

PR #29215 完整报告

sgl-project/sglang

[bench] Add agentic-trace multi-turn dataset to bench_serving

合并时间: 2026-07-07 10:45

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29215>

执行摘要

- 一句话: 为 bench_serving 新增 agentic-trace 多轮数据集加载
- 推荐动作: 建议阅读 AgenticTraceDataset 的实现, 这是为 bench_serving 添加新数据集的参考范例。重点关注 from_args 工厂方法和 load 方法中偏移量取模、最大轮数截断等逻辑。

功能与动机

bench_serving 已经支持多轮回放, 但缺少预构建 agentic traces 的数据集加载器, 以便对多轮 agentic 工作负载进行端到端基准测试。

实现拆解

实现分为以下步骤:

1. 新增 **AgenticTraceDataset** 类 (agentic_trace.py): 继承 BaseDataset, 通过 from_args 解析 CLI 参数, load 方法读取 JSON 文件, 提取 conversations 列表, 支持 offset 旋转对话顺序和 max_turns 截断轮数, 最终生成 DatasetRow 列表。每轮输出长度默认为 220。
2. 注册数据集 (__init__.py): 导入 AgenticTraceDataset 并添加到 DATASET_MAPPING 字典, 映射到 'agentic-trace' 键。
3. 扩展 CLI 参数 (serving.py): 在 --dataset-name 的 choices 中增加 'agentic-trace', 并新增 --dataset-offset (旋转偏移量) 和 --agentic-max-turns (最大轮数限制) 两个可选参数。
4. 添加单元测试 (test_benchmark_datasets_api.py): 增加三个测试: 基础采样、偏移和最大轮数组合、无效输入异常。使用临时 JSON 文件模拟 agentic trace 数据。
5. 更新文档 (bench_serving.mdx): 在数据集列表中增加 agentic-trace 说明, 并详述相关 flags。

关键文件:

- python/sglang/benchmark/datasets/agentic_trace.py (模块 基准测试; 类别 source; 类型 core-logic; 符号 AgenticTraceDataset, from_args, load): 核心数据集加载器, 定义了 AgenticTraceDataset 类及加载逻辑。

- test/registered/bench_fn/test_benchmark_datasets_api.py (模块 测试; 类别 test; 类型 test-coverage; 符号 `_write_agentic_trace_json`, `test_agentic_trace_sampler`, `test_agentic_trace_offset_and_max_turns`, `test_agentic_trace_invalid_input_raises`) : 新增 agentic-trace 数据集加载器的单元测试, 覆盖正常采样、偏移和最大轮数、异常输入。
- python/sglang/benchmark/serving.py (模块 基准测试; 类别 source; 类型 core-logic) : CLI 入口, 新增 dataset-name 选项和两个新参数。
- python/sglang/benchmark/datasets/__init__.py (模块 基准测试; 类别 source; 类型 dependency-wiring) : 数据集注册入口, 将 agentic-trace 映射到 AgenticTraceDataset。
- docs_new/docs/developer_guide/bench_serving.mdx (模块 文档; 类别 other; 类型 documentation) : 更新文档, 说明 agentic-trace 数据集的用法和 flags。

关键符号: `AgenticTraceDataset.from_args`, `AgenticTraceDataset.load`, `test_agentic_trace_sampler`, `test_agentic_trace_offset_and_max_turns`

关键源码片段

python/sglang/benchmark/datasets/agentic_trace.py

核心数据集加载器, 定义了 `AgenticTraceDataset` 类及加载逻辑。

```
import json
import os
from argparse import Namespace
from dataclasses import dataclass
from typing import List, Optional

import numpy as np
from transformers import PreTrainedTokenizerBase

from sglang.benchmark.datasets.common import BaseDataset, DatasetRow

# 默认每轮输出长度, 匹配 OpenHands 风格 trace 的平均 Assistant 回复长度
DEFAULT_AGENTIC_OUTPUT_LEN = 220

@dataclass
class AgenticTraceDataset(BaseDataset):
    """Multi-turn agentic trace loader (e.g. OpenHands / SWE-smith traces)."""

    dataset_path: str
    num_requests: int
    fixed_output_len: Optional[int]
    offset: int
    max_turns: Optional[int]

    @classmethod
    def from_args(cls, args: Namespace) -> 'AgenticTraceDataset':
        # 从命令行参数构造实例, 映射各个 CLI flag
```

```

return cls(
    dataset_path=args.dataset_path,
    num_requests=args.num_prompts,
    fixed_output_len=args.sharegpt_output_len,
    offset=args.dataset_offset,
    max_turns=args.agentic_max_turns,
)

def load(
    self, tokenizer: PreTrainedTokenizerBase, model_id=None
) -> List[DatasetRow]:
    if not os.path.isfile(self.dataset_path):
        raise FileNotFoundError(f'Dataset not found at {self.dataset_path}')

    with open(self.dataset_path, 'r', encoding='utf-8') as f:
        data = json.load(f)

    conversations = data.get('conversations', [])
    if not conversations:
        raise ValueError(f'No conversations found in {self.dataset_path}.')

    # 计算偏移量（取模）并旋转对话列表，使不同 sweep step 从不同起点开始
    offset = self.offset % len(conversations)
    if offset:
        conversations = conversations[offset:] + conversations[:offset]

    # 确定每轮输出长度，优先使用用户指定值，否则用默认值 220
    output_len = self.fixed_output_len or DEFAULT_AGENTIC_OUTPUT_LEN

    filtered_dataset: List[DatasetRow] = []
    for conversation in conversations:
        if self.num_requests > 0 and len(filtered_dataset) >= self.num_requests:
            break

        # 提取每轮的消息列表，过滤掉空轮
        prompt = [turn['messages'] for turn in conversation if turn.get('messages')]
        if self.max_turns:
            prompt = prompt[: self.max_turns]
        if not prompt:
            continue

        # 仅用于记录，实际多轮回放忽略此 prompt_len
        prompt_len = int(conversation[0].get('prompt_tokens', 0))

        filtered_dataset.append(
            DatasetRow(
                prompt=prompt,
                prompt_len=prompt_len,
                output_len=output_len,
            )
        )

```

```

    )
)

if not filtered_dataset:
    raise ValueError(
        f'No usable conversations loaded from {self.dataset_path}.'
    )

# 打印摘要信息
num_turns = [len(row.prompt) for row in filtered_dataset]
print(
    f'#Conversations: {len(filtered_dataset)} '
    f'(offset={offset}, turns/conv min={min(num_turns)} '
    f'max={max(num_turns)} avg={np.mean(num_turns):.1f})'
)
print(f'#Output tokens per turn: {output_len}')
return filtered_dataset

```

test/registered/bench_fn/test_benchmark_datasets_api.py

新增 agentic-trace 数据集加载器的单元测试，覆盖正常采样、偏移和最大轮数、异常输入。

```

def _write_agentic_trace_json(self):
    trace = {
        'metadata': {'source': 'test'},
        'conversations': [
            [ # 第一个 conversation, 包含三个 turn (第三个为空)
                {
                    'messages': [
                        {'role': 'system', 'content': 'You are an agent.'},
                        {'role': 'user', 'content': 'Fix the bug.'},
                    ],
                    'prompt_tokens': 100,
                },
                {
                    'messages': [{'role': 'user', 'content': 'Tool output: ok.'}],
                    'prompt_tokens': 200,
                },
                {'messages': []}, # 空 turn, 应被跳过
            ],
            [ # 第二个 conversation, 仅一个 turn
                {
                    'messages': [{'role': 'user', 'content': 'Run the tests.'}],
                    'prompt_tokens': 50,
                },
            ],
        ],
    }
    path = self.tmpdir_path / 'agentic_trace.json'
    with open(path, 'w') as f:

```

```

        json.dump(trace, f)
    return str(path)

def test_agentic_trace_sampler(self):
    dataset_path = self._write_agentic_trace_json()
    args = make_args(
        dataset_name='agentic-trace',
        dataset_path=dataset_path,
        num_prompts=10,
    )
    dataset = AgenticTraceDataset.from_args(args)
    rows = dataset.load(self.tokenizer)
    self.assertEqual(len(rows), 2) # 两个 conversation
    self.assertTrue(all(isinstance(row, DatasetRow) for row in rows))
    # 未指定 output_len, 使用默认值 220
    self.assertTrue(
        all(row.output_len == DEFAULT_AGENTIC_OUTPUT_LEN for row in rows)
    )
    # 第一个 conversation 的第三个空 turn 被跳过, 因此只有 2 个有效 turn
    self.assertEqual(len(rows[0].prompt), 2)
    # 第二个 conversation 只有 1 个 turn
    self.assertEqual(len(rows[1].prompt), 1)
    self.assertEqual(rows[0].prompt[0][0]['role'], 'system')
    self.assertEqual(rows[0].prompt_len, 100)
    self.assertEqual(rows[1].prompt_len, 50)

def test_agentic_trace_offset_and_max_turns(self):
    dataset_path = self._write_agentic_trace_json()
    args = make_args(
        dataset_name='agentic-trace',
        dataset_path=dataset_path,
        num_prompts=10,
        sharegpt_output_len=64,
        dataset_offset=1, # 偏移 1, 旋转后第二个 conversation 成为第一个
        agentic_max_turns=1, # 每个 conversation 只取第一轮
    )
    dataset = AgenticTraceDataset.from_args(args)
    rows = dataset.load(self.tokenizer)
    self.assertEqual(len(rows), 2) # 仍有两个 conversation
    # 偏移后第一个 conversation 应为原本的第二个 (单 turn)
    self.assertEqual(len(rows[0].prompt), 1)
    # 限制最大 1 轮, 所以第二个 conversation (原第一个) 也只有 1 轮
    self.assertEqual(len(rows[1].prompt), 1)
    self.assertEqual(rows[0].prompt_len, 50)
    self.assertEqual(rows[0].output_len, 64) # 使用指定的 output_len

```

评论区精华

无实质技术讨论。Reviewer [zijiexia](#) 直接批准，作者 [kpham-sgl](#) 评论 'Should be safe to merge'。

- PR 审核与合并决策 (other): PR 被批准并合并。

风险与影响

- 风险：该 PR 仅影响 benchmark 工具链，不改变任何推理代码路径，回归风险较低。主要风险在于 JSON 输入格式异常处理，已通过单元测试覆盖空对话、无效文件等场景。
- 影响：对用户：新增 benchmark 数据集选项，允许对多轮 agentic 工作负载进行端到端性能测试。对系统：无影响。对团队：扩展了 benchmark 框架，易于添加更多数据集。
- 风险标记：benchmark-only, 新增数据集，无回归风险

关联脉络

- 暂无明显关联 PR