PR #42648 完整报告

vllm-project/vllm

Add HumanEval and GSM8K benchmarks to datasets

合并时间：2026-05-16 04:01

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42648

执行摘要

- 一句话：添加 HumanEval 和 GSM8K 基准测试数据集
- 推荐动作：对于需要扩展基准数据集的开发者，这是一个良好的参考实例，展示了如何遵循现有模式添加 HuggingFace 数据集。

功能与动机

支持对代码生成和数学推理场景的基准测试，完善 vllm 基准测试数据集覆盖。

实现拆解

1. 新增数据集类：在 vllm/benchmarks/datasets/datasets.py 文件中添加 HumanEvalDataset 和 GSM8KDataset 两个类，继承自 HuggingFaceDataset，并设置 DEFAULT_OUTPUT_LEN=256 和对应的 SUPPORTED_DATASET_PATHS。
2. 实现 sample 方法：每个类都实现了 sample 方法，从 HuggingFace 数据集中读取数据，应用聊天模板，生成 SampleRequest 列表。
3. 注册数据集：在 get_samples 函数中增加 elif 分支，匹配新数据集路径时设置对应的 dataset_class 和默认的 hf_split/hf_subset。
4. 更新文档：在 docs/benchmarking/cli.md 中添加两个新数据集的表格行和命令行示例。

关键文件：

- vllm/benchmarks/datasets/datasets.py（模块 基准工具；类别 source；类型 core-logic；符号 HumanEvalDataset, GSM8KDataset, get_samples）：核心变更文件，新增两个数据集类并注册到 get_samples 函数。
- docs/benchmarking/cli.md（模块 文档；类别 docs；类型 documentation）：添加新数据集的文档说明和命令行示例。

关键符号：HumanEvalDataset.sample, GSM8KDataset.sample

关键源码片段

vllm/benchmarks/datasets/datasets.py

核心变更文件，新增两个数据集类并注册到 get_samples 函数。

```
class HumanEvalDataset(HuggingFaceDataset):
    """
```

```
    HumanEval Dataset.
    https://huggingface.co/datasets/openai/openai_humaneval
    """
    DEFAULT_OUTPUT_LEN = 256
    SUPPORTED_DATASET_PATHS = {"openai/openai_humaneval"}

    def sample(
        self,
        tokenizer: TokenizerLike,
        num_requests: int,
        request_id_prefix: str = "",
        no_oversample: bool = False,
        output_len: int | None = None,
        enable_multimodal_chat: bool = False,
        skip_chat_template: bool = False,
        **kwargs,
    ) -> list[SampleRequest]:
        # 使用默认输出长度，若未指定
        output_len = output_len if output_len is not None else self.DEFAULT_OUTPUT_LEN
        sampled_requests: list[SampleRequest] = []

        for i, item in enumerate(self.data):
            if len(sampled_requests) >= num_requests:
                break
            prompt = item["prompt"] # HumanEval 数据集中每条记录包含 prompt 字段

            # 应用聊天模板
            if not skip_chat_template:
                prompt = tokenizer.apply_chat_template(
                    [{"role": "user", "content": prompt}],
                    add_generation_prompt=True,
                    tokenize=False,
                )

            prompt_len = len(tokenizer(prompt).input_ids)
            sampled_requests.append(
                SampleRequest(
                    prompt=prompt,
                    prompt_len=prompt_len,
                    expected_output_len=output_len,
                    request_id=request_id_prefix + str(i),
                )
            )
        self.maybe_oversample_requests(
            sampled_requests, num_requests, request_id_prefix, no_oversample
        )
        return sampled_requests

class GSM8KDataset(HuggingFaceDataset):
```

```
"""
GSM8K Dataset.
https://huggingface.co/datasets/openai/gsm8k
"""
DEFAULT_OUTPUT_LEN = 256
SUPPORTED_DATASET_PATHS = {"openai/gsm8k"}

def sample(
    self,
    tokenizer: TokenizerLike,
    num_requests: int,
    request_id_prefix: str = "",
    no_oversample: bool = False,
    output_len: int | None = None,
    enable_multimodal_chat: bool = False,
    skip_chat_template: bool = False,
    **kwargs,
) -> list[SampleRequest]:
    output_len = output_len if output_len is not None else self.DEFAULT_OUTPUT_LEN
    sampled_requests: list[SampleRequest] = []

    for i, item in enumerate(self.data):
        if len(sampled_requests) >= num_requests:
            break
        # GSM8K 数据集中使用 question 字段作为输入
        prompt = item["question"]

        if not skip_chat_template:
            prompt = tokenizer.apply_chat_template(
                [{"role": "user", "content": prompt}],
                add_generation_prompt=True,
                tokenize=False,
            )

        prompt_len = len(tokenizer(prompt).input_ids)
        sampled_requests.append(
            SampleRequest(
                prompt=prompt,
                prompt_len=prompt_len,
                expected_output_len=output_len,
                request_id=request_id_prefix + str(i),
            )
        )
    self.maybe_oversample_requests(
        sampled_requests, num_requests, request_id_prefix, no_oversample
    )
    return sampled_requests
```

评论区精华

Gemini Code Assist bot 提出了 5 条高优先级建议，涉及硬编码覆盖用户参数、docstring 错误、返回类型提示不一致。所有问题均在最终提交中得到修复。

- 硬编码 hf_split 覆盖用户参数 (correctness): 最终提交中已修改为 args.hf_split = args.hf_split if args.hf_split else 'test' 的形式。
- Docstring 错误（MT-Bench 误用） (documentation): 最终 docstring 已更正为 'HumanEvalDataset Dataset.'。
- 返回类型提示不一致 (style): 最终提交中返回值注释已统一为 list[SampleRequest]。

风险与影响

- 风险：无显著技术风险。新增代码仅扩展已有框架，不修改现有逻辑。
- 影响：用户可直接使用 --dataset-path openai/openai_humaneval 或 openai/gsm8k 进行基准测试；对系统核心功能无影响。
- 风险标记：缺少测试覆盖

关联脉络

- 暂无明显关联 PR