后端大模型接入功能详解

文档说明

本文档说明语联灵犀后端系统中大模型接入相关的各个部分的功能。

文档版本：v1.0
最后更新：2024年
适用系统：语联灵犀后端系统

目录

一、概述

1.1 功能定位

大模型接入层是语联灵犀后端系统的核心组件，主要功能包括：

- **意图识别**：将用户的自然语言输入转换为结构化的工具调用指令
- **多模型支持**：支持多种大模型服务（OpenAI API、本地模型等）
- **智能降级**：当大模型不可用时，自动切换到基于规则的意图识别
- **工具调度**：根据大模型识别的意图，调度相应的工具执行任务

1.2 支持的接入方式

1. **OpenAI 兼容 API**
   - 阿里云百炼平台（DashScope）
   - OpenAI 官方 API
   - 其他兼容 OpenAI 格式的 API 服务
2. **本地模型**
   - 通过 HTTP API 调用的本地部署模型
   - 支持 Qwen、ChatGLM 等模型

3. **降级方案**
    - 基于规则的意图识别
    - 无需大模型即可运行

二、核心模块功能

2.1 LLMService（大模型服务抽象层）

文件位置： `app/core/llm_service.py`

功能概述

提供统一的大模型调用接口，支持多种大模型服务，并具备完善的降级机制。

主要功能

1. **统一调用接口**
    - 提供统一的 `chat()` 方法调用大模型
    - 自动识别并选择调用方式（OpenAI API 或本地模型）
    - 支持温度参数调节，控制输出随机性
2. **多模型支持**
    - **OpenAI 兼容 API**：支持调用 OpenAI 官方 API 和兼容格式的 API（如 DashScope）
    - **本地模型**：支持通过 HTTP API 调用本地部署的模型
    - 自动检测并选择合适的调用方式
3. **JSON 格式响应**
    - 自动检测提示词中的 JSON 格式要求
    - 自动启用 JSON 格式响应模式
    - 确保返回结果符合 JSON 规范
4. **降级机制**
    - 当未配置 API Key 时，自动使用规则识别
    - 当 API 调用失败时，自动降级到规则识别
    - 规则识别支持：新闻查询、天气查询、股票查询、计算等
5. **错误处理**
    - 完善的错误捕获和处理机制
    - 详细的错误日志记录
    - 认证错误时提供配置提示

降级方案功能

当大模型不可用时，系统自动使用基于规则的意图识别：

- **新闻查询**：识别关键词"新闻"、"资讯"，提取查询关键词和数量限制
- **天气查询**：识别关键词"天气"、"气温"，提取城市名称和查询天数
- **股票查询**：识别关键词"股票"，提取股票代码和查询天数
- **计算**：识别关键词"计算"或数学运算符，提取计算表达式

2.2 PromptTemplate（提示词模板管理）

文件位置： `app/core/prompt.py`

功能概述

管理所有与大模型交互的提示词模板，包括意图识别提示词、工具选择提示词等。

主要功能

1. **意图识别提示词**
    - 定义系统角色和职责
    - 列出所有可用工具及其功能说明
    - 说明每个工具的参数要求
    - 提供返回格式示例（单工具、多工具、计算等）
2. **工具说明**
    - **weather**：天气查询工具，支持查询指定城市的天气信息（7天预报）
    - **news**：新闻检索工具，支持根据关键词检索新闻
    - **stock**：股票数据查询工具，支持查询股票历史价格数据
    - **calculate**：数值计算工具，支持执行数学计算
    - **document**：文档生成工具，支持生成报告、邮件、总结等文档
3. **多工具支持**
    - 支持识别多个任务并返回工具链
    - 说明工具链的执行顺序
    - 支持工具间的数据传递（如前一个工具的结果作为后一个工具的输入）
4. **工具选择提示词**
    - 支持多轮工具调用场景
    - 根据之前的工具执行结果，选择下一个要调用的工具

2.3 Agent（智能 Agent）

文件位置： `app/core/agent.py`

功能概述

系统的核心调度器，负责协调大模型和工具调度，实现从用户自然语言输入到工具执行的完整流程。

主要功能

1. **意图识别**
    - 接收用户的自然语言输入
    - 调用大模型进行意图识别
    - 解析大模型返回的 JSON 结果
    - 提取工具名称和参数
2. **单工具调用**
    - 识别单个工具需求
    - 提取并验证工具参数
    - 调用相应工具执行任务
    - 返回工具执行结果
3. **多工具链式调用**
    - 识别包含多个任务的用户需求
    - 按顺序执行多个工具
    - 支持工具间的数据传递（如将天气数据传递给文档生成工具）
    - 记录工具链执行信息
4. **参数处理**
    - **天气工具**：验证和补全城市名称、查询天数
    - **新闻工具**：验证和补全查询关键词、数量限制
    - **股票工具**：支持股票名称到代码的映射（如"贵州茅台" → "600519"）
    - **计算工具**：提取和验证计算表达式
    - **文档工具**：处理模板类型、内容提示、上下文数据
5. **多任务检测**
    - 检测用户输入是否包含多个任务
    - 识别不同任务类型的关键词
    - 优先处理计算任务
6. **降级处理**
    - 当大模型调用失败时，自动使用规则识别
    - 当 JSON 解析失败时，自动使用规则识别
    - 确保系统始终可用

2.4 Config（配置管理）

文件位置： `app/config.py`

功能概述

管理应用配置，支持从环境变量和 `.env` 文件读取配置。

大模型相关配置

1. **LLM_API_KEY**
   - 大模型 API Key
   - 用于认证大模型 API 请求
2. **LLM_BASE_URL**
   - API 基础 URL
   - 如果未配置，使用默认的 DashScope 端点
   - 如果配置为本地地址，会调用本地模型
3. **LLM_MODEL**
   - 模型名称
   - 指定要使用的大模型（如 `qwen-turbo`、`gpt-3.5-turbo`）

三、工作流程

3.1 单工具调用流程

```
用户输入自然语言
    ↓
Agent 接收输入
    ↓
构建意图识别提示词（包含工具列表、用户需求、格式要求）
    ↓
调用大模型 API
    ↓
大模型返回 JSON 格式的意图识别结果
    ↓
解析 JSON，提取工具名称和参数
    ↓
处理参数（验证、补全、转换）
    ↓
调用相应工具
    ↓
返回工具执行结果
```

3.2 多工具链式调用流程

```
用户输入包含多个任务的自然语言
    ↓
Agent 检测到多任务
    ↓
调用大模型识别所有需要的工具
    ↓
大模型返回工具链（按顺序排列）
    ↓
按顺序执行每个工具
    ↓
如果当前工具是文档工具，将前一个工具的结果作为上下文数据
    ↓
返回所有工具的执行结果
```

3.3 降级流程

```
用户输入自然语言
    ↓
尝试调用大模型
    ↓
如果未配置 API Key 或调用失败
    ↓
使用规则识别（基于关键词匹配）
    ↓
提取工具名称和参数
    ↓
调用相应工具
    ↓
返回工具执行结果
```

四、配置说明

4.1 环境变量配置

在 `backend/.env` 文件中配置：

```
# 大模型配置
LLM_API_KEY=your-api-key-here
LLM_BASE_URL=https://dashscope.aliyuncs.com/compatible-mode/v1
LLM_MODEL=qwen-turbo
```

4.2 配置示例

使用阿里云百炼平台（DashScope）

```
LLM_API_KEY=sk-xxxxxxxxxxxx
LLM_BASE_URL=https://dashscope.aliyuncs.com/compatible-mode/v1
LLM_MODEL=qwen-turbo
```

使用 OpenAI 官方 API

```
LLM_API_KEY=sk-xxxxxxxxxxxx
LLM_BASE_URL=https://api.openai.com/v1
LLM_MODEL=gpt-3.5-turbo
```

使用本地模型

```
LLM_API_KEY=not-needed
LLM_BASE_URL=http://localhost:8001/v1
LLM_MODEL=Qwen/Qwen-7B-Chat
```

不使用大模型（降级方案）

```
# 不设置 LLM_API_KEY 或设置为空
LLM_API_KEY=
```

4.3 配置说明

- **LLM_API_KEY**：大模型 API Key，如果未配置，系统会自动使用降级方案
- **LLM_BASE_URL**：API 基础 URL，如果未配置，默认使用 DashScope 端点
- **LLM_MODEL**：模型名称，根据实际使用的大模型设置

附录

A. 文件清单

- `app/core/llm_service.py` ：大模型服务抽象层
- `app/core/prompt.py` ：提示词模板管理
- `app/core/agent.py` ：智能 Agent
- `app/config.py` ：配置管理

B. 相关文档

- `docs/LLM_INTEGRATION.md` ：大模型接入指南
- `backend/ENV_SETUP.md` ：环境配置说明

- `backend/QUICK_START.md` ：快速开始指南

文档结束