

PR #26379 完整报告

sgl-project/sglang

Revert "fix(tool_call): normalize non-standard JSON Schema types in tool params"

合并时间: 2026-05-26 15:49

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/26379>

执行摘要

- 一句话: 回退工具参数 JSON Schema 类型规范化功能
- 推荐动作: 该 PR 为紧急回退, 变更简单明确, 无需精读。但建议关注后续对原功能的重新实现, 特别是增加充足的测试覆盖以避免 CI 问题。

功能与动机

原 PR #23476 为处理工具参数中的非标准 JSON Schema 类型 (如 `varchar`、`enum`、`int`) 添加了归一化函数, 但该改动长期未同步主分支并导致 CI 持续失败 (PR 评论中作者 hnyls2002 指出 'stales with the latest main for 2 weeks... Just bombed the CI...'), 因此决定回退以恢复主分支健康。

实现拆解

1. 删除测试文件: 移除 `test/registered/unit/function_call/test_normalize_json_schema_types.py` (391 行), 该文件包含 `TestNormalizeJsonSchemaTypes` 类及 7 个测试方法。
2. 删除核心逻辑: 在 `python/sglang/srt/function_call/utils.py` 中删除约 149 行, 包括常量 `_STANDARD_JSON_SCHEMA_TYPES`、别名映射 `_JSON_SCHEMA_TYPE_ALIASES`、前缀规则 `_PREFIX_RULES`, 以及三个函数 `_matches_type_prefix`、`_normalize_single_type`、`normalize_json_schema_types`。
3. 简化验证入口: 在 `python/sglang/srt/entrypoints/openai/serving_chat.py` 中移除对 `normalize_json_schema_types` 的导入和 `_validate_request` 中的调用 (包括对应的 `RecursionError` 处理), 恢复为仅调用 `Draft202012Validator.check_schema`。

关键文件:

- `test/registered/unit/function_call/test_normalize_json_schema_types.py` (模块 类型归一化; 类别 `test`; 类型 `deletion`; 符号 `TestNormalizeJsonSchemaTypes`, `_assert_accepts`, `test_enum_alias_becomes_string`, `test_varchar_alias_becomes_string`): 完整的单元测试文件被删除, 覆盖了所有归一化场景的测试用例, 表明回退彻底移除测试覆盖。
- `python/sglang/srt/function_call/utils.py` (模块 工具函数; 类别 `source`; 类型 `core-logic`; 符号 `_matches_type_prefix`, `_normalize_single_type`, `normalize_json_schema_types`): 删除了归一化核心逻辑: `normalize_json_schema_types`、`_normalize_single_type`、`_matches_type_prefix` 函数及相关常量定义。

- python/sglang/srt/entrypoints/openai/serving_chat.py (模块 服务端点; 类别 source; 类型 dependency-wiring) : 移除了 normalize_json_schema_types 的导入和调用, 恢复原有直接 Draft202012Validator.check_schema 验证。

关键符号: _matches_type_prefix, _normalize_single_type, normalize_json_schema_types

关键源码片段

test/registered/unit/function_call/test_normalize_json_schema_types.py

完整的单元测试文件被删除, 覆盖了所有归一化场景的测试用例, 表明回退彻底移除测试覆盖。

```
"""单元测试: 工具参数 JSON Schema 类型别名归一化。"""
import unittest
from jsonschema import Draft202012Validator, SchemaError
from sglang.srt.function_call.utils import normalize_json_schema_types

class TestNormalizeJsonSchemaTypes(unittest.TestCase):
    def _assert_accepts(self, schema: dict) -> None:
        # 验证归一化后的 schema 能通过 Draft 2020-12 校验
        Draft202012Validator.check_schema(schema)

    def test_enum_alias_becomes_string(self):
        schema = {
            "type": "object",
            "properties": {"color": {"type": "enum", "enum": ["red", "green", "blue"]}},
        }
        normalize_json_schema_types(schema)
        # 期望 "enum" 被映射为标准类型 "string"
        self.assertEqual(schema["properties"]["color"]["type"], "string")
        self._assert_accepts(schema)
```

python/sglang/srt/function_call/utils.py

删除了归一化核心逻辑: normalize_json_schema_types、_normalize_single_type、_matches_type_prefix 函数及相关常量定义。

```
# 非标准 type 别名映射 (常见于 DB/ORM 导出的 schema)
_JSON_SCHEMA_TYPE_ALIASES: Dict[str, str] = {
    "str": "string", "text": "string", "varchar": "string",
    "bool": "boolean", "binary": "boolean",
    "bigint": "integer", "smallint": "integer", "tinyint": "integer",
    "double": "number", "decimal": "number", "real": "number",
    "arr": "array", "tuple": "array", "set": "array",
    "map": "object",
}

# 前缀边界字符, 防止误匹配 (如 "int" 不应匹配 "internal")
_PREFIX_BOUNDARY_CHARS = frozenset("0123456789[<(\ \t")
_PREFIX_RULES: Tuple[Tuple[Tuple[str, ...], str], ...] = (
    ("int", "uint", "long", "short", "unsigned"), "integer",
```

```

    ("num", "float"), "number"),
    ("list",), "array"),
    ("dict",), "object"),
)

def _matches_type_prefix(base: str, prefixes: Tuple[str, ...]) -> bool:
    """检查 base 是否匹配前缀规则（精确匹配或边界后紧跟非标识符字符）。"""
    for p in prefixes:
        if base == p:
            return True
        if (len(base) > len(p) and base.startswith(p)
            and base[len(p)] in _PREFIX_BOUNDARY_CHARS):
            return True
    return False

def _normalize_single_type(raw: Any) -> Any:
    """将单个 type 值归一化为标准 JSON Schema 类型，非字符串原样返回。"""
    if not isinstance(raw, str):
        return raw
    if raw in _STANDARD_JSON_SCHEMA_TYPES:
        return raw
    base = raw.split("(", 1)[0].strip().lower() # 去除括号参数如 varchar(255)
    if base in _STANDARD_JSON_SCHEMA_TYPES:
        return base
    mapped = _JSON_SCHEMA_TYPE_ALIASES.get(base)
    if mapped is not None:
        return mapped
    for prefixes, target in _PREFIX_RULES:
        if _matches_type_prefix(base, prefixes):
            return target
    return raw # 未知类型保留原样

def normalize_json_schema_types(schema: Any) -> None:
    """递归遍历 schema，原地改写所有非标准 type 值为标准类型。"""
    if isinstance(schema, list):
        for item in schema:
            normalize_json_schema_types(item)
        return
    if not isinstance(schema, dict):
        return
    if "type" in schema:
        t = schema["type"]
        if isinstance(t, str):
            schema["type"] = _normalize_single_type(t)
        elif isinstance(t, list):
            schema["type"] = [_normalize_single_type(item) for item in t]
    # 递归进入嵌套关键字
    for key in ("properties", "patternProperties", "$defs", "definitions", "dependentSchemas"):
        nested = schema.get(key)

```

```
if isinstance(nested, dict):
    for v in nested.values():
        normalize_json_schema_types(v)
for key in ("anyOf", "oneOf", "allOf", "prefixItems"):
    nested = schema.get(key)
    if isinstance(nested, list):
        for v in nested:
            normalize_json_schema_types(v)
for key in ("items", "additionalProperties", "not", "if", "then", "else"):
    nested = schema.get(key)
    if isinstance(nested, (dict, list)):
        normalize_json_schema_types(nested)
```

评论区精华

无实质性 review 讨论。仅自动审查机器人 gemini-code-assist[bot] 评论无反馈。

- 自动审查无反馈 (other): 无需要处理

风险与影响

- 风险：回退后，非标准 JSON Schema 类型将再次导致 Draft202012Validator.check_schema 报错并返回 HTTP 400，原问题重现。但鉴于原改动导致 CI 失败，回退是目前恢复主分支稳定的合理选择。需注意后续应重新设计更稳定的归一化方案，并增加充分的测试覆盖。
- 影响：用户：之前依赖非标准类型的 tool schema 会再次被拒绝。系统：CI 应通过，主分支稳定性恢复。团队：需重新评估原问题的修复方案，考虑更稳健的实现和更全面的测试。
- 风险标记：回退引入原问题，缺少替代方案，需后续修复

关联脉络

- PR #23476 fix(tool_call): normalize non-standard JSON Schema types in tool params: 被此 PR 回退，原 PR 引入了归一化功能但导致 CI 不稳定。