

PR #30832 完整报告

sgl-project/sglang

Add 'anyOf' schema support for qwen3_coder tool call parser

合并时间: 2026-07-23 05:14

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/30832>

执行摘要

- 一句话: 支持 anyOf schema 解析 qwen3_coder 工具调用参数
- 推荐动作: 值得精读。展示了如何通过复用通用 schema 推断函数扩展专用解析器的类型支持, 代码改动简洁、测试充分, 是低风险修复的范例。

功能与动机

PR 作者发现 qwen3_coder 工具调用参数解析器无法解析可选类型 (如 `list[str] | None`) 的参数, 导致参数被错误地解析为字符串。'I noticed the qwen3_coder tool call parameter parser couldn't parse optional types in the tool call. I had a parameter of type `list[str] | None` which was getting parsed as a str.'

实现拆解

1. 添加 `_get_param_type` 方法: 在 `python/sglang/srt/function_call/qwen3_coder_detector.py` 中新增 `_get_param_type` 方法, 调用已有工具函数 `infer_type_from_json_schema` 从参数 `schema` 中推断类型, 若推断结果为 `None` 则回退为 `"string"`。
2. 修改 `_convert_param_value`: 将原本直接读取 `param_config[param_name]["type"]` 的逻辑替换为调用 `_get_param_type`, 使参数类型转换支持 `anyOf` 等复杂 schema。
3. 增强 `infer_type_from_json_schema`: 在 `python/sglang/srt/function_call/utils.py` 中为 `anyOf/oneOf` 分支添加可选类型检测: 若推断出的类型集合中恰有两个类型且包含 `"null"`, 则返回另一非空类型 (而非回退为 `"string"`)。
4. 补充测试: 在 `test/registered/unit/function_call/test_function_call_parser.py` 中新增 4 个测试用例: `test_anyof_array_parameter_conversion`、`test_anyof_array_parameter_conversion_null`、`test_streaming_anyof_array_parameter_conversion` 和 `test_nested_anyof_array_with_multiple_types_parameter_conversion`, 覆盖数组、null、流式及嵌套多类型的场景。

关键文件:

- `python/sglang/srt/function_call/qwen3_coder_detector.py` (模块 工具调用; 类别 source; 类型 core-logic; 符号 `_get_param_type`): 核心修复文件, 新增 `_get_param_type` 方法, 并修改 `_convert_param_value` 使用通用类型推断。

- python/sglang/srt/function_call/utils.py (模块 工具调用; 类别 source; 类型 core-logic)
: 通用 schema 类型推断函数 infer_type_from_json_schema 增强, 增加对可选类型 (null) 的处理分支。
- test/registered/unit/function_call/test_function_call_parser.py (模块 工具调用; 类别 test; 类型 test-coverage; 符号 test_anyof_array_parameter_conversion, test_anyof_array_parameter_conversion_null, test_streaming_anyof_array_parameter_conversion, test_nested_anyof_array_with_multiple_types_parameter_conversion) : 新增 4 个测试用例覆盖 anyOf schema 的数组、null、流式及多类型嵌套场景。

关键符号: `_get_param_type`, `infer_type_from_json_schema`, `_convert_param_value`

关键源码片段

python/sglang/srt/function_call/qwen3_coder_detector.py

核心修复文件, 新增 `_get_param_type` 方法, 并修改 `_convert_param_value` 使用通用类型推断。

```
# python/sglang/srt/function_call/qwen3_coder_detector.py

# ... 省略其他导入
from sglang.srt.function_call.utils import infer_type_from_json_schema

def _get_param_type(self, param_schema: Any) -> str:
    """从 JSON schema 参数推断解析器转换类型。"""
    inferred_type = infer_type_from_json_schema(param_schema)
    # 如果推断结果为 None, 则回退为 "string"
    if inferred_type is None:
        return "string"
    return str(inferred_type).strip().lower()

def _convert_param_value(self, param_value, param_name, param_config, func_name):
    # ... 前面空值处理
    # 替换之前直接读取 param_config[param_name]["type"] 的方式
    param_type = self._get_param_type(param_config[param_name])
    if param_type in ["string", "str", "text", ...]:
        return param_value
    elif param_type.startswith("int"):
        # ... 整数转换
```

python/sglang/srt/function_call/utils.py

通用 schema 类型推断函数 `infer_type_from_json_schema` 增强, 增加对可选类型 (null) 的处理分支。

```
# python/sglang/srt/function_call/utils.py

# 在 infer_type_from_json_schema 函数中的 anyOf/oneOf 分支内 (约第 319 行) :
if types:
```

```
# 如果所有类型相同，返回统一类型
if len(set(types)) == 1:
    return types[0]
# 新增：如果是可选类型（正好有两个类型且包含 null），返回非 null 类型
if len(set(types)) == 2 and "null" in types:
    return [t for t in types if t != "null"][0]
# 当类型不同时，优先返回 string（最安全）
if "string" in types:
    return "string"
# 否则返回第一个类型
return types[0]
```

评论区精华

- `_get_param_type` 是否必要：reviewer alexnails 指出 `infer_type_from_json_schema` 内部已有字典类型检查，初始版本中 `_get_param_type` 重复了该检查。作者 [ilyasher-harmonic](#) 根据意见移除了冗余判断。
- 测试覆盖建议：alexnails 建议添加更极端的边缘测试，作者已补充 `test_nested_anyof_array_with_multiple_types_parameter_conversion`。
 - 是否需要 `_get_param_type` 中的类型检查 (design)：开发人员确认冗余并移除了该检查，最终版本直接调用 `infer_type_from_json_schema`。

风险与影响

- 风险：低风险。核心改动仅涉及参数类型推断路径，且已有单元验证。但 `infer_type_from_json_schema` 的增强可能影响其他调用方（如其他格式检测器），目前该函数仅被 `qwen3_coder_detector.py` 和新引入的 `_get_param_type` 使用，影响范围可控。
- 影响：对使用 `qwen3_coder` 工具调用的用户，解析器现在能正确处理 `anyOf` schema 定义 的参数，如 `list[str] | None` 不会再被误解析为字符串。对系统无性能影响，测试覆盖率提升。
- 风险标记：核心解析路径变更，测试覆盖不够全面

关联脉络

- PR #31975 Treat partial_json_parser AssertionError as incomplete JSON: 同文件 `python/sglang/srt/function_call/utils.py` 的此前变更，增强了 schema 推断逻辑。