

AgentFromPku: 多角色多模态校园交互系统

WindGram¹ Elty-Nov¹

Abstract

AgentFromPku 是一个基于多 Agent 架构的智能交互系统，旨在通过动态切换不同风格的 AI 角色（如“派蒙”“塞万提斯之魂”），结合文本、图片、音频的多模态输出，为用户提供沉浸式的校园文化交互体验。系统采用模块化设计，实现了 API 通信、流式响应处理、GUI 界面构建及场景化交互逻辑，通过 Dify 平台构建智能体框架，并针对不同角色特性进行参数调优，最终实现了“赛博游北大”的创新应用场景。

1. 项目主题与背景

AgentFromPku 项目聚焦于人工智能与校园文化的融合，旨在构建一个支持多角色动态切换、多模态输出的智能交互系统。项目以北京大学为场景背景，通过不同特性的 AI 角色（如知识型的“朱光潜_agent”、活泼型的“派蒙_agent”），结合文本对话、图片生成、音频合成等功能，为用户提供沉浸式的校园文化体验。系统的核心目标是实现自然、生动的人机交互，同时展现 AI 技术在文化传播与教育场景中的应用潜力。

1.1. 需求分析

1.1.1. 核心需求

- 构建多 Agent 交互系统，支持动态切换不同风格的 AI 角色（如“派蒙”“塞万提斯之魂”）
- 实现多模态输出（文本/图片/音频）的集成处理
- 提供用户友好的 GUI 界面，支持场景化交互体验

1.1.2. 功能需求

- 参数动态调节：针对不同 Agent 特性调整 LLM 参数（Temperature/top_p）
- 智能路由：根据用户输入自动触发 Agent 切换机制
- 文件处理：自动下载并管理 Dify 平台生成的多媒体文件
- 错误处理：API 密钥缺失时的降级处理方案

1.1.3. 非功能需求

- 响应速度：流式响应保证交互实时性（通过 stream_handler 实现）
- 可维护性：模块化代码结构（GUI/API/场景管理分离）
- 扩展性：支持后续新增 Agent 类型和输出格式

1.2. 技术选型

系统采用以下技术栈实现多模态交互与 Agent 管理：

- 文本处理：调用 DeepSeek 模型，处理自然语言对话与生成
- 图片处理：集成硅基流动 API，实现图片生成与处理
- 音频处理：使用 fishaudio 接口，完成语音合成与播放控制
- 交互平台：基于 Dify 搭建智能体框架，实现 Agent 逻辑与流水线处理
- GUI 开发：采用 tkinter 构建用户界面，实现场景化交互体验
- 流式处理：自定义 stream_handler 实现实时响应更新

2. 系统设计与实现

2.1. 整体框架

系统采用四层架构设计，确保模块化与可扩展性：

- 核心功能层：包含 API 通信与流式处理模块，负责与外部服务交互
- UI 层：管理界面构建、聊天气泡与信息面板的显示逻辑
- 业务逻辑层：处理场景切换与 Agent 整合，实现智能路由
- 入口层：程序主入口，负责初始化与资源加载

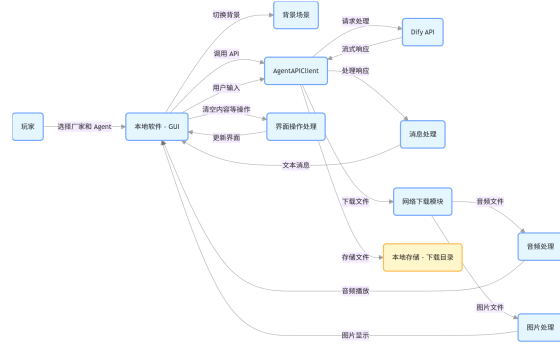


Figure 1. 项目架构图

2.2. 代码结构

系统代码采用如下层次结构：

核心功能层

`api_client.py` (API通信)

`stream_handler.py` (流式处理)

UI层

`ui_builder.py` (主界面)

`chat_bubble.py` (聊天气泡)

`info_panel.py` (信息面板)

业务逻辑层

`scene_switcher.py` (场景切换)

`agent_gui.py` (功能整合)

入口层

`main.py` (程序入口)

2.2.1. 核心功能层详解

api_client.py: 实现 Dify API 客户端，负责文件上传/下载、音频播放控制、图片处理、流式响应处理及会话管理。

stream_handler.py: 管理流式交互，处理请求队列、实时更新 UI、处理多媒体消息及会话管理。

2.2.2. UI 层详解

ui_builder.py: 构建 GUI 界面，管理主窗口布局、背景图片、工具栏组件及响应式设计。

chat_bubble.py: 管理聊天消息显示，处理用户/AI 消息气泡的创建、滚动逻辑与样式控制。

info_panel.py: 显示角色信息，包含人物照片、姓名介绍及固定布局管理。

2.2.3. 业务逻辑层详解

agent_gui.py: 整合 UI 与 API 功能，初始化界面、设置默认角色、连接 API 客户端及提供测试用例。

scene_switcher.py: 根据对话内容切换背景与角色，管理多场景状态、API 密钥更新及会话重置。

2.3. API 调用与 Dify 平台集成

2.3.1. 调用方式

- 硬编码 base url，使用.json 文件存储 api_key 并通过.gitignore 保护
- 鲁棒性设计：若未获取到 key，提示用户手动输入
- 输入内容均为文本，输出支持文本、图片、音频的多模态处理

2.3.2. DIFY 平台处理逻辑

- 输出文件时返回文件名与网址，程序自动下载至/downloads 文件夹
- 音频文件自动重命名为.mp3，提供”播放”按钮支持播放控制
- 本地部署方案：通过 docker 部署，导出 DSL 后上传至官网调用

2.3.3. AGENT 框架设计

核心框架：系统实现以下核心 Agent：

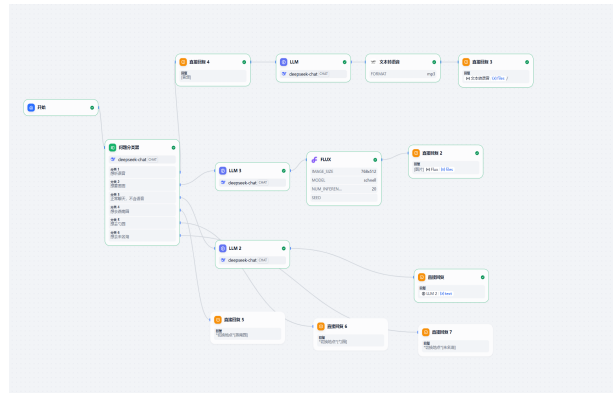


Figure 2. Dify 框架图

- 派蒙 __agent: 活泼型角色，用于轻松对话与校园引导
- 朱光潜 __agent: 知识型角色，专注美学与人文知识解答
- 塞万提斯 __agent: 文学型角色，擅长诗歌创作与文学解读
- Nyw__agent: 综合型角色，支持多场景交互

2.3.4. AGENT 切换机制

用户输入中包含场景关键词时，系统自动触发切换逻辑：

1. 云端 DifyAgent 判断用户意图，调用场景切换函数
2. 本地程序更新背景图片、API 密钥及角色信息
3. 重置会话状态，实现无缝角色过渡

3. 参数调优与评估

3.1. LLM 参数调整

针对不同 Agent 特性,主要调整 Temperature 与 top_p 参数：

3.1.1. TEMPERATURE 参数

控制输出分布平滑度，影响生成多样性：

Table 1. Temperature 参数取值与应用场景

取值区间	生成效果	典型场景
0 0.3	高度确定性输出	事实性问答/代码生成
0.3 0.7	平衡多样性与连贯性	对话系统/一般文本生成
0.7 1.0	显著增加创造性	诗歌/故事创作
>1.0	高风险随机输出	通常避免使用

应用案例：

- 派蒙 _agent: temp=0.5，平衡连贯性与活泼风格
- 塞万提斯 _agent: temp=1.0，增强诗歌创作的发散性
- 朱光潜 _agent: temp=0.3，确保知识解答的准确性

3.1.2. Top_P 参数（核采样）

动态概率累积筛选机制，影响生成质量与多样性：

Table 2. Top_P 参数取值与生成特点

取值区间	筛选范围	生成特点
0.9 1.0	保留绝大多数可能 TOKEN	接近原始分布
0.7 0.9	截断低概率长尾	平衡质量与多样性
<0.7	强筛选可能导致语义断裂	需要配合温度调整

应用案例：

- 派蒙 _agent: top_p=0.9，确保对话流畅性
- 塞万提斯 _agent: top_p=0.8，在创造性与连贯性间平衡
- 朱光潜 _agent: top_p=0.95，保证知识输出的完整性

3.2. 参数交互效应

Temperature 与 top_p 联合使用时存在显著交互效应：

- temp=0.3+top_p=0.95：生成内容精确但略有变化，适合技术文档
- temp=0.7+top_p=0.8：平衡创意与逻辑，适合对话系统
- temp=1.0+top_p=0.7：高创造性但需注意语义连贯性，适合诗歌创作

4. 用户反馈与系统优化

4.1. 优势亮点

- 角色与内容特色：派蒙等角色紧跟潮流，贴合年轻人喜好；AI 回答融入生活气息，介绍北大建筑时补充人物细节、推荐活动，生动展现人文气息。
- 知识与交互体验：能解答校园历史、景点典故、专业知识（如朱光潜美学、塞万提斯文学），虚拟人物语言风格各异，回复逻辑清晰，带幽默调侃，增强真实感。
- 多模态与场景：支持画画、语音输出，搭建北大景点及虚拟人物场景，部分界面配精美图片，实现“赛博游北大”沉浸体验。

4.2. 待优化问题

- 性能效率：AI 思考及响应时间长，存在”问简短问题等待久””高并发时响应慢”情况。
- 功能体验：交互界面布局待优化；场景切换生硬，缺乏反馈；部分功能缺失（如”不会唱歌””校歌识别错误”），图片、角色拓展需求强烈。
- 内容与引导：彩蛋等隐藏内容难挖掘；初始使用缺乏引导；语音输出存在瑕疵（如读括号中动作），部分回答情感、语调生硬。

4.3. 迭代方向

- 技术优化：优化 AI 响应逻辑、服务器并发处理，提升交互速度
- 体验打磨：调整界面布局，优化场景切换反馈，补充功能（如音乐、更多角色/图片），完善语音输出自然度

- 内容引导：设计显性引导流程，强化隐藏内容提示，提升用户上手体验

5. 项目分工

5.1. WindGram

- 项目架构设计
- 代码框架搭建
- Dify 通用框架搭建与朱光潜智能体的制作
- 技术选型（Dify 平台、fishaudio 等）
- api 调用、多模态输出、流式响应、对话框、场景切换基本代码框架实现
- Github 仓库管理与代码版本控制
- 项目文档排版（LaTeX）与演示视频录制

5.2. Elty-Nov

- UI 界面细节调整与用户体验优化
- Github 仓库协同管理与代码整合
- fishaudio 声音合成接口集成与音频处理
- 派蒙、塞万提斯、nyw_Agent 的搭建、修改与参数调优
- API 密钥管理与安全机制实现
- 聊天气泡、API 管理模块的具体实现与图片资源整合
- 项目文档撰写与用户反馈整理

6. 项目总结与展望

AgentFromPku 项目成功实现了 AI 技术与校园文化的创新结合，通过多角色交互、多模态输出及场景化设计，展现了人工智能在文化传播与教育场景中的应用潜力。系统获用户”科技与人文奇妙碰撞，沉浸式数字化校园之旅”的积极评价，同时在性能效率、功能体验等方面明确了优化方向。

未来工作将聚焦于：

- 扩展更多 AI 角色与校园场景，丰富交互内容
- 优化多模态处理流程，提升响应速度与资源利用率
- 引入用户画像技术，实现个性化交互体验
- 探索与校园服务系统的集成，增强实用价值

A. 附录

A.1. Git 仓库信息

项目代码已开源至 GitHub，仓库地址：

```
git@github.com:WindGram/AgentFromPku.git
```

仓库包含完整的代码实现、配置文件、用户反馈数据及项目文档。

A.2. 派蒙调参结果

详细的派蒙 Agent 参数调整实验结果见《派蒙调参结果.xlsx》，包含不同 Temperature 与 top_p 组合下的生成效果对比，最终选定 temp=0.5、top_p=0.9 作为最优参数组合，兼顾对话的流畅性、创造性与角色设定的契合度。

A.3. 用户反馈汇总

项目收集了十多位用户的真实使用反馈，具体内容见《AgentFromPku 用户反馈.pdf》，反馈涵盖功能体验、性能表现、内容丰富度等多个维度，为系统迭代提供了重要依据。

A.4. 图片

项目架构和 Dify 框架清晰图片