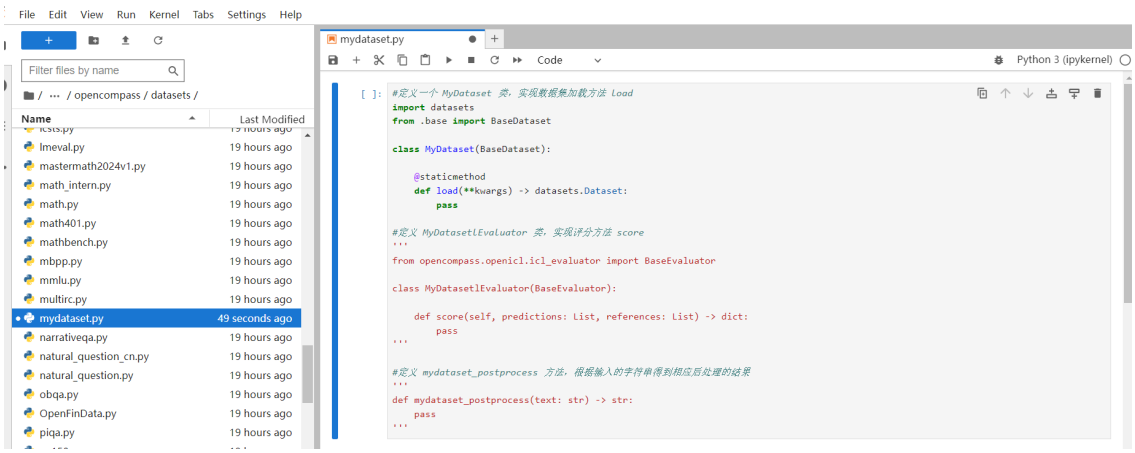
自定义数据集客观评测

自建客观数据集步骤参考 https://opencompass.readthedocs.io/zh-cn/latest/advanced_guides/new_dataset.html

1.在 `opencompass/datasets` 文件夹新增数据集脚本 `mydataset.py`

该脚本包含

- 数据集及其加载方式，需要定义一个 `MyDataset` 类，实现数据集加载方法 `load`，该方法为静态方法，需要返回 `datasets.Dataset` 类型的数据。这里我们使用 huggingface dataset 作为数据集的统一接口，避免引入额外的逻辑。
- （可选）如果 OpenCompass 已有的评测器不能满足需要，需要用户定义 `MyDatasetlEvaluator` 类，实现评分方法 `score`，需要根据输入的 `predictions` 和 `references` 列表，得到需要的字典。由于一个数据集可能存在多种 metric，需要返回一个 metrics 以及对应 scores 的相关字典。
- （可选）如果 OpenCompass 已有的后处理方法不能满足需要，需要用户定义 `mydataset_postprocess` 方法，根据输入的字符串得到相应后处理的结果。
- 脚本代码



2.在配置文件中新增以下配置

```
#新增配置
from opencompass.datasets import MyDataset, MyDatasetlEvaluator, mydataset_postprocess

mydataset_eval_cfg = dict(
    evaluator=dict(type=MyDatasetlEvaluator),
    pred_postprocessor=dict(type=mydataset_postprocess))

mydataset_datasets = [
    dict(
        type=MyDataset,
        ...,
        reader_cfg=...,
        infer_cfg=...,
        eval_cfg=mydataset_eval_cfg)
]
```

自定义数据集主观评测（待更新）

由于客观评测只能反映模型在一些性能数据上的指标，没法完全真实地反映模型在与人类对话时的表现，因此需要在真实的对话场景下通过主观评测的方式翻译模型的真实性能。 而由于完全靠人力来进行主观评测是费时费力的，因此有很多利用模型来进行主观评测的方式。 这些方式主要可以分为以下几类：打分，对战，多模型评测等。

自建主观数据集步骤参考链接 https://opencompass.readthedocs.io/zh-cn/latest/advanced_guides/subjective_evaluation.html

为了探究模型的主观能力采用JudgeLLM作为人类评估者的替代品（LLM-as-a-Judge）。

流行的评估方法

- Compare模式：将模型的回答进行两两比较，以计算对战其胜率。
- Score模式：针对单模型的回答进行打分（例如：Chatbot Arena）。

目前已支持的主观评测数据集

1. AlginBench（https://github.com/THUDM/AlignBench）
2. MTBench （https://github.com/lm-sys/FastChat）
3. AlpacaEvalv2 （https://github.com/tatsu-lab/alpaca_eval）
4. CompassArena（内部数据集）

主观评测的具体流程

1. 评测数据集准备
2. 使用API模型或者开源模型进行问题答案的推理
3. 使用选定的评价模型(JudgeLLM)对模型输出进行评估
4. 对评价模型返回的预测结果进行解析并计算数值指标

第一步：数据准备

对于对战模式和打分模式，我们各提供了一个demo测试集如下：

```
### 对战模式示例
[
    {
        "question": "如果我在空中垂直抛球，球最初向哪个方向行进？",
        "capability": "知识-社会常识",
        "others": {
            "question": "如果我在空中垂直抛球，球最初向哪个方向行进？",
            "evaluating_guidance": "",
            "reference_answer": "上"
        }
    },...]

### 打分模式数据集示例
[
    {
        "question": "请你扮演一个邮件管家，我让你给谁发送什么主题的邮件，你就帮我扩充好邮件正文，并打印在聊天框里。你需要根据我提供的邮件收件人以及邮件主题，来斟酌用词，并使用合适的敬语。现在请给导师发送邮件，询问他是否可以下周三下午15:00进行科研同步会，大约200字。",
        "capability": "邮件通知",
        "others": ""
    },
```

如果要准备自己的数据集，请按照以下字段进行提供，并整理为一个json文件：

- ‘question’：问题描述
- ‘capability’：题目所属的能力维度
- ‘others’：其他可能需要对题目进行特殊处理的项目

以上三个字段是必要的，用户也可以添加其他字段，如果需要对每个问题的prompt进行单独处理，可以在’others’字段中进行一些额外设置，并在Dataset类中添加相应的字段。

第二步：构建评测配置（对战模式）

对于两回答比较，更详细的config setting请参考 `config/eval_subjective_compare.py`，下面我们提供了部分简略版的注释，方便用户理解配置文件的含义。

```
from mmengine.config import read_base
from opencompass.models import HuggingFaceCausalLM, HuggingFace, OpenAI

from opencompass.partitioners import NaivePartitioner
from opencompass.partitioners.sub_naive import SubjectiveNaivePartitioner
from opencompass.runners import LocalRunner
from opencompass.runners import SlurmSequentialRunner
from opencompass.tasks import OpenICLInferTask
from opencompass.tasks.subjective_eval import SubjectiveEvalTask
```

```
from opencompass.summarizers import Corev2Summarizer

with read_base():
    # 导入预设模型
    from .models.qwen.hf_qwen_7b_chat import models as hf_qwen_7b_chat
    from .models.chatglm.hf_chatglm3_6b import models as hf_chatglm3_6b
    from .models.qwen.hf_qwen_14b_chat import models as hf_qwen_14b_chat
    from .models.openai.gpt_4 import models as gpt4_model
    from .datasets.subjective_cmp.subjective_corev2 import subjective_datasets

# 评测数据集
datasets = [*subjective_datasets]

# 待测模型列表
models = [*hf_qwen_7b_chat, *hf_chatglm3_6b]

# 推理配置
infer = dict(
    partitioner=dict(type=NaivePartitioner),
    runner=dict(
        type=SlurmSequentialRunner,
        partition='llmeval',
        quotatype='auto',
        max_num_workers=256,
        task=dict(type=OpenICLInferTask)),
)
# 评测配置
eval = dict(
    partitioner=dict(
        type=SubjectiveNaivePartitioner,
        mode='m2n', # m个模型 与 n个模型进行对战
        #  在m2n模式下，需要指定base_models和compare_models，将会对base_models和
compare_models生成对应的两两pair（去重且不会与自身进行比较）
        base_models = [*hf_qwen_14b_chat], # 用于对比的基线模型
        compare_models = [*hf_baichuan2_7b, *hf_chatglm3_6b] # 待评测模型
    ),
    runner=dict(
        type=SlurmSequentialRunner,
        partition='llmeval',
        quotatype='auto',
        max_num_workers=256,
        task=dict(
            type=SubjectiveEvalTask,
        judge_cfg=gpt4_model # 评价模型
        )),
)
work_dir = './outputs/subjective/' #指定工作目录，在此工作目录下，若使用--reuse参数启动
评测，将自动复用该目录下已有的所有结果

summarizer = dict(
    type=Corev2Summarizer,  #自定义数据集Summarizer
    match_method='smart', #自定义答案提取方式
)
```

此外，在数据集的配置config中，还可以选择两回答比较时的回答顺序，请参考 `config/eval_subjective_compare.py`，当 `infer_order` 设置为 `random` 时，将对两模型的回复顺序进行随机打乱，当 `infer_order` 设置为 `double` 时，将把两模型的回复按两种先后顺序进行判断。

第二步：构建评测配置（打分模式）

对于单回答打分，更详细的config setting请参考 `config/eval_subjective_score.py`，该config的大部分都与两回答比较的config相同，只需要修改评测模式即可，将评测模式设置为 `singlescore`。

第三步 启动评测并输出评测结果

```
python run.py configs/eval_subjective_score.py -r
```

- `-r` 参数支持复用模型推理和评估结果。

JudgeLLM的评测回复会保存在 `output/.../results/timestamp/xxmodel/xxdataset/.json` 评测报告则会输出到 `output/.../summary/timestamp/report.csv`。

Opencompass 已经支持了很多的JudgeLLM，实际上，你可以将Opencompass中所支持的所有模型都当作JudgeLLM使用。 我们列出目前比较流行的开源JudgeLLM：

1. Auto-J，请参考 `configs/models/judge_llm/auto_j`

如果使用了该方法，请添加引用:

```
@article{li2023generative,
  title={Generative judge for evaluating alignment},
  author={Li, Junlong and Sun, Shichao and Yuan, Weizhe and Fan, Run-Ze and Zhao,
Hai and Liu, Pengfei},
  journal={arXiv preprint arXiv:2310.05470},
  year={2023}
}
@misc{2023opencompass,
    title={OpenCompass: A Universal Evaluation Platform for Foundation Models},
    author={OpenCompass Contributors},
    howpublished = {\url{https://github.com/open-compass/opencompass}},
    year={2023}
}
```

1. JudgeLM，请参考 `configs/models/judge_llm/judgelm`

如果使用了该方法，请添加引用:

```
@article{zhu2023judgelm,
  title={JudgeLM: Fine-tuned Large Language Models are Scalable Judges},
  author={Zhu, Lianghui and Wang, Xinggang and Wang, Xinlong},
  journal={arXiv preprint arXiv:2310.17631},
  year={2023}
}
@misc{2023opencompass,
    title={OpenCompass: A Universal Evaluation Platform for Foundation Models},
    author={OpenCompass Contributors},
    howpublished = {\url{https://github.com/open-compass/opencompass}},
    year={2023}
}
```

1. PandaLM，请参考 `configs/models/judge_llm/pandalm`

如果使用了该方法，请添加引用:

```
@article{wang2023pandalm,
  title={PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization},
  author={Wang, Yidong and Yu, Zhuohao and Zeng, Zhengran and Yang, Linyi and Wang, Cunxiang and Chen, Hao and Jiang, Chaoya and Xie, Rui and Wang, Jindong and Xie, Xing and others},
  journal={arXiv preprint arXiv:2306.05087},
  year={2023}
}
@misc{2023opencompass,
    title={OpenCompass: A Universal Evaluation Platform for Foundation Models},
    author={OpenCompass Contributors},
    howpublished = {\url{https://github.com/open-compass/opencompass}},
    year={2023}
}
```

主观多轮对话评测

在OpenCompass中我们同样支持了主观的多轮对话评测，以MT-Bench为例，对MTBench的评测可以参见`configs/eval_subjective_mtbench.py`

在多轮对话评测中，你需要将数据格式整理为如下的dialogue格式

```
"dialogue": [
            {
                "role": "user",
                "content": "Imagine you are participating in a race with a group of people. If you have just overtaken the second person, what's your current position? Where is the person you just overtook?"
            },
            {
                "role": "assistant",
                "content": ""
            },
            {
                "role": "user",
```

```
                "content": "If the \"second person\" is changed to \"last
person\" in the above question, what would the answer be?"
            },
            {
                "role": "assistant",
                "content": ""
            }
        ],
```

值得注意的是，由于MTBench各不同的题目类型设置了不同的温度，因此我们需要将原始数据文件按照温度分成三个不同的子集以分别推理，针对不同的子集我们可以设置不同的温度，具体设置参加 `configs\datasets\subjective\multiround\mtbench_single_judge_diff_temp.py`

如果使用了该方法，请添加引用:

```
@misc{zheng2023judging,
      title={Judging LLM-as-a-judge with MT-Bench and Chatbot Arena},
      author={Lianmin Zheng and Wei-Lin Chiang and Ying Sheng and Siyuan Zhuang
and Zhanghao Wu and Yonghao Zhuang and Zi Lin and Zhuohan Li and Dacheng Li and
Eric. P Xing and Hao Zhang and Joseph E. Gonzalez and Ion Stoica},
      year={2023},
      eprint={2306.05685},
      archivePrefix={arXiv},
      primaryClass={cs.CL}
}
@misc{2023opencompass,
    title={OpenCompass: A Universal Evaluation Platform for Foundation Models},
    author={OpenCompass Contributors},
    howpublished = {\url{https://github.com/open-compass/opencompass}},
    year={2023}
}
```

实战：AlignBench 主观评测

数据集准备

```
mkdir -p ./data/subjective/

cd ./data/subjective
git clone https://github.com/THUDM/AlignBench.git

# data format conversion
python ../../../tools/convert_alignmentbench.py --mode json --jsonl
data/data_release.jsonl
```

配置文件

请根据需要修改配置文件 `configs/eval_subjective_alignbench.py`

启动评测

按如下方式执行命令后，将会开始答案推理和主观打分，如只需进行推理，可以通过制定`-m infer`实现

```
HF_EVALUATE_OFFLINE=1 HF_DATASETS_OFFLINE=1 TRANSFORMERS_OFFLINE=1 python run.py
configs/eval_subjective_alignbench.py
```

提交官方评测（Optional）

完成评测后，如需提交官方榜单进行评测，可以使用它`tools/convert_alignmentbench.py`进行格式转换。

- 请确保已完成推理，并获得如下所示的文件:

```
outputs/
└── 20231214_173632
    ├── configs
    ├── logs
    ├── predictions # 模型回复
    ├── results
    └── summary
```

- 执行如下命令获得可用于提交的结果

```
python tools/convert_alignmentbench.py --mode csv --exp-folder
outputs/20231214_173632
```

- 进入`submission`文件夹获得可用于提交的`.csv`文件

```
outputs/
└── 20231214_173632
    ├── configs
    ├── logs
    ├── predictions
    ├── results
    ├── submission # 可提交文件
    └── summary
```

数据污染评估

数据污染评估简介

数据污染 是指本应用在下游测试任务重的数据出现在了大语言模型 (LLM) 的训练数据中，从而导致在下游任务 (例如，摘要、自然语言推理、文本分类) 上指标虚高，无法反映模型真实泛化能力的现象。

由于数据污染的源头是出现在 LLM 所用的训练数据中，因此最直接的检测数据污染的方法就是将测试数据与训练数据进行碰撞，然后汇报两者之间有多少语料是重叠出现的，经典的 GPT-3 论文中的表 C.1 会报告了相关内容。

但如今开源社区往往只会公开模型参数而非训练数据集，在此种情况下 如何判断是否存在数据污染问题或污染程度如何，这些问题还没有被广泛接受的解决方案。OpenCompass 提供了两种可能的解决方案。

实验评估步骤

https://opencompass-cn.readthedocs.io/zh-cn/latest/advanced_guides/contamination_eval.html

大海捞针（待实践）

大海捞针测试简介

大海捞针测试（灵感来自 NeedleInAHaystack）是指通过将关键信息随机插入一段长文本的不同位置，形成大语言模型 (LLM) 的Prompt，通过测试大模型是否能从长文本中提取出关键信息，从而测试大模型的长文本信息提取能力的一种方法，可反映LLM长文本理解的基本能力。

数据集介绍

Skywork/ChineseDomainModelingEval 数据集收录了 2023 年 9 月至 10 月期间发布的高质量中文文章，涵盖了多个领域。这些文章确保了公平且具有挑战性的基准测试。 该数据集包括特定领域的文件：

- zh_finance.jsonl 金融
- zh_game.jsonl 游戏
- zh_government.jsonl 政务
- zh_movie.jsonl 电影
- zh_tech.jsonl 技术
- zh_general.jsonl 综合 这些文件用于评估LLM对不同特定领域的理解能力。

实验评估步骤

https://opencompass.readthedocs.io/zh-cn/latest/advanced_guides/needleinahaystack_eval.html

进阶作业 （待更新）

将自定义数据集提交至OpenCompass官网

提交地址：[https://hub.opencompass.org.cn/dataset-submit?lang=object%20Object]
提交指南：https://mp.weixin.qq.com/s/_s0a9nYRye0bmqVdwXRVCg
Tips：不强制要求配置数据集对应榜单（ leaderboard.xlsx ），可仅上传 EADME_OPENCOMPASS.md 文档