GameSage: Master Every Game with the Power of an LLM-based Agent

Zhen Qin
zhenq3@illinois.edu
University of Illinois Urbana-Champaign
Champaign, Illinois, USA

Yixiao Wang
yixiao8@illinois.edu
University of Illinois Urbana-Champaign
Champaign, Illinois, USA

1 Introduction

We propose GameSage, an intelligent, plugin-based information integration agent designed specifically for gamers across all platforms, including PC, console, and mobile gaming communities. GameSage leverages advanced Large Language Model (LLM) technology combined with the ReAct (Reasoning and Acting) framework to perform multi-turn reasoning, enabling dynamic orchestration of web-based tools and platform-specific plugins. The system will be implemented as a standalone web application featuring modular plugins tailored for popular platforms such as Bilibili, Zhihu, NGA, and others.

The key functions of GameSage include natural language game query processing, enabling users to intuitively ask game-related questions and receive accurate, contextually appropriate responses. The system significantly enhances information retrieval and extraction capabilities by performing intelligent cross-platform searches and efficiently extracting high-quality gaming content, such as guides, community discussions, videos, and other media, through specialized plugins. Furthermore, GameSage effectively synthesizes and summarizes extracted information from multiple diverse sources into coherent, precise, and helpful answers, ensuring gamers can access comprehensive, relevant gaming knowledge effortlessly.

2 Motivation

The motivation behind GameSage is driven by the limitations of current Large Language Models, which frequently struggle to provide effective, helpful, real-time, and detailed gaming guides, often producing inaccurate or misleading content due to hallucination issues. Additionally, mainstream gaming portals often offer incomplete or outdated information. As experienced gamers ourselves, we have observed that valuable and practical gaming insights typically emerge from game forums, gaming videos, and user-generated content in comment sections. GameSage aims to bridge this gap by aggregating these fragmented yet highly valuable sources, delivering comprehensive and reliable gaming knowledge directly to users.

3 Approach

Our approach utilizes a hybrid architecture combining the ReAct framework and a plugin-based system, integrating LLM-driven reasoning for dynamic task decomposition, information retrieval, and intelligent tool selection. The core tools within our architecture include a router, platform-specific plugins, and a summarizer. The router effectively directs queries by matching user requests to the most suitable platform plugins, optimizing information retrieval tailored to each platform's content structure and strengths.

Platform-specific plugins represent the heart of our information extraction process, precisely engineered to gather and process content from platforms. For instance, the Bilibili plugin leverages LLM-generated optimized search queries to identify relevant gaming videos, retrieving both the video's ASR-transcribed content and a curated set of user comments. To enhance the accuracy and granularity of comment analysis, we will employ statistical machine learning techniques, aiming for robust extraction of meaningful insights.

The summarizer, driven predominantly by an LLM, integrates and synthesizes the diverse data streams collected by the plugins, delivering concise, comprehensive, and accurate responses directly to the users. Technologically, our implementation will involve Python as the primary programming language, frameworks such as LangChain or CrewAI for agent orchestration, aiohttp and asyncio for web scraping, and various APIs and libraries for seamless integration and efficient development.

4 Evaluation

For evaluation, the primary method will involve human assessments based on a Likert scale, measuring the quality of the agent's final output across four critical dimensions: information accuracy, effectiveness, detail, and stability. Additionally, inference speed will serve as another important evaluation metric, ensuring the system not only provides high-quality responses but also operates efficiently and responsively.

5 Timeline

Shown in Table 1.

Table 1: Project development timeline

Week	Milestone
0	Agent MVP (DONE)
1	Complete Bilibili plugin & ASR transcription
2	Add LangChain/CrewAI for multi-agent flow
3	Improve ReAct structure (MCP)
4	Add more plugins for different platforms
5	Final polish + Evaluation + Demo preparation

6 Task Division

- Zhen Qin: plugins development
- Yixiao Wang: LLM + ReAct structure