LLM for RecSys

Ainura Zakirova [1], Irina Maltseva [1], Andrei Semenov [2], Robert Zaraev [2], Dmitri Kiselev [3]

[1]Innopolis university,[2] ITMO university, [3] AIRI

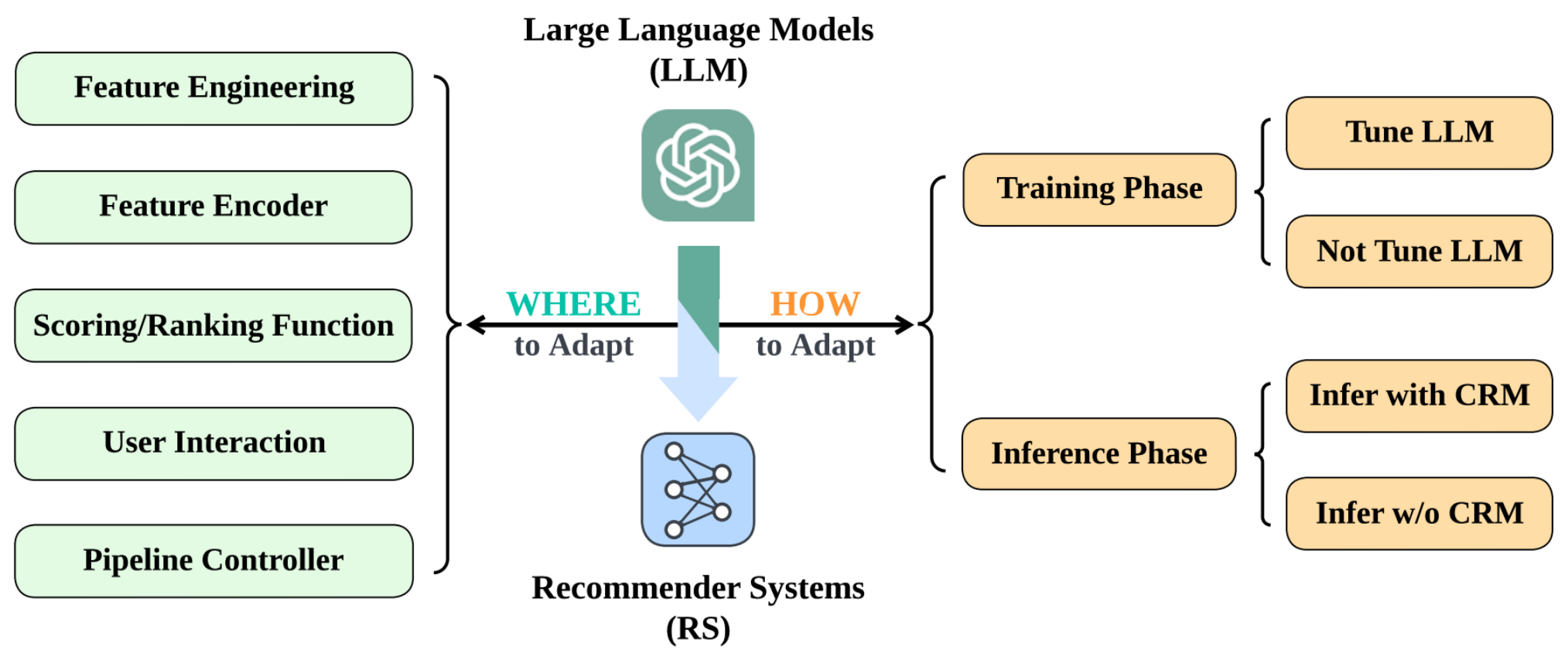
Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across various tasks, including recommendation systems. However, comparing the performance of LLM-based models with traditional benchmarks has been challenging due to the absence of a unified evaluation framework. To address this, we developed LLM4Rec, a comprehensive framework that integrates LLMs into multiple stages of the recommendation process. This framework facilitates fair performance comparisons between LLM-based recommendation systems and traditional models.

Literature Review

Recent research on applying LLMs to recommendation systems identifies several key applications [2]:

- LLMs generate detailed user profiles by summarizing experiences and adding personalized traits to item descriptions.
- They serve as feature encoders, transforming open-world and semantic knowledge into dense vectors to enhance user and item representations.
- LLMs are utilized for scoring and ranking tasks, leveraging their text understanding and reasoning capabilities.
- Their conversational abilities can be used to make recommendations interactive and personalized, enhancing transparency in user interactions.
- LLMs can function as autonomous agents, controlling the recommendation system pipeline, adapting strategies based on feedback, and simulating specific roles to further personalize the user experience.



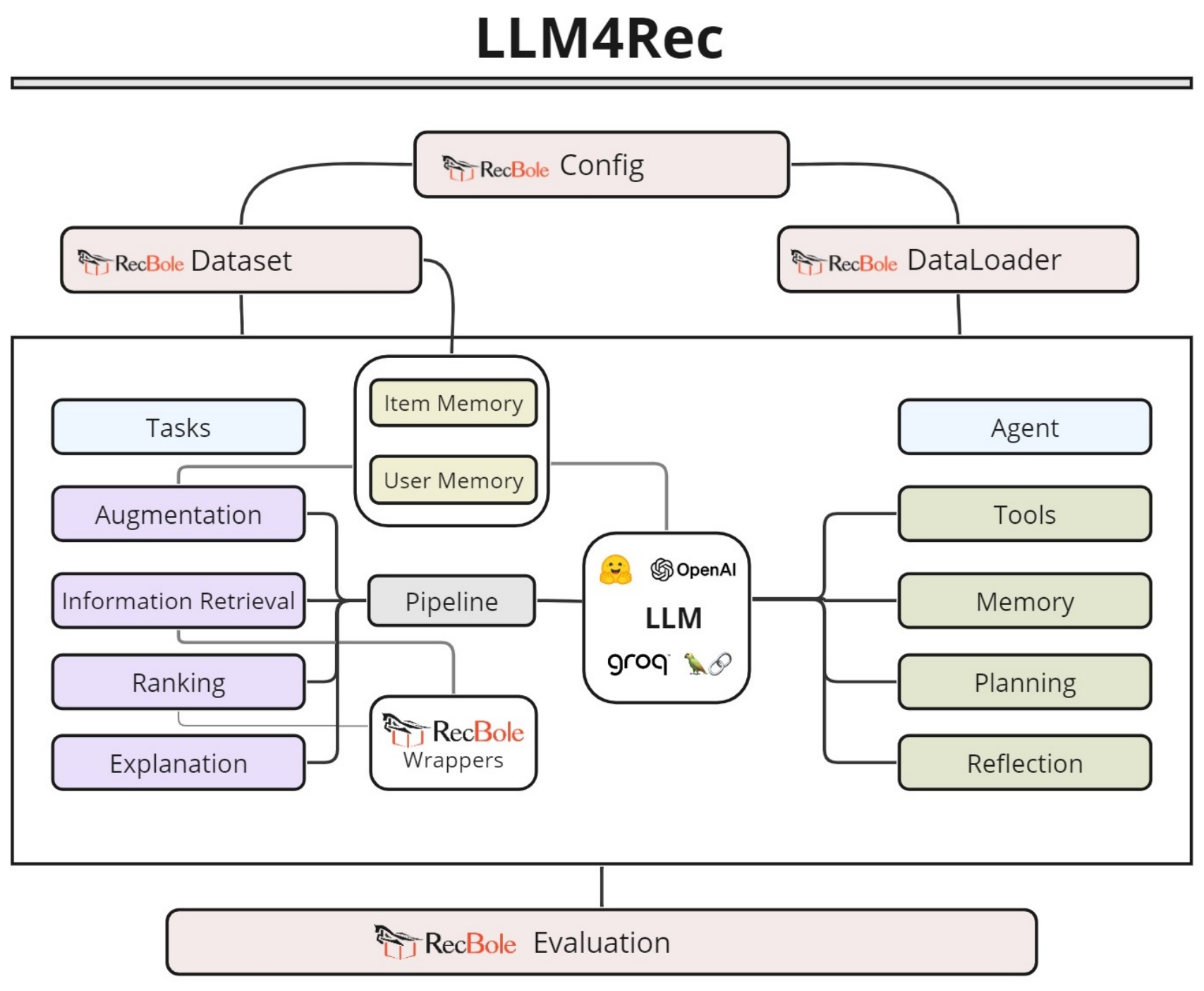
How to use LLM in RecSys

Methodology

LLM4Rec is designed to help researchers develop and evaluate recommendation models that leverage LLMs. Based on our review of current literature on LLM applications in recommendation systems, we have identified and implemented key use cases of LLM within the recommendation pipeline. To facilitate a standardized environment for performance evaluation and comparison with traditional models, we have integrated LLM4Rec with the RecBole [3] framework, which supports a wide range of datasets and models.

Framework structure

The framework is structured into two key components. The first is dedicated to constructing the experimental setup with methods adopted from RecBole. The second component focuses on the integration of LLMs into the recommendation system pipeline. There are two applications supported: running LLM-based recommendation tasks in a sequential pipeline and implementing LLM agents within the recommendation system.

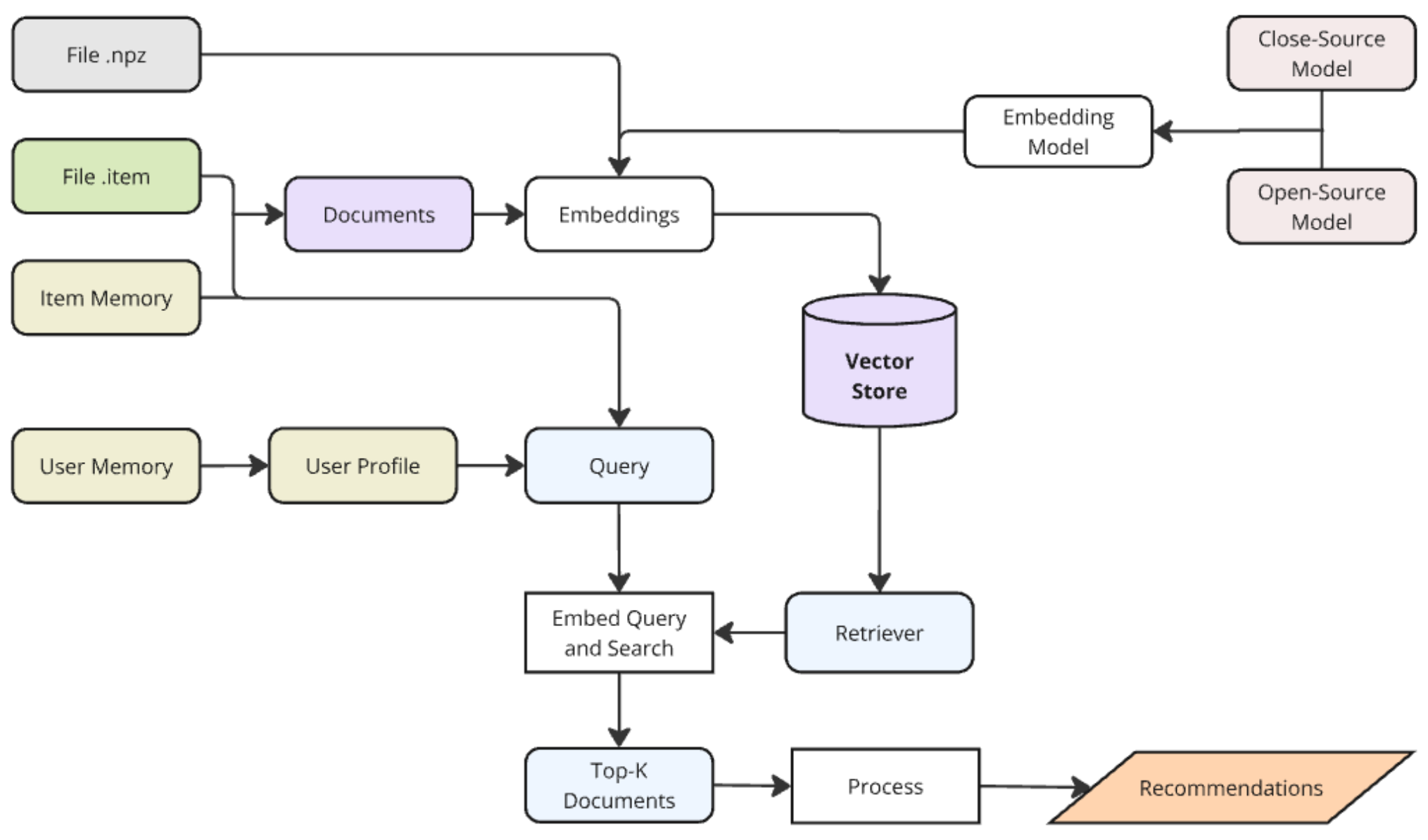


Framework structure

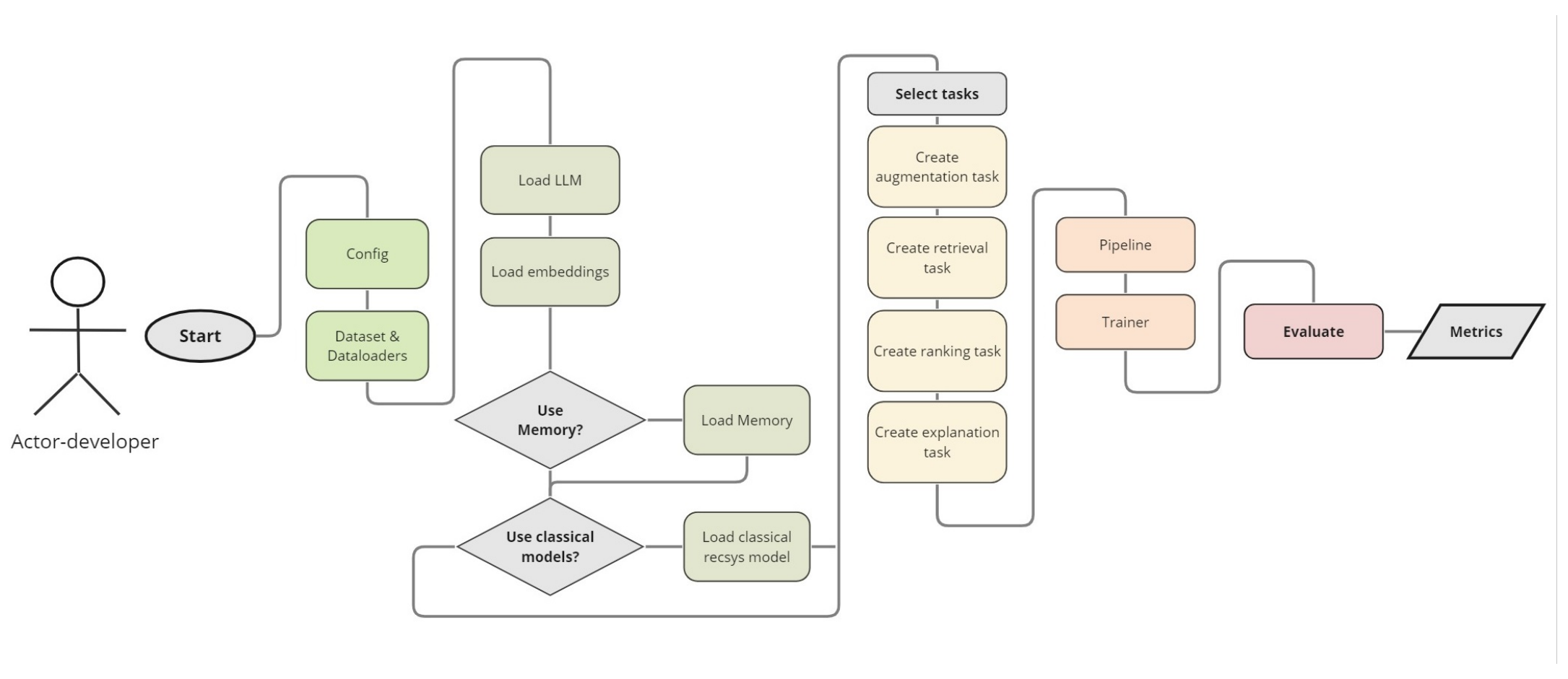
Sequential pipeline of recommendation tasks

The proposed framework incorporates a sequential use of Large Language Models across various stages of the recommendation system, forming an integrated pipeline. The components implemented within this pipeline include:

- **Information retrieval**: uses a FAISS retrieval model to efficiently find relevant items.
- **Ranking**: employs an LLM-based ranking system (LLMRank [1]) to reorder the retrieved items.
- **Augmentation**: builds user and item memories, populated from historical interactions and external data sources, to enhance the recommendation context.
- **Explanation**: uses LLMs to generate explanations for each recommendation provided to the user.



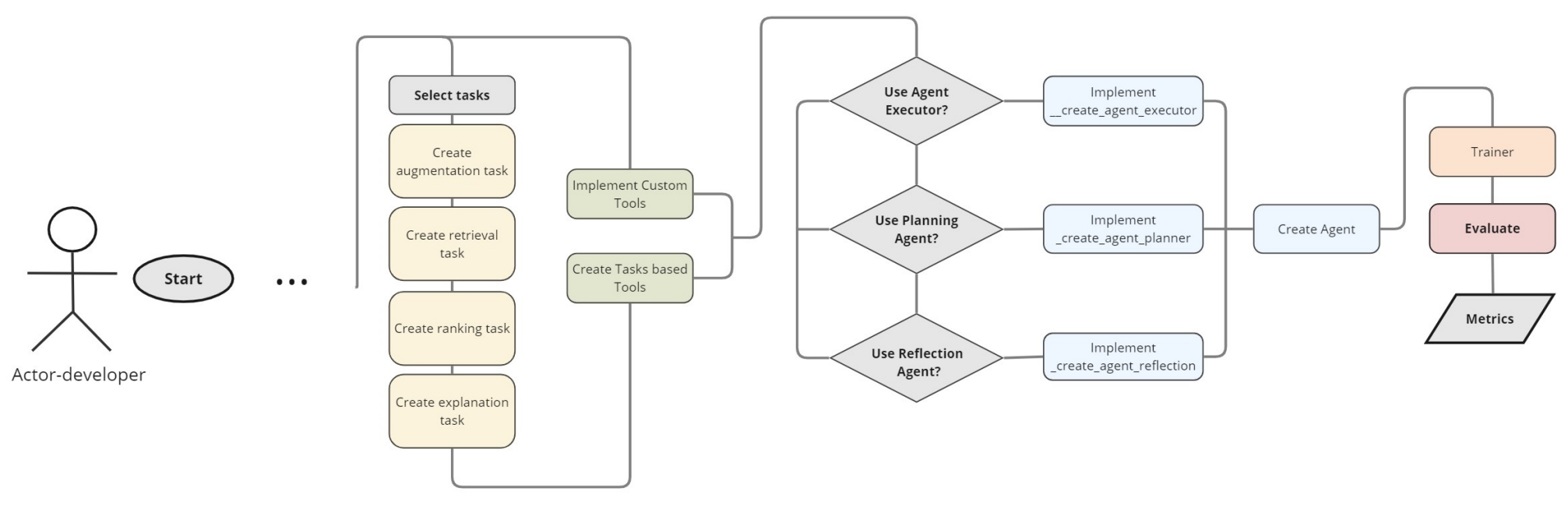
Information retrieval task example



Example of sequential pipeline of tasks

Agents

The framework is designed to support LLM based agents, that can be integrated into conversational systems and manage the recommendation system pipeline. It includes key related components such as various tools, memory, and mechanisms for reflection and planning to facilitate the use of these agents.



Example of agent pipeline

Experiments

We evaluated our framework through a series of experiments using the MovieLens-100K dataset. Our analysis included comparisons between both open-source and closed-source models, as well as with traditional recommendation system models. For Information Retrieval component we compared various configurations of item features, including overviews extracted from Wikipedia.

Table 1: Comparison of Information Retrieval

Model	Item Features	Recall@20	Recall@50	Recall@100
all-MiniLM-L6-v2	title genres year	0.0456	0.0912	0.1495
all-MiniLM-L6-v2	title genres year + augmentation	0.0180	0.0467	0.0880
text-embedding-ada-002	title genres year	0.0467	**0.0986**	**0.1569**
text-embedding-ada-002	title genres year + augmentation	0.0350	0.0679	0.1050
ALS	-	**0.0615**	0.0891	0.1135

Table 2: Comparison of sequental pipeline

Retrieval Model	Ranker model	Recall@10	Recall@20	NDCG@10	NDCG@20
text-embedding-ada-002	LLAMA3-70B	0.0286	0.0414	0.0157	0.0189
text-embedding-ada-002	gpt-3.5-turbo	0.0286	**0.0456**	0.0162	0.0204
all-MiniLM-L6-v2	gpt-3.5-turbo	**0.0329**	**0.0456**	**0.0183**	**0.0216**

Table 3: Performance of traditional recommender system model

Model	Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
SASRec	0.0647	0.1166	0.2195	0.0382	0.0548	0.0808

In our experiments, the best results for information retrieval were achieved using embeddings that excluded movie overviews from Wikipedia.These overviews, although comprehensive, may include irrelevant details that negatively impact similarity searches. While closed-source OpenAI models outperformed open-source alternatives, they were less effective than traditional models like ALS embeddings for information retrieval and the SASRec recommendation model.

Conclusion

We have developed LLM4Rec, a comprehensive framework for integrating and reproducibly evaluating LLM-based components within recommendation systems. While LLM-based recommendation systems currently underperform traditional algorithms in some scenarios, they show promise in addressing challenges such as the cold-start problem and improving conversational recommender systems.

References

[1] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, and W. X. Zhao. Large language models are zero-shot rankers for recommender systems, 2024.

[2] J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, H. Zhang, Y. Liu, C. Wu, X. Li, C. Zhu, H. Guo, Y. Yu, R. Tang, and W. Zhang. How can recommender systems benefit from large language models: A survey, 2024.

[3] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian, Y. Min, Z. Feng, X. Fan, X. Chen, P. Wang, W. Ji, Y. Li, X. Wang, and J.-R. Wen. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms, 2021.

https://github.com/ainura-z/llm-for-rec