Multi-Agent System (MAS) for USOS Courses Recommendations

Łojek Grzegorz, Wilczyński Jakub
Supervisor: Szczuka Marcin

Objectives

The goal of this project is to develop an intelligent recommendation system to help students with choosing general university subjects in the USOS system. These subjects must fit individual criteria such as area of interest, time availability, and curriculum relevance.

We aim to:

- Implement a MAS (Multi-Agent System) that utilizes agents to filter, rank, and recommend courses.
- Compare and evaluate some of MAS architectural styles and different LLMs to determine which combination is most effective for this task.

Introduction

Choosing general university subjects is a tedious task for students. The number of available options is large, course descriptions are often vague, and manual searching requires matching preferences with available timeslots and personal preferences.

Multi-Agent Systems (MAS) offer a promising solution to this issue. By dividing the recommendation task into specialized subtasks handled by cooperating agents (e.g., filtering, ranking, interpreting user needs), MAS architectures can provide intelligent and personalized recommendations in an efficient and scalable way (Mahmood et al., 2013).

In this project, we propose using MAS combined with NLP techniques and LLMs to interpret student preferences and automatically find suitable USOS courses.

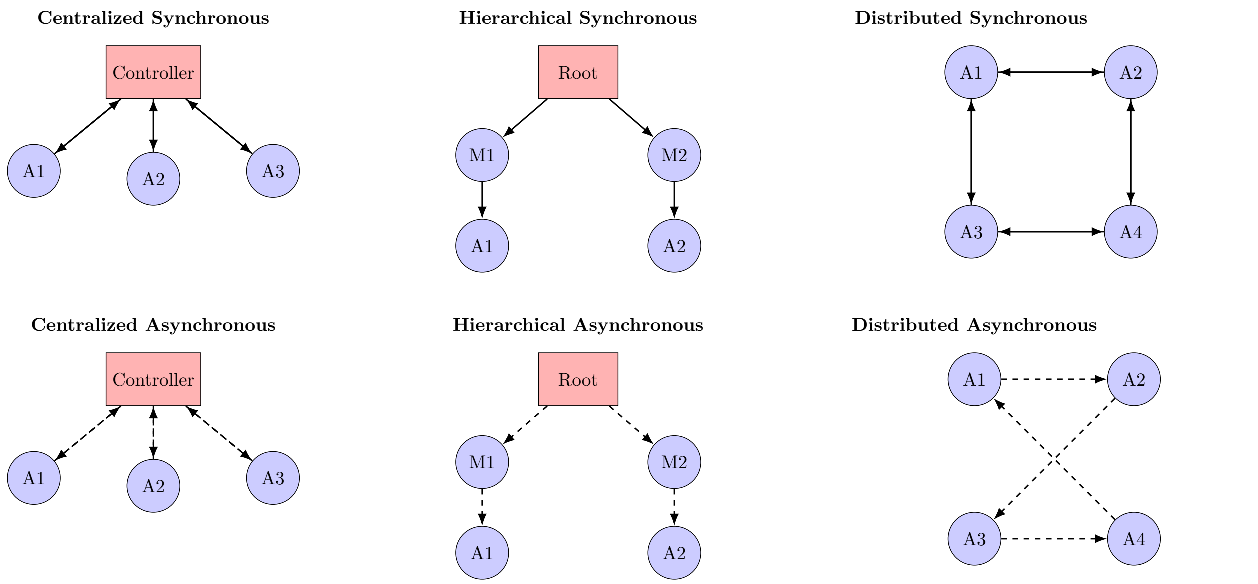
MAS Architectures

In traditional single-agent systems, one central agent is responsible for all processing, limiting scalability and fault tolerance. In contrast, Multi-Agent Systems (MAS) distribute tasks among multiple autonomous agents, improving efficiency and allowing specialization.

MAS can be categorized by architecture (Davidsson & Svahnberg, 2005):

- **Centralized Synchronous Architectures:** All agents communicate through a central controller, operating in synchronized steps. Easy to manage but prone to bottlenecks and failures at the central node.
- **Centralized Asynchronous Architectures:** Similar to the above, but agents operate independently, communicating with the controller when ready. More flexible but still centralized.
- **Hierarchical Synchronous Architectures:** Agents are structured in a hierarchy, with synchronous operations at each level. Suitable for layered decision-making but limited in flexibility.

- **Hierarchical Asynchronous Architectures:** Agents in hierarchy communicate independently, improving responsiveness and scalability.
- **Distributed Synchronous Architectures:** No central controller. All agents operate in sync, communicating peer-to-peer. Offers fault tolerance but requires strict coordination.
- **Distributed Asynchronous Architectures:** Agents act autonomously and communicate as needed. Highly scalable and robust, but harder to control or debug.



Each architecture has trade-offs depending on the complexity of tasks, communication overhead, and fault tolerance. We plan to implement and test different architectures to determine which performs best for our recommendation scenario.

Evaluation

To evaluate the effectiveness of our MAS-based recommendation system, we employ several standard methods:

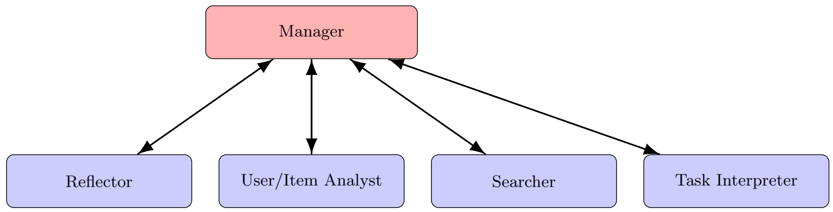
- **Precision**: Measures how many of the recommended courses are relevant. This metric helps assess the system's ability to present relevant items among the top results (Herlocker et al., 2004).
- **Recall**: Assesses how many of the relevant courses are included in the recommendations. This is particularly useful for understanding the system's comprehensiveness (Herlocker et al., 2004).
- **NDCG (Normalized Discounted Cumulative Gain)**: Evaluates the quality of ranking by assigning higher importance to relevant items ranked earlier in the recommendation list (Järvelin & Kekäläinen, 2002).
- **Response Time**: Captures the latency between a user request and the system's response. This metric reflects the responsiveness and real-time performance of the MAS.
- **User Satisfaction**: Based on user feedback regarding how well the system meets their expectations, measured through surveys or interviews (We will review some responses to chceck if they are accurate).
- **Architectural Efficiency**: Includes system-level metrics such as the number of inter-agent messages, message complexity, and agent idle time. These are used to assess internal system performance and to compare synchronous vs. asynchronous agent architectures.

Related Work

Some existing frameworks demonstrating variations on the mentioned before architectures:

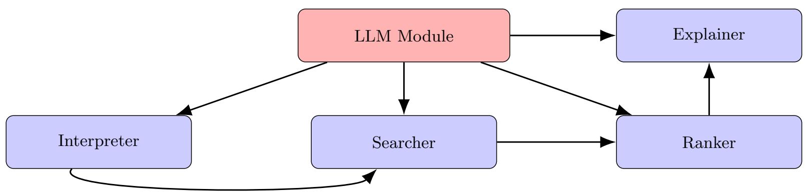
- **MACRec**– Implements a modular MAS for various recommendation tasks. It uses agents like Manager, Analyst, Reflector, etc., each handling a distinct part of the process (Wang et al., 2024).

MACRec Framework Modular MAS with specialized agents coordinated by Manager



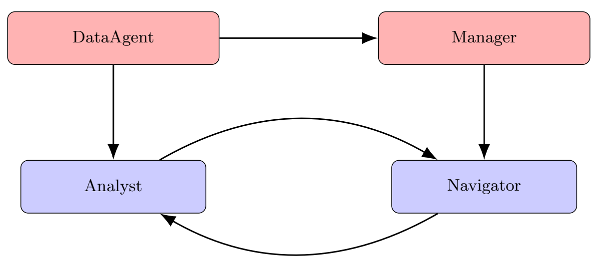
- **MATCHA** – Built by Roblox, MATCHA integrates LLMs into MAS. It supports free-text recommendation queries and tasks like ranking and explanation generation (Hui et al., 2025).

MATCHA FrameworkLLM-powered MAS with multi-agent cooperation for conversational recommendations



- **MAS4POI** – Designed for point-of-interest recommendations, using specialized agents such as Navigator and Analyst (Wu et al., 2024).

MAS4POI Framework Specialized MAS for location recommendations with data and navigation agents



- **Swarms** - Framework with already implemented MAS architectures. It might be an alternative to crewai where we would have to implement everything from scratch. (http://swarms.ai/)

Figure 1: Swarms logo

References

Davidsson, P., & Svahnberg, M. (2005). Characterization and evaluation of multi-agent system architectural styles. *Software Engineering for Multi-Agent Systems*, *3914*. https://doi.org/10.1007/11738817_11

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5–53. https://doi.org/10.1145/963770.963772

Hui, Z., Wei, X., Jiang, Y., Gao, K., Wang, C., Ong, F., Yoon, S.-e., Pareek, R., & Gong, M. (2025). Matcha: Can multi-agent collaboration build a trustworthy conversational recommender? https://arxiv.org/abs/2504.20094

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, *20*(4), 422–446. https://doi.org/10.1145/582415.582418

Mahmood, M. A., El-Bendary, N., Platoš, J., Hassanien, A. E., & Hefny, H. A. (2013). An intelligent multi-agent recommender system. *Innovations in Bio-Inspired Computing and Applications*, *237*, 201–213. https://doi.org/10.1007/978-3-319-01781-5_19

Wang, Z., Yu, Y., Zheng, W., Ma, W., & Zhang, M. (2024). Macrec: A multi-agent collaboration framework for recommendation. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2760–2764. https://doi.org/10.1145/3626772.3657669

Wu, Y., Peng, Y., Yu, J., & Lee, R. S. T. (2024). Mas4poi: A multi-agents collaboration system for next poi recommendation. https://arxiv.org/abs/2409.13700

Useful links

MACRec github:
https://github.com/wzf2000/MACRec

MAS4POI github:
https://github.com/yuqian2003/MAS4POI

Swarms github:
https://github.com/kyegomez/swarms