PR #42412 完整报告

vllm-project/vllm

[Feature] Add instruction support for score/rerank chat templates

合并时间：2026-05-14 09:41

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42412

执行摘要

- 一句话：为 score/rerank 端点添加 instruction 与 chat_template_kwargs 支持
- 推荐动作：值得精读。该 PR 是 scoring API 功能补齐的重要一步，展示了如何通过 Pydantic validator 组合字段、如何在预处理管道中引入新参数，以及如何设计向后兼容的 chat 模板。特别推荐给负责 entrypoints 和维护定制化 rerank 服务的开发者。

功能与动机

Issue #42391 报告 /v1/rerank 忽略 chat_template_kwargs，导致 Qwen3-Reranker 无法设置 per-task Instruct，迫使开发者修改源代码或手动拼接 prompt。本次变更使 API 支持 instruction 字段和 chat_template_kwargs，让模板渲染器能接收自定义指令。

实现拆解

1. 在请求 schema 中声明新字段：在 vllm/entrypoints/pooling/scoring/protocol.py 的 ScoringRequestMixin 中添加 instruction 和 chat_template_kwargs 字段，并通过 Pydantic 的 model_validator 将 instruction 自动合并到 chat_template_kwargs 中。
2. 修正在线预处理路径：在 BiEncoderIOProcessor.pre_process_online 和 CrossEncoderIOProcessor.pre_process_online 中，将键集合从 ('mm_processor_kwargs', 'cache_salt') 扩展为包含 'chat_template_kwargs'，确保字段能被提取到 prompt_extras 中。
3. 修正离线预处理路径：在 CrossEncoderIOProcessor.pre_process_offline 中（以及通过基类 _pre_process 的修改），从 ctx.pooling_params.extra_kwargs 提取 chat_template_kwargs 并传递给 _pre_process，统一在线 / 离线行为。
4. 深化渲染调用链：在 _pre_process 中新增 chat_template_kwargs 参数，传递给 get_score_prompt；get_score_prompt 再将其作为 **kwargs 传给 safe_apply_chat_template，同时预检查保留键名（chat_template, tools, tokenize）避免冲突。
5. 更新官方模板：为 qwen3_reranker.jinja 和 qwen3_vl_reranker.jinja 添加 instruction 和 instruct 的 fallback 链，并保持默认值不变，实现后向兼容。
6. 增加端到端测试：在 test_cross_encoder_online_vision.py 中新增三个测试用例，分别验证 /score 和 /rerank 的 instruction 字段生效，以及 instruction 与 chat_template_kwargs 的等价性。

关键文件：

- vllm/entrypoints/pooling/scoring/protocol.py（模块 评分入口；类别 source；类型 core-logic；符号 ScoringRequestMixin, _merge_instruction_into_kwargs）：核心变更点：在 ScoringRequestMixin 中定义了 instruction 和 chat_template_kwargs 字段，并通过 model_validator 自动合并，是功能入口。
- vllm/entrypoints/pooling/scoring/io_processor.py（模块 评分入口；类别 source；类型 core-logic；符号 BiEncoderIOProcessor.pre_process_online, CrossEncoderIOProcessor.pre_process_online, CrossEncoderIOProcessor.pre_process_offline, _pre_process）：实现在线 / 离线路径传递 chat_template_kwargs 到模板渲染，确保两端行为一致。
- tests/entrypoints/pooling/scoring/test_cross_encoder_online_vision.py（模块 跨编码器；类别 test；类型 test-coverage；符号 test_score_api_instruction_field, test_rerank_api_instruction_field, test_rerank_api_instruction_field_matches_chat_template_kwargs）：提供 e2e 测试，验证 instruction 字段在 score/rerank 端点中生效，并覆盖 instruction 与 chat_template_kwargs 的等价性。
- examples/pooling/score/template/qwen3_reranker.jinja（模块 模板文件；类别 other；类型 core-logic）：官方模板更新，支持 instruction/instruct 变量 fallback。
- examples/pooling/score/template/qwen3_vl_reranker.jinja（模块 模板文件；类别 other；类型 core-logic）：多模态版本模板同步更新。

关键符号：_merge_instruction_into_kwargs, pre_process_online, pre_process_offline, _pre_process, get_score_prompt, test_score_api_instruction_field, test_rerank_api_instruction_field, test_rerank_api_instruction_field_matches_chat_template_kwargs

关键源码片段

vllm/entrypoints/pooling/scoring/protocol.py

核心变更点：在 ScoringRequestMixin 中定义了 instruction 和 chat_template_kwargs 字段，并通过 model_validator 自动合并，是功能入口。

```
# vllm/entrypoints/pooling/scoring/protocol.py (partial)

class ScoringRequestMixin(PoolingBasicRequestMixin, ClassifyRequestMixin):
    # 新增顶层 instruction 字段，允许直接传入指令
    instruction: str | None = Field(
        default=None,
        description=(
            'Task instruction prepended to each scored pair via the chat '
            'template. Equivalent to passing '
            'chat_template_kwargs={"instruction": ...}.'
        ),
    )

    # 新增通用 chat_template_kwargs 字段，与 embedding/chat 端点一致
```

```
    chat_template_kwargs: dict[str, Any] | None = Field(
        default=None,
        description=(
            'Additional keyword args to pass to the chat template renderer. '
            'Will be accessible by the score/rerank chat template.'
        ),
    )

    @model_validator(mode='after')
    def _merge_instruction_into_kwargs(self) -> 'ScoringRequestMixin':
        '''Fold the top-level `instruction` field into `chat_template_kwargs`.
        If `chat_template_kwargs` already contains an `instruction` key,
        it is preserved (user-defined keys take precedence).
        This keeps the API consistent: callers can use either form.
        '''
        if self.instruction is not None:
            merged = dict(self.chat_template_kwargs or {})
            merged.setdefault('instruction', self.instruction)
            self.chat_template_kwargs = merged
        return self
```

vllm/entrypoints/pooling/scoring/io_processor.py

实现在线 / 离线路径传递 chat_template_kwargs 到模板渲染，确保两端行为一致。

```
# vllm/entrypoints/pooling/scoring/io_processor.py (partial)

def get_score_prompt(
    self,
    scoring_data: ScoringData,
    chat_template: str | None = None,
    max_tokens_per_query: int = 0,
    max_tokens_per_doc: int = 0,
    # 新增 chat_template_kwargs 参数，接收来自 prompt_extras 的 kwarg
    chat_template_kwargs: dict[str, Any] | None = None,
):
    # ... 其余处理 ...
    try:
        _safe_kwargs = chat_template_kwargs or {}
        # 安全校验：防止用户传入会与 safe_apply_chat_template 固定参数冲突的键
        _reserved = {'chat_template', 'tools', 'tokenize'}
        _unexpected = _reserved & _safe_kwargs.keys()
        if _unexpected:
            raise ValueError(
                'chat_template_kwargs contains reserved keys that '
                f'conflict with fixed scorer arguments: {_unexpected}'
            )
        full_prompt = safe_apply_chat_template(
            model_config, tokenizer,
            messages,
```

```
            chat_template=chat_template,
            tools=None,
            tokenize=False,
            **_safe_kwargs, # 用户自定义 kwargs 展开传入
        )
        # ...
    except ChatTemplateResolutionError:
        # fallback ...

# 在 _pre_process 中取出 chat_template_kwargs 并传递给 get_score_prompt
# 从 prompt_extras 中拿出键
engine_prompt, engine_prompt_tokens = scoring_dataset_builder(
    # ...
    chat_template_kwargs=prompt_extras.get('chat_template_kwargs') if prompt_extras else
    None,
)
```

tests/entrypoints/pooling/scoring/test_cross_encoder_online_vision.py

提供 e2e 测试，验证 instruction 字段在 score/rerank 端点中生效，并覆盖 instruction 与 chat_template_kwargs 的等价性。

```
# tests/entrypoints/pooling/scoring/test_cross_encoder_online_vision.py (new)

INSTRUCTION = (
    'Given a multimodal retrieval query, retrieve candidates that '
    'visually or textually match the requested scene, object, or action.'
)

@pytest.mark.asyncio
async def test_rerank_api_instruction_field_matches_chat_template_kwargs(
    server: tuple[RemoteOpenAIServer, str],
):
    '''验证通过顶层 instruction 字段与 chat_template_kwargs 传入相同指令
    产生的 prompt_tokens 完全一致，且大于默认值。
    '''
    remote_server, _ = server
    doc_list = [document, {'content': [documents[0]]}, {'content': [documents[1]]}, {'content':
    [documents[0], documents[1]]}]

    # 默认响应（无指令）
    default_resp = requests.post(remote_server.url_for('rerank'),
                                 json={'model': MODEL_NAME, 'query': query, 'documents': doc_list})
    default_resp.raise_for_status()
    default = RerankResponse.model_validate(default_resp.json())

    # 使用 instruction 字段
    field_resp = requests.post(remote_server.url_for('rerank'),
                               json={'model': MODEL_NAME, 'query': query, 'documents': doc_list,
                               'instruction': INSTRUCTION})
```

```
field_resp.raise_for_status()
field = RerankResponse.model_validate(field_resp.json())

# 使用 chat_template_kwargs
kwargs_resp = requests.post(remote_server.url_for('rerank'),
                            json={'model': MODEL_NAME, 'query': query, 'documents': doc_list,
                            'chat_template_kwargs': {'instruction': INSTRUCTION}})
kwargs_resp.raise_for_status()
kwargs = RerankResponse.model_validate(kwargs_resp.json())

# 等价性断言
assert field.usage.prompt_tokens == kwargs.usage.prompt_tokens
assert field.usage.prompt_tokens > default.usage.prompt_tokens
```

评论区精华

1. BiEncoder 与离线路径遗漏：gemini-code-assist[bot] 指出 chat_template_kwargs 仅添加在 CrossEncoder 在线路径，BiEncoder 在线路径和离线路径均未覆盖，会导致不一致行为。KrxGu 随后在迭代中修复。
2. 新增顶层 instruction 字段：noooop 在 review 中建议同时添加 instruction 作为便捷字段，使 API 更易用，并为 qwen3_vl_reranker.jinja 也补充支持。KrxGu 采纳并更新了模板。
3. 测试整合：noooop 建议将独立的测试文件合并到已有的 test_cross_encoder_online_vision.py 中，避免重复加载模型。KrxGu 移除了单独文件，将测试用例附加到现有集成测试中。
4. 模板新行导致 CI 失败：KrxGu 在评论中报告模板变更额外引入换行使 prompt tokens 计数增加，noooop 要求定位并修复，后通过调整模板解决了问题。

- BiEncoder 和离线路径遗漏 (correctness): KrxGu 在后续提交中添加了 BiEncoder 在线路径和 CrossEncoder 离线路径的 chat_template_kwargs 传递。
- 新增顶层 instruction 字段 (design): KrxGu 新增 instruction 字段并更新模板，测试合并到现有测试文件。
- 测试文件整合 (testing): KrxGu 移除了独立的测试文件，将测试追加到现有视觉模型测试中。
- 模板新行导致 prompt tokens 变化 (correctness): 已调整模板渲染空白语义，恢复了原来的 token 计数。

风险与影响

- 风险：
  1. 默认行为不变：模板 fallback 链确保未传入 instruction 时与旧行为一致，但模板渲染结果可能因 Jinja 语法变化产生微小差异（如空白控制），已通过检测 token 数测试验证。
1. 离线路径初始遗漏风险：初始实现仅覆盖在线路径，离线调用（如使用 LLM.score()）会静默丢弃 chat_template_kwargs，经 review 发现后补全。
2. TypeError 风险：用户可能在 chat_template_kwargs 中传入与 safe_apply_chat_template 签名冲突的键（chat_template, tools, tokenize），已添加显式检查并抛出 ValueError。

3. 向后兼容性：新增字段不会破坏现有 API，但旧客户端若误传 instruction 或 chat_template_kwargs 过去无效，现在会生效，仅影响新请求。 - 影响：
    1. 用户影响：使用 scoring/rerank API 的开发者现在可以自定义指令文本，对 Qwen3-Reranker 类模型至关重要；API 文档和 schema 自动生效。
4. 系统影响：无性能或资源影响；仅增加字段校验和透传。
5. 团队影响：后续新增 chat template data 模型需参考此实现；维护成本低。
6. 影响范围：所有通过 /v1/rerank、/v1/score 或将来的 scoring 端点发起的在线请求，以及 LLM.score() 等离线调用。BiEncoder 和 CrossEncoder 均受益。 - 风险标记：BiEncoder 与离线路径初始遗漏，模板渲染空白控制敏感，chat_template_kwargs 键冲突风险

关联脉络

- 暂无明显关联 PR