

PR #47590 完整报告

vllm-project/vllm

[Bugfix][Pooling] Forward instruction to Jina reranker scoring prompts

合并时间: 2026-07-05 13:39

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47590>

执行摘要

- 一句话: 修复 Jina reranker 评分时 instruction 被忽略
- 推荐动作: 值得精读的设计决策是将 sanitize_input 内联化以减少耦合; 但需注意未解决的安全建议。开发者可关注后续是否引入循环清理。

功能与动机

Jina reranker 路径构建 prompt 时未将 instruction 传入现有 formatter, 导致 instruction 请求静默忽略, 评分结果与默认相同。

实现拆解

1. 在 vllm/entrypoints/pooling/scoring/io_processor.py 的 _pre_process 方法中, 从 prompt_extras 提取 instruction (位于 chat_template_kwargs 内)。
2. 将该 instruction 传递给 format_docs_prompts_func, 并在该函数内部对 instruction 也执行特殊 token 清洗 (将 sanitize_input 从类方法改为内部嵌套函数, 避免在共享 score 路径中引入 Jina 专有引用)。
3. 新增 examples/pooling/token_embed/jina_reranker_v3_online.py, 展示发送带 instruction 的 /score 和 /rerank 请求。
4. 扩展 tests/models/language/pooling/test_jina_reranker_v3.py, 添加 _get_score_response、_get_rerank_response 辅助函数和 _test_online_instruction 测试用例, 验证 instruction 确实改变 prompt token 数和分数。

关键文件:

- vllm/entrypoints/pooling/scoring/io_processor.py (模块 评分 IO; 类别 source; 类型 core-logic; 符号 format_docs_prompts_func, _pre_process, sanitize_input) : 核心 bugfix 所在: 从 prompt_extras 提取 instruction 并传递给 formatter, 同时将 sanitize_input 重构为内部函数。
- tests/models/language/pooling/test_jina_reranker_v3.py (模块 Jina 测试; 类别 test; 类型 test-coverage; 符号 _get_scores, _get_score_response, _get_rerank_response, _test_online_instruction) : 增加 instruction 测试覆盖: 通过 _get_score_response 和 _get_rerank_response 传递 instruction, 验证 token 数变化和分数差异。
- examples/pooling/token_embed/jina_reranker_v3_online.py (模块 示例; 类别 source; 类型 example; 符号 post_http_request, print_response, parse_args, main) : 新增示例

展示 instruction 的使用方式，便于用户快速上手。

关键符号：format_docs_prompts_func, _pre_process, sanitize_input, _get_score_response, _get_rerank_response, _test_online_instruction, main

关键源码片段

vllm/entrypoints/pooling/scoring/io_processor.py

核心 bugfix 所在：从 prompt_extras 提取 instruction 并传递给 formatter，同时将 sanitize_input 重构为内部函数。

```
# vllm/entrypoints/pooling/scoring/io_processor.py 关键变更：
# format_docs_prompts_func 新增 instruction 参数，并在内部清洗。
# sanitize_input 从静态方法改为嵌套函数，避免共享路径引用 Jina。
```

```
class JinaRankingIOProcessorMixin:
    @staticmethod
    def format_docs_prompts_func(
        query: str,
        docs: list[str],
        special_tokens: dict[str, str] | None = None,
        instruction: str | None = None, # 新增 instruction 参数
        no_thinking: bool = True,
    ) -> str:
        default_special_tokens = {
            "query_embed_token": "<lquery_token>",
            "doc_embed_token": "<lembed_token>",
        }
        if special_tokens is None:
            special_tokens = default_special_tokens

        # 嵌套 sanitize_input，避免在共享路径中使用 JinaRankingIOProcessorMixin
        def sanitize_input(text: str) -> str:
            for token in special_tokens.values():
                text = text.replace(token, "")
            return text

        query = sanitize_input(query)
        docs = [sanitize_input(doc) for doc in docs]

        prefix = (省略固定 system 提示...)
        suffix = (省略固定 assistant 标记...)

        prompt = (
            f"I will provide you with {len(docs)} passages..."
        )

        if instruction:
            # instruction 也需要清洗特殊 token，防止污染
```

```

        instruction = sanitize_input(instruction)
        prompt += f"<instruct>\n{instruction}\n</instruct>\n"

    doc_prompts = [...]
    prompt += [...]
    return prefix + prompt + suffix

# _pre_process 中提取 instruction 示例:
def _pre_process(self, scoring_data, tok_params, prompt_extras=None):
    queries = self.ensure_str(scoring_data.data_1)
    docs = self.ensure_str(scoring_data.data_2)
    chat_template_kwargs = (
        prompt_extras.get("chat_template_kwargs") if prompt_extras else None
    )
    instruction = (
        chat_template_kwargs.get("instruction") if chat_template_kwargs else None
    )
    # 将 instruction 传递给 format_docs_prompts_func
    prompts = [
        self.format_docs_prompts_func(
            query=queries[0], docs=docs, instruction=instruction
        )
    ]
    return self._preprocess_cmpl_offline(...)

```

tests/models/language/pooling/test_jina_reranker_v3.py

增加 instruction 测试覆盖：通过 `_get_score_response` 和 `_get_rerank_response` 传递 instruction，验证 token 数变化和分数差异。

tests/models/language/pooling/test_jina_reranker_v3.py 新增辅助函数

```
from vllm.entrypoints.pooling.scoring.protocol import RerankResponse, ScoreResponse
```

统一的 score 请求封装，通过 `**extra_body` 支持 instruction 等额外参数

```

def _get_score_response(server, query, document, **extra_body):
    payload = {
        "model": model_name,
        "queries": query,
        "documents": document,
    }
    payload.update(extra_body) # 合并 instruction 或 chat_template_kwargs
    score_response = requests.post(
        server.url_for("score"),
        json=payload,
    )
    score_response.raise_for_status()
    return ScoreResponse.model_validate(score_response.json())

```

统一的 rerank 请求封装，同样支持 `**extra_body`

```

def _get_rerank_response(server, query, document, **extra_body):
    payload = {
        "model": model_name,
        "query": query,
        "documents": document,
    }
    payload.update(extra_body) # 合并 instruction 等
    rerank_response = requests.post(
        server.url_for("rerank"),
        json=payload,
    )
    rerank_response.raise_for_status()
    return RerankResponse.model_validate(rerank_response.json())

```

测试 instruction 是否实际生效

```

def _test_online_instruction(server):
    docs = documents[:2]
    default_score = _get_score_response(server, query, docs)
    instruction_score = _get_score_response(
        server, query, docs, instruction=INSTRUCTION,
    )
    kwargs_score = _get_score_response(
        server, query, docs,
        chat_template_kwargs={"instruction": INSTRUCTION},
    )
    # instruction 应增加 prompt token 数
    assert instruction_score.usage.prompt_tokens > default_score.usage.prompt_tokens
    assert kwargs_score.usage.prompt_tokens == instruction_score.usage.prompt_tokens
    # instruction 应改变分数
    assert [d.score for d in instruction_score.data] != [d.score for d in default_score.data]

```

examples/pooling/token_embed/jina_reranker_v3_online.py

新增示例展示 instruction 的使用方式，便于用户快速上手。

examples/pooling/token_embed/jina_reranker_v3_online.py

示例：通过 instruction 参数影响评分结果

```

def main(args):
    score_url = f"http://{args.host}:{args.port}/score"
    rerank_url = f"http://{args.host}:{args.port}/rerank"
    model_name = args.model

    query = "Which passage is about sports?"
    documents = [
        "Basketball is played by two teams on a court.",
        "Green tea contains antioxidants and may support metabolism.",
    ]
    instruction = "Rank passages about sports higher than passages about nutrition."

    # 在 score 请求中直接传入 instruction

```

```
score_prompt = {
    "model": model_name,
    "queries": query,
    "documents": documents,
    "instruction": instruction,
}
score_response = post_http_request(prompt=score_prompt, api_url=score_url)
print_response("Score", score_prompt, score_response)

# 在 rerank 请求中同样支持 instruction
rerank_prompt = {
    "model": model_name,
    "query": query,
    "documents": documents,
    "instruction": instruction,
}
rerank_response = post_http_request(prompt=rerank_prompt, api_url=rerank_url)
print_response("Rerank", rerank_prompt, rerank_response)
```

评论区精华

- nooooop指出“所有评分类型模型都经过此路径，应避免在此处引入 JinaRankingIOProcessorMixin 相关逻辑”。作者采纳建议，将 `sanitize_input` 从类方法改为 `format_docs_prompts_func` 内的嵌套函数。
- depthfirst-app[bot]提示低严重性问题：嵌套 `sanitize_input` 使用单次 `str.replace`，可能被递归 token 模式绕过（如 `<|<lembed_token|>embed_token|>`），导致特殊 token 注入、分数偏差或 API 崩溃。该建议未被解决。
- 避免在通用评分路径中引入 Jina 特定逻辑 (design)：作者将 `sanitize_input` 从类方法改为 `format_docs_prompts_func` 内部的嵌套函数，避免在共享路径中引用 `JinaRankingIOProcessorMixin`。
- 单次 `replace` 的 token 注入风险 (security)：未在本次 PR 中解决，可能被忽略或留待后续。

风险与影响

- 风险：
 1. 安全风险：`sanitize_input` 单次 `replace` 可能被递归 token 绕过，存在特殊 token 注入风险，虽为 LOW 但应关注。
 2. 回归风险：变更仅在 Jina 评分路径生效，不影响其他评分模型；`_pre_process` 中提取 `chat_template_kwargs` 对其他模型无影响，风险低。
 3. 测试覆盖：新增测试验证了 `instruction` 行为，但未覆盖递归 token 注入场景。
- 影响：
 - 用户：Jina reranker 用户现在可以正确使用 `instruction` 参数，任务指令实际作用于评分。
 - 系统：无性能影响，仅增加极少数逻辑判断。
 - 团队：需评估 `residual` 注入风险，建议后续改用循环清理。

- 风险标记：未解决的 token 注入风险，共享路径耦合权衡

关联脉络

- PR #47082 [Misc] Preserve cross-encoder pooling extra kwargs: 同为 pooling 评分参数传递修复，解决 extra_kwargs 丢失问题，与本次 instruction 未传递同属参数转发类 bugfix。
- PR #46966 [Misc] Validate Pooling cache_salt Values: 同为 pooling 前端修复，确保请求参数正确校验，与本次 instruction 参数正确传递属于同一改进方向。
- PR #46939 [Misc] Forward request-level prompt extras for cross-encoder scoring: 修复跨编码器打分时 prompt extras 丢失，与本次 instruction 丢失问题类似。