Imperial College London
Department of Bioengineering

Final Year Project: Planning Report

Word Count: 3843

Author
Nicolas Dehandschoewercker

Dr. Anil Bharath

June 14, 2025

1 Project Specification

Recommender systems, such as Netflix's personalised film suggestions, have become integral to modern user experiences by filtering vast amounts of data to provide tailored recommendations. However, despite their success, these systems face persistent challenges. One notable issue is the *cold-start problem*, which arises when insufficient user data prevents accurate recommendations. Traditional collaborative filtering techniques rely heavily on user-item interaction data, rendering them ineffective in scenarios where new users or items are introduced.

Additionally, many Classical Recommender Systems (referred to as CRS throughout this report) lack *scrutability* and *interpretability*, despite efforts to introduce these factors into models [1][2]. This means users cannot understand or influence why certain recommendations are made. This lack of transparency can lead to reduced trust in the system, particularly when recommendations deviate from user expectations. For example, a user exploring adventure films may receive an unexpected comedy recommendation without understanding its rationale. Furthermore, in CRS, instead of directly expressing preferences, users must engage in multiple interactions with similar items to gradually shift the CRS' understanding of their preferences (so-called "latent representations").

Traditional recommender systems also operate within a limited knowledge scope, confined to user-item interaction sequences and basic metadata. They lack the broader world knowledge and reasoning capabilities necessary to understand complex user preferences and make nuanced recommendations that account for real-world context and relationships between items.

The emergence of Large Language Models (LLMs) has opened new possibilities for enhancing recommendation systems. These models demonstrate remarkable capabilities in natural language understanding, contextual comprehension, and human-like reasoning. Their ability to leverage extensive pre-trained knowledge while engaging in conversations suggests a path towards what we might call an Artificial General Recommender (AGR) [3], a system capable of providing personalised recommendations across any domain through natural dialogue.

This project proposes a narrative-driven recommendation [4] framework that integrates conversational feedback with LLMs. Through this framework, we aim to improve user engagement, and transparency. For instance, a travel recommender system could elicit preferences about destinations and activities via dialogue, providing a clear rationale behind its suggestions (e.g., "You might like Paris because you enjoy cultural landmarks like the Louvre").

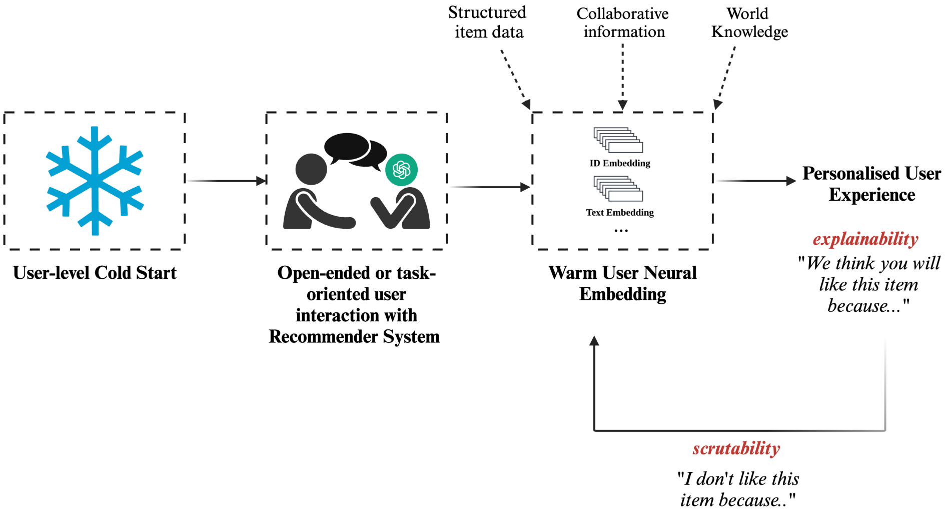

Structured item data
Collaborative information
World Knowledge
ID Embedding
Text Embedding
...
User-level Cold Start
Open-ended or task-oriented user interaction with Recommender System
Warm User Neural Embedding
Personalised User Experience
explainability
"We think you will like this item because..."
scrutability
"I don't like this item because.."

Figure 1. Our proposed framework for cold-start resolution through narrative-driven recommendation.

Our primary objective is to develop a framework consisting of three key components: (1) methods for generating rich user representations through open-ended dialogue, (2) a scrutable memory architecture balancing short-term interests with long-term preferences, and (3) structured item representations capturing hierarchical relationships between recommendable entities.

Due to the absence of specific training data in the conversational recommendations setting, we investigate LLM-powered synthetic *agents* for generating realistic training data, building on recent work by Mysore et al. [4] and Ramos et al. [5] in synthetic user profile generation. We wish to validate that LLMs can act as good approximators of human behaviour in a conversational setting.

We validate our approach in the travel domain, which presents ideal characteristics for testing: hierarchical relationships between items (e.g., destinations and activities), the core requirement for rich world knowledge, and natural alignment with narrative-driven interaction (based on the user's personality traits) [3]. Our research addresses three fundamental questions:

- RQ1: Can open-ended dialogue effectively generate warm-start user embeddings capturing both explicit and implicit preferences?

  We define the subsequent research questions as "stretch objectives" contingent on time constraints and project scope.

- RQ2: What memory architectures best support scrutable and explainable recommendations while balancing varying temporal scales of user interests (long- and short-term interests)?

- RQ3: Can synthetic agents effectively model user conversations to produce synthetic data for training NDR conversational recommenders?

Through this implementation, we aim to demonstrate principles that can generalise to other complex recommendation scenarios and make steps towards Artificial General Recommenders [3].

2 Ethical Analysis

This section analyzes the ethical implications of our research and broader societal impacts of recommender systems. These are summarised in Table 1 below.

Risk Category	**Problem Analysis**	**Mitigation Strategy**
Data Biases and Fairness	Studies show that LLM-based recommender systems can exhibit demographic bias [6] and create unequal opportunities across user groups. The choice of data sources can further amplify these biases.	While bias mitigation is beyond this study's scope, we acknowledge key approaches for production systems: • Implementation of Counterfactually Fair Prompting [7] • Use of recommendation fairness benchmarks (FAIRLLM [8]) • Exclusive use of validated open-source datasets
Opinion Manipulation and Echo Chambers	Recommender systems can limit exposure to diverse viewpoints, contributing to societal polarization. In news recommendations, users may become isolated in "information bubbles" that reinforce existing beliefs while limiting exposure to alternative perspectives. [9]	• Proposition of the implementation of a "temperature" parameter for controlled recommendation diversity • If study objectives are reached, this risk is mitigated, through user control over recommendations through scrutability and explainability features, addressing limitations in current platforms
User Privacy and Conversation Storage	The new conversational nature of the system presents privacy challenges through: • Data leakage via prompt injection [10] • GDPR compliance requirements • Potential exposure to third-party model providers	• Use of open-source models with post-analysis data deletion when user or study subject data is involved. • Explicit user consent protocols for A/B testing segments of the study (if performed, see Project section below). • Fully Homomorphic Encryption for production (outside the scope of this project) to convert user data into a representation that maintains privacy [11] • BPIIA guidelines implementation (outside the scope of this project) [10]

User Autonomy and Financial Well-being	Recommender systems may encourage impulsive purchases and unhealthy financial behaviors, particularly in e-commerce and travel sectors.	• Cooling-off periods for sensitive recommendations (e.g. in gambling, e-commerce) • Detection of sensitive use-cases in LLMs (especially in foundational models specifically geared toward the recommendation task) • Clear commercial content labeling

Table 1. Risk factors relating to this study, and the broader implementation of advanced recommenders in society. Red rows denote high-risk cases, and yellow columns refer to medium-risk cases.

3 Literature Review

3.1 Classical Recommender Systems: brief historical overview

The first iteration of recommender systems, Traditional Collaborative Filtering (1994) [12] introduced the fundamental concept of leveraging user similarities for recommendations. The assumption was that users who agreed once, are likely to agree again in the future. This approach succeeded in basic information filtering but failed to scale effectively with growing user bases.

Linden et al. in 2003, at Amazon, [13] introduced Item-to-Item Collaborative Filtering, which addressed these scalability limitations by focusing on item-item relationships rather than user-user relationships. This increased the processing efficiency of large datasets, adding suitability of recommenders in large-scale commercial applications, though it remained limited in capturing higher-dimensional user preferences.

In 2009, Koren et al. Matrix Factorization (MF) techniques [14], during the Netflix Prize competition, which can be seen as the unification of both prior paradigms. MF decomposes the interaction matrix into the product of two lower-dimensional user and item matrices, capturing latent features, and hence more nuanced user preferences. However, the linear nature of these models limited their ability to model complex relationships.

The integration of deep learning through Two Tower Networks [15] (by YouTube) enabled non-linear processing of rich feature sets. This architecture, with its parallel processing of user and item features, provided a foundation for more sophisticated recommendation models. Neural Collaborative Filtering [16] built upon this by combining matrix factorization principles with neural networks, enabling both linear and non-linear feature interactions.

BERT4Rec (2019), introduced the notion of temporal preferences, taking into account user interaction history in so-called "sequential recommendation" tasks through bidirectional self-attention mechanisms.

In summary, research in recommender systems have reached significant success in predicting user preferences through user-user similiarities, user-item latent feature representations, and temporal patterns. However, key challenges remain in improving *explainability*, *scrutability*, and *semantic understanding* augmented by *world knowledge*. While non-LLM approaches have attempted to address these challenges through Knowledge Graph methods [1][2], they have shown limited improvement in overall explainability.

Given the demonstrated capabilities of Large Language Models (LLMs) across diverse NLP tasks, and the inherent interpretability of text-based inputs, we explore how LLMs can be prompted with natural language descriptions of user preferences to enhance recommendation transparency.

3.2 LLMs in recommender systems

Large Language Models have had an impact in all stages of the recommendation pipeline. Prior work has focused on finding the optimal use of LLMs in one, or multiple stages. The current pipeline is illustrated in Figure 2.

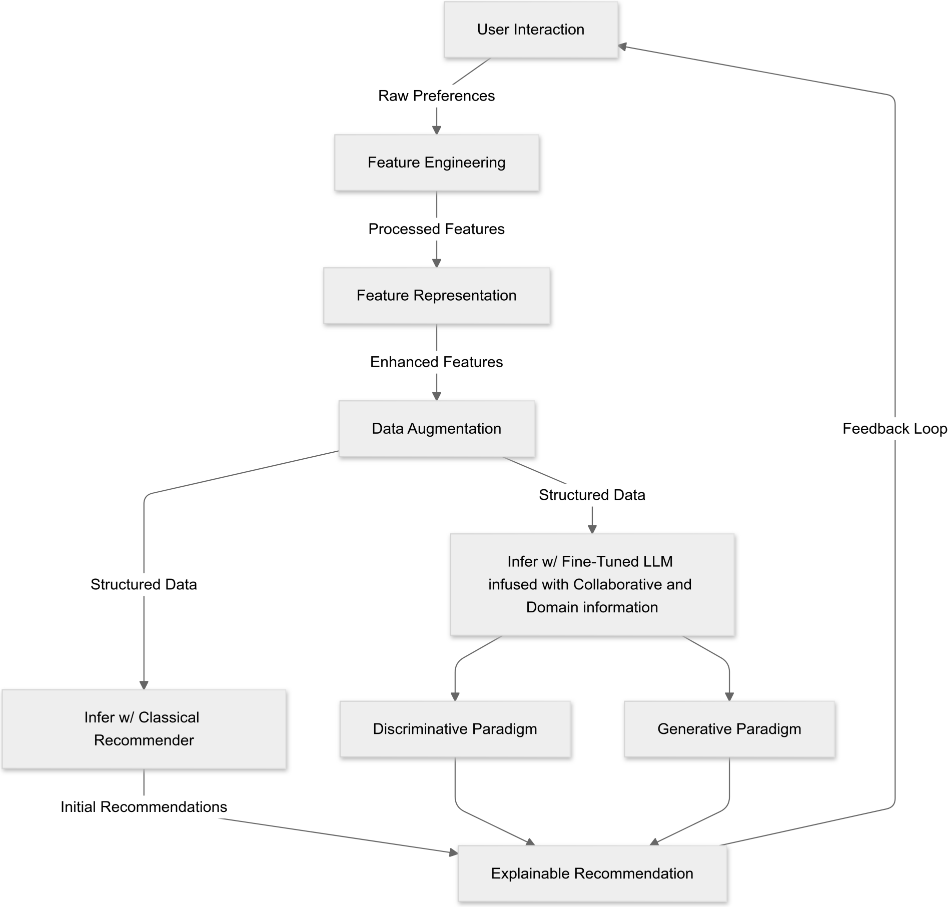


Figure 2. The illustration of deep learning based recommender system pipeline. We characterize the modern recommender system as an information cycle that consists of six stages: user interaction, feature engineering, feature representation, data augmentation, inference (both classical and LLM-based, see section 3.4), and explainable recommendation, which are denoted by different phases in the pipeline. Different studies have focused on using LLMs across different parts of this pipeline.

3.3 LLMs for Feature Engineering, Representation and Data Augmentation

Feature engineering is the process of transforming raw user data into structured formats. In Recommender Systems we are characterising users and items with these features.

Feature Representation is then the conversion of these features into numerical embeddings that capture semantics, collaborative information, item-item relationships and temporal information. Optimality of a feature representation is defined as the balance between the size of the embedding, and the relevance of the final items recommended. For example KALM4Rec, [17] created an item retrieval pipeline using keywords on both the user and item side, combined with a dual approach, using Content-Based Recommendation for fast relevant item retrieval and Message Passing on Graphs to capture higher-order relationships.

Data augmentation is a set of techniques for enriching the original dataset with, additional synthetically generated data points, or enhanced features. For instance, KAR [18] augments user and item embeddings with both factual world-knowledge and reasoning knowledge on user preferences, before converting the embedding back into its numerical form via a hybrid-expert adaptor.

3.4 Model Selection

The integration of LLMs into recommender systems can be characterized through two fundamental design decisions: whether to fine-tune the LLM, and how to incorporate it with classical recommendation models. These decisions create four distinct architectural approaches, as illustrated in Figure 3 The first decision - whether to fine-tune the LLM - influences how effectively the model can adapt to domain-specific recommendation tasks while balancing computational costs and performance. The second decision determines the relationship between the LLM and classical recommender systems, where the LLM can either serve as an augmentation to traditional models (enhancing specific components like feature extraction or explanation generation) or function as the primary recommendation engine with classical models providing supporting signals.

At the time of this paper, research is trending toward using a pure LLM-based approach (Quadrant 4) and not tuning an LLM, and inferring a classical recommender (Quadrant 2) and . We will focus on examples of these in sections 3.4.1 and 3.4.2. In either case, in-domain collaborative knowledge is injected into the LLM.

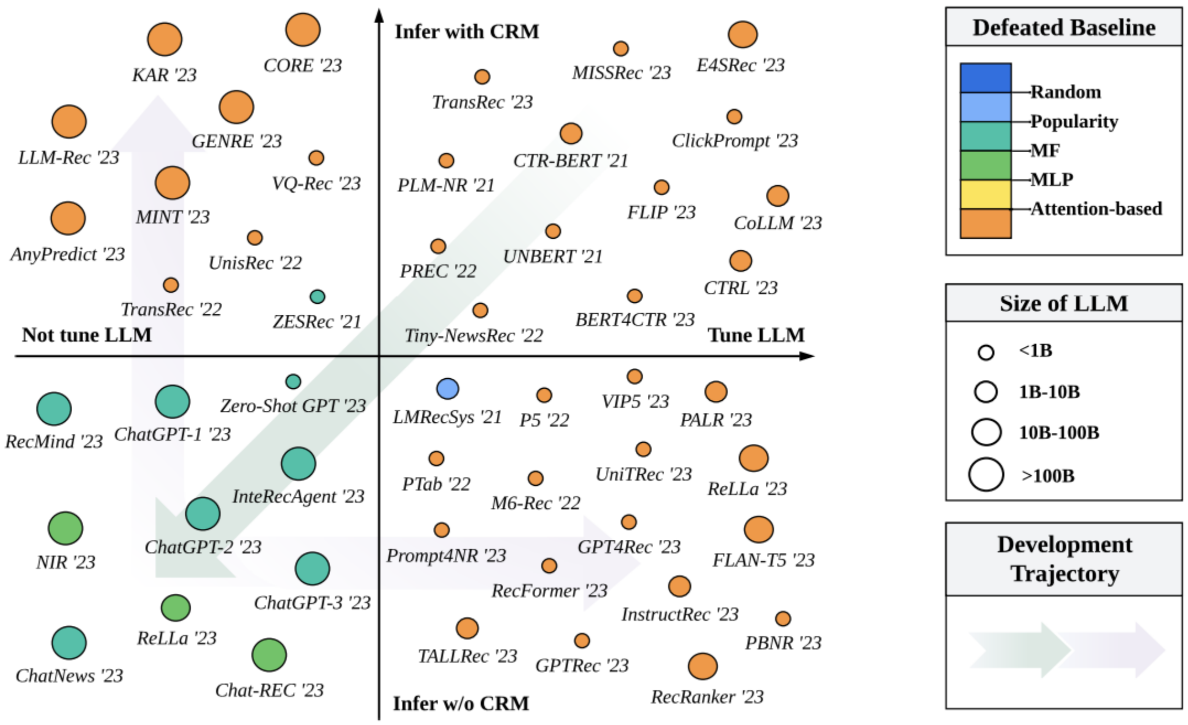


Figure 3. Four-quadrant classification framework for how Large Language Models are adapted to recommender systems. Each circle represents a research work, labeled with the model name below it. The size of the circle corresponds to the largest LLM size used in the study. The color of the circle indicates the baseline model that the corresponding work surpasses, with categories including Random, Popularity, Matrix Factorization (MF), Multi-Layer Perceptron (MLP), and Attention-based models. The horizontal axis differentiates approaches that either tune the LLM (right) or use it without tuning (left), while the vertical axis classifies systems based on whether they integrate classical recommendation models (CRM, above) or function independently of CRM (below). Light-colored arrows indicate the development trajectory of the field. For example, GPT4Rec in quadrant 4 outperforms Attention-based baselines and uses a 1B parameter model. Taken from Fig.4. of [19, p. 3]

The key insight from Figure 3 is the importance of collaborative information. There is a clear performance boundary between quadrant 3 and all other quadrants. The conclusion is that in the absence of a CRS, collaborative knowledge needs to be infused in the LLM at the model-level, through fine-tuning. Quadrant 3 uses "frozen" LLMs (parameters remain unchanged), which is why approaches in this section under-perform relative to other quadrants.

3.4.1 Fine-tuned LLM as the foundational model

When using the LLM as the "foundational" model, there are two key approaches, which are geared toward different recommendation tasks.

In the ***discriminative*** case, the inference is preceded by a retrieval module (e.g. BM25 [20]), and focuses on re-ranking tasks (given a natural-language user input), or item scoring (generating the estimated rating the user would give to an item). Using LLMs in these cases has been shown to have comparable performance to traditional methods, in few-shot, fine-tuned scenarios, but with far greater data efficiency [21].

In the ***generative*** case the LLM generates the items directly from either from within prompts (necessitates a retrieval module) or with general knowledge. Cao et al. [22]. focused on fine-tuning an LLM for sequential recommendation (given a sequence of interactions with items, predicting the next item) introducing pre-training methods such as Masked Item Modeling (MIM) and Bayesian Personal Ranking (BPR), and achieved new SoTa performance in retrieval tasks.

The key advantage of the LLM-as-foundational-model approach, is that it can leverage the LLM's world knowledge, and facilitates explainability and scrutability due to its Natural Language nature. Ramos et al., for instance, [5] focused on *scrutability* in *warm-start scenarios* introducing the User Profile Recommendation (UPR), that entirely replaces latent embeddings with Natural Language Profiles explicitly entered by the user, and a fine-tuned LLAMA model [23] to produce the recommendation.

3.4.2 Classical Recommender System as the foundational model, enhanced by LLM

An alternative approach maintains traditional collaborative filtering architectures as the foundation while using LLMs to enhance specific components. This is done to both better capture collaborative information, enhance the input data with world and user knowledge and provide retrocompatibility with existing recommenders. it also tends to be more computationally efficient that current LLM-only solutions. The key drawback tends to be the loss of explainability.

Systems like KAR [18], and U-BERT [24] exemplify this method. U-BERT first trains its model using reviews from domains with abundant data (like books or electronics) to learn how users typically express their preferences and opinions. It then applies this learned knowledge to make better recommendations in domains with less data (like specialty products).

3.5 LLMs for Synthetic Data and Recommendation Simulation

Although not directly related to the objectives of this study, some studies have attempted to use LLMs as simulators of human behavior, modeling items and users as agents, and using their interactions to understand collaborative information [25] [26]. This is creating new avenues for using agentic solutions to generate synthetic data for NL-based recommendation engines [5], especially given the current lack of a dataset for training conversational recommenders [19].

4 Implementation Plan

This implementation of this project is divided into the following categories.

4.1 Background Research

The background research phase of this project spanned from October 1, 2024, to January 10, 2025, during which we focused on establishing the theoretical and methodological foundation for the work. This included several strategic pivots described in Section 8 and the definition of a commercializable end-goal. During this phase, papers on large language models [23] and recommender systems [19] [27] were pivotal in developing a deep understanding of the field.

The deliverable from this phase is the Planning Report.

4.2 Local Model Setup

The subsequent phase involves the setup of local models and is scheduled to occur from January 11 to January 24, 2025. This stage includes the implementation and initial testing of locally run LLMs for recommendation tasks, complemented by cost estimation and optimization analysis. Tools that will be used for this analysis are LLM elo rankings[1]. We will focus specifically on small (1B-100B), new, high-performing models such as DeepSeek 67B [28], that can be run locally or on Google Colab[2]. We will be paying close attention to the underlying model architecture, which has been shown to influence performance [27].

The cost-estimation will define the project direction given cost constraints, with all possible directions outlined in section 5. This decision will be the deliverable of this phase, along with an Excel cost estimation.

4.3 Testing Benchmark Models

Following the local model setup, the project will focus on testing benchmark models from January 25 to February 10, 2025.

This stage aims to establish baseline performance metrics to reference against in our final analysis, while getting a sense for the technical implementation of each method.

For this, we will use publicly available code from Classical Recommender Systems, with detailed typical benchmarks found in litterature outlined in section 6.

The output will be a modular test suite of evaluation metrics such as precision, recall, and NDCG, which also serves as a risk-mitigation strategy outlined in 5. The objective is to allow us to make more rapid iterations in the short testing phase, to get the best possible final results.

4.4 Feature Representation and Conversation Interface

The design phase for the representation and conversation interface is scheduled from February 11 to March 5, 2025. During this phase, we will be focusing on experimenting with different user and item representations (dependent on decisions made relative to cost, and optimality). This has been shown to have a high impact on the final performance of the recommender system [19]. This will be drawing from insights from KALM4REC [17], and its compact, keyword-based user-item representations. We will also use AgentCF [25] and Wang et. al. [26] for natural language based, context-aware user memory representations, due to the conversational nature or RQ1.

The deliverable will be structured data following the chosen feature representation, extracted from datasets outlined in 6. This phase also serves as a buffer period that we can shorten, depending on time-constraints and unexpected delays.

4.5 Synthetic Data Generation, Training and Validation

From March 6 to March 19, 2025, the project will focus on generating synthetic data to support model training and validation. This phase is informed by work such as UPR [**UPR**] and RecMind [29] which have already demonstrated the utility of synthetic datasets in mitigating data sparsity issues. The output will be a conversational dataset, that will be open-sourced for use in future conversational LLM research projects by the community.

[1] https://lmsys.org/blog/2023-05-03-arena/

[2] https://colab.google/

The synthetic data will be used to train and evaluate model performance, and its generation will transition into a six-week training and validation phase. Between March 20 and May 1, 2025, iterative training, validation, and benchmarking will refine the model. The deliverable for this phase is a set of well-documented training benchmarks that highlight the model's progress.

4.6 Final Report Writing

The final report writing phase will occur from May 2 to June 15, 2025. During this phase, the entire project, including its methods, results, and discussions, will be documented. The final deliverable for this phase is the complete and submitted final report.

4.7 Paper Writing and Publishing

Following the completion of the final report, the project will, depending on the quality of the results, transition into the paper writing and publishing phase, scheduled from June 16 to September 15, 2025.The paper will aim for submission to a peer-reviewed journal or conference. The extended timeline for this phase allows for a thorough review and refinement process to ensure high-quality output.

4.8 Final project outcomes

This section outlines best-case scenario outcomes. Outcomes of contingency plans are outlined in the Section 5.

- A functional NDR-based recommender system with appropriate benchmarks.
- A conversational dataset for use in subsequent NDR Recommender System research.
- A final report and presentation showcasing the results of the study.

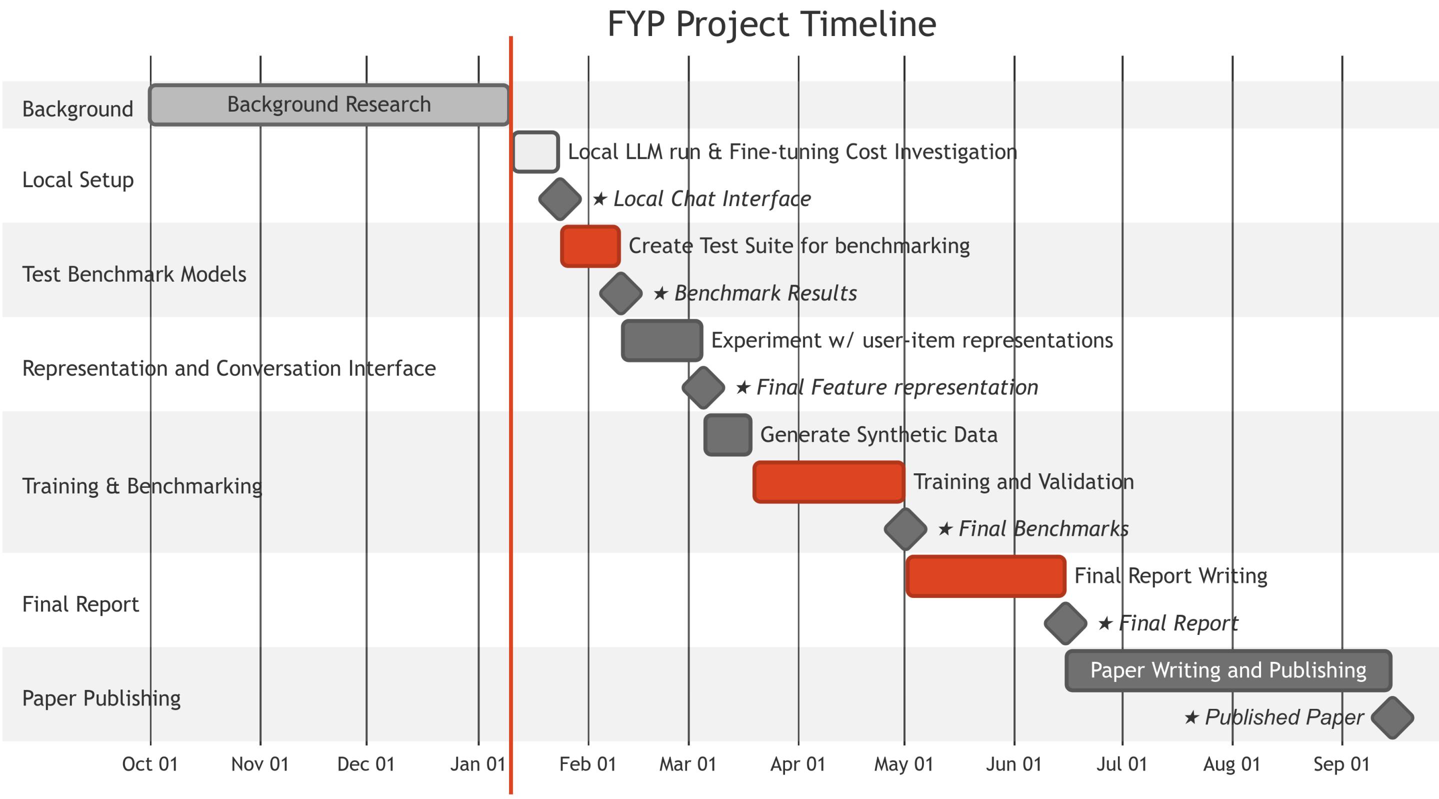


Figure 4. Gantt Chart of Project timeline

5 Risk Register

Key risks regarding the outlined implementation plan, along with contingencies are outline in Figure 5.

The contingency plan begins with a cost estimation of fine-tuning large language models for recommendation tasks.

If fine-tuning is too expensive, the project will use a classical recommender system enhanced by non fine-tuned LLMs, leveraging pre-trained models.

If fine-tuning costs are acceptable, an LLM-only recommender will be developed to maximize generative and reasoning capabilities. Should both approaches prove too time-consuming, the focus will shift to a much simpler conversational recommender system, integrating methods from the RecMind (conversational)[29] and UPR [5] (one prompt input) papers to create an "interactive" iteration of the original method.

If the conversational recommender is also infeasible within the timeframe, the project will adopt a simpler approach by developing an LLM-powered recommender system using ELO ranking, which has been outlined as being a key need in the field [19].

In all cases, synthetic conversational data generation is a core deliverable, providing valuable resources for training and validation. This contingency plan ensures progress and meaningful contributions, regardless of the chosen pathway.

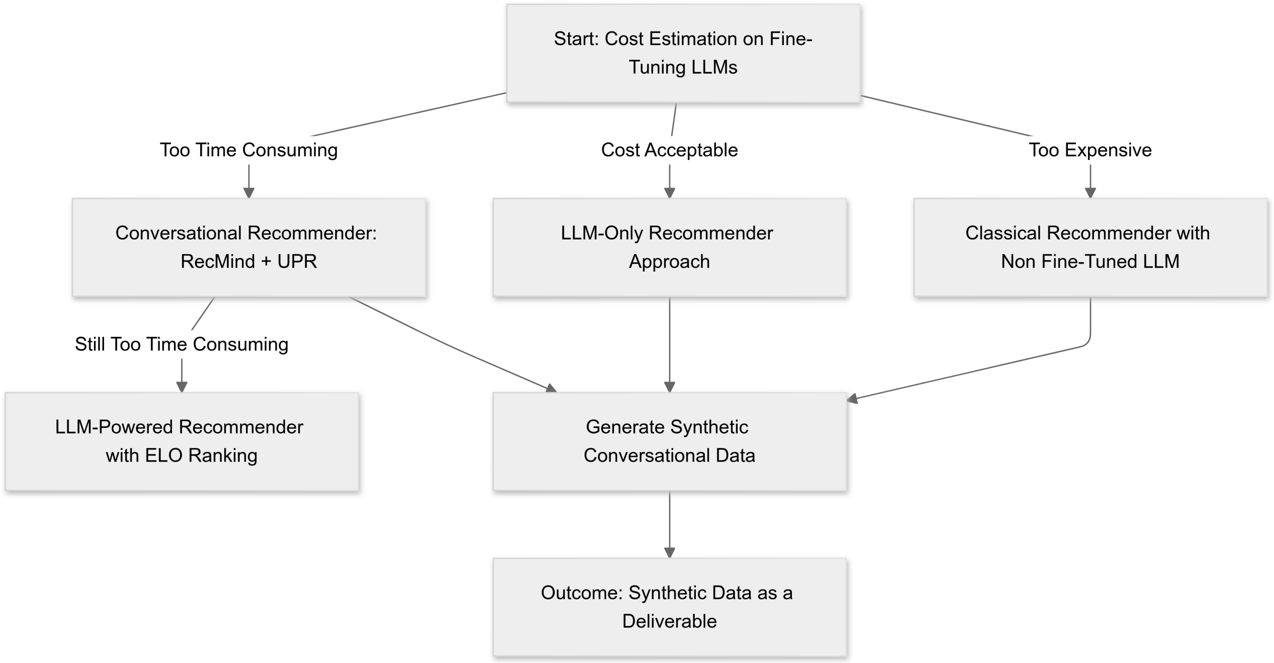

Start: Cost Estimation on Fine-Tuning LLMs
Too Time Consuming
Cost Acceptable
Too Expensive
Conversational Recommender: RecMind + UPR
LLM-Only Recommender Approach
Classical Recommender with Non Fine-Tuned LLM
Still Too Time Consuming
LLM-Powered Recommender with ELO Ranking
Generate Synthetic Conversational Data
Outcome: Synthetic Data as a Deliverable

Figure 5. Issue tree of contingency plans in place

Other potential risks are outlined in Table 2.

Table 2. Additional Risks and Mitigation Strategies

Risk	**Mitigation Strategy**
Synthetic data generation fails to produce meaningful results	Conduct a detailed analysis to identify reasons for failure, including data preprocessing issues or model limitations, and focus the project on documenting these findings as a contribution. Alternatively, time-allowing, set up A/B testing platform and find a suitable number of test subjects to validate the model on.
Travel domain proves inappropriate for recommendation tasks	Switch to a different domain, such as Steam games, while applying the same methods and frameworks to ensure continuity of the approach.
Travel domain remains infeasible for results-driven recommendations	Shift focus to benchmarking different user memory representation methods (e.g., based on keywords) and compare them with a single-user profile baseline for robust evaluation.
Time constraints impact progress	Reduce time spent optimising user representations by abandoning stretch research questions (e.g., RQ2 and RQ3). Instead, focus on finalizing one fixed representation method and conducting thorough analysis within the remaining timeframe.

6 Evaluation

7 Evaluation Methods

We follow established benchmarks and metrics commonly used in recommender systems research, ensuring alignment with state-of-the-art approaches. Classical methods such as matrix factorization neural collaborative filtering [16], and LightGCN [30] will serve as baselines, as these are frequently used in similar studies [17] [18] [25]. More advanced frameworks, including non fine-tuned GPT-4-based conversational models and knowledge-augmented systems, may also be referenced for comparison.

The primary metric for ranking quality will be Normalized Discounted Cumulative Gain, which evaluates both the relevance of recommendations and their ranking order. Precision, recall, and hit ratio ($HR@k$) will quantify the retrieval of relevant items. Coverage will measure the diversity of recommendations, and Area Under the Curve will assess performance in cold-start scenarios.

Finally, ablation studies will be performed, quantifying the correlation between performance, and each of the final architectural components.

8 Preliminary Results

The initial phase of this research focused on developing a comprehensive understanding of classical recommendation systems and their evolution. As outlined in Section 3.1, we began by examining traditional collaborative filtering approaches from GroupLens [12] and their progression through matrix factorization techniques [14] to deep learning architectures with Two Tower Networks [15].

Our research direction initially explored knowledge graph-based approaches for enhancing recommendation explainability, following methods similar to those described by Bellini et al. [1] and Li et al. [2]. However, after analyzing recent developments in LLM applications for recommendation systems, particularly the emergence of narrative-driven recommendation frameworks [4], we identified a more promising direction for addressing our core research questions around user preference capture and recommendation transparency.

To validate our technical capabilities and test initial hypotheses about conversational recommendation systems, we developed a proof-of-concept (PoC) implementation using LangChain[3] and LangGraph[4] frameworks, integrated with Firebase[5] for persistent storage. This prototype focused on:

- Implementing open-ended dialogue management for preference elicitation
- Developing a structured memory architecture (segmented in different categories) to maintain user context
- Testing different approaches to storing and retrieving user preferences during extended conversations

The PoC demonstrated the feasibility of maintaining coherent user representations across multiple interaction sessions, though it also highlighted key challenges in optimal preference representation that informed our subsequent research direction. While the implementation successfully maintained conversation context and user preferences, it revealed the need for more sophisticated embedding techniques to capture the nuanced relationships between user preferences and recommended items.

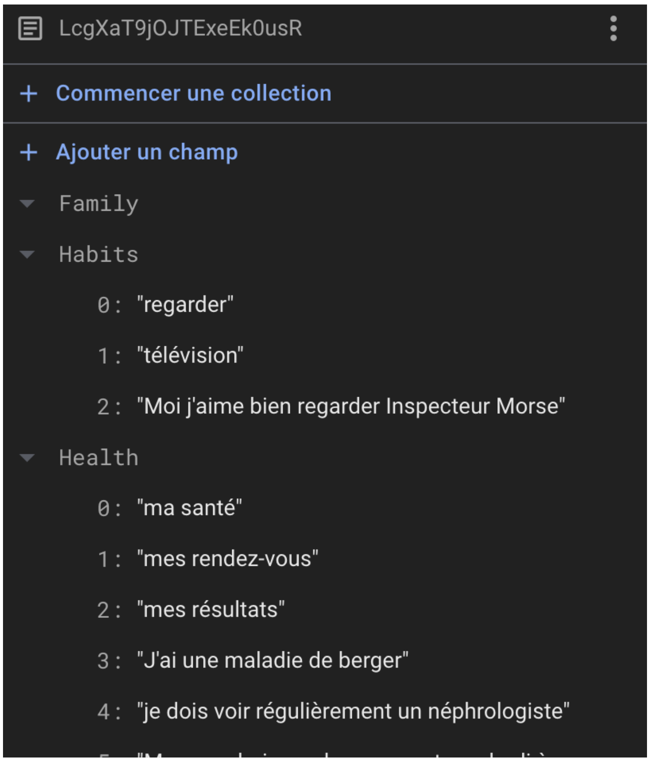
This early experimentation proved invaluable in shaping our current approach, particularly in understanding the practical challenges of implementing theoretical concepts from papers such as KAR [18] and KALM4Rec [17]. It also informed our decision to focus on the specific architectural components outlined in Section 4, particularly the emphasis on structured user representations and memory architecture.

The lessons learned from this preliminary work have been incorporated into our current implementation plan, particularly in the Feature Representation and Conversation Interface phase described in Section 4.4, where we will build upon these initial experiments with more sophisticated representation techniques drawing from recent work in user behavior simulation [26] and autonomous language agents [25].

[3] https://www.langchain.com/

[4] https://www.langchain.com/langgraph

[5] https://firebase.google.com/



```
Family
Habits
    0: "regarder"
    1: "télévision"
    2: "Moi j'aime bien regarder Inspecteur Morse"
Health
    0: "ma santé"
    1: "mes rendez-vous"
    2: "mes résultats"
    3: "J'ai une maladie de berger"
    4: "je dois voir régulièrement un néphrologiste"
```

Figure 6. Firebase record for Proof of Concept user memory representation after open-ended user conversation, built with LangChain, LangGraph, Firebase, FastAPI, and LangServe

References

[1] Vito Bellini et al. “Knowledge-aware Autoencoders for Explainable Recommender Systems”. In: *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*. DLRS 2018. Vancouver, BC, Canada: Association for Computing Machinery, 2018, pp. 24–31. ISBN: 9781450366175. DOI: 10.1145/3270323.3270327. URL: https://doi.org/10.1145/3270323.3270327 (pages 1, 4, 11).

[2] Chen Li et al. “Ripple Knowledge Graph Convolutional Networks for Recommendation Systems”. In: *Machine Intelligence Research* 21.3 (2024), pp. 481–494. DOI: 10.1007/s11633-023-1440-x. URL: https://doi.org/10.1007/s11633-023-1440-x (pages 1, 4, 11).

[3] Guo Lin and Yongfeng Zhang. “Sparks of Artificial General Recommender (AGR): Early Experiments with ChatGPT”. 2023. arXiv: 2305.04518 [cs.IR]. URL: https://arxiv.org/abs/2305.04518 (pages 1, 2).

[4] Sheshera Mysore, Andrew Mccallum, and Hamed Zamani. “Large Language Model Augmented Narrative Driven Recommendations”. In: *Proceedings of the 17th ACM Conference on Recommender Systems*. RecSys ’23. Singapore, Singapore: Association for Computing Machinery, 2023, pp. 777–783. ISBN: 9798400702419. DOI: 10.1145/3604915.3608829. URL: https://doi.org/10.1145/3604915.3608829 (pages 1, 2, 11).

[5] Jerome Ramos et al. “Transparent and Scrutable Recommendations Using Natural Language User Profiles”. 2024. arXiv: 2402.05810 [cs.IR]. URL: https://arxiv.org/abs/2402.05810 (pages 2, 7, 10).

[6] Yashar Deldjoo. “Understanding Biases in ChatGPT-based Recommender Systems: Provider Fairness, Temporal Stability, and Recency”. 2024. arXiv: 2401.10545 [cs.IR]. URL: https://arxiv.org/abs/2401.10545 (page 3).

[7] Yongqi Li et al. “Prompting Large Language Models for Counterfactual Generation: An Empirical Study”. 2024. arXiv: 2305.14791 [cs.CL]. URL: https://arxiv.org/abs/2305.14791 (page 3).

[8] Jizhi Zhang et al. “Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation”. In: *Proceedings of the 17th ACM Conference on Recommender Systems*. RecSys ’23. ACM, Sept. 2023, pp. 993–999. DOI: 10.1145/3604915.3608860. URL: http://dx.doi.org/10.1145/3604915.3608860 (page 3).

[9] Yougang Lyu et al. “Cognitive Biases in Large Language Models for News Recommendation”. 2024. arXiv: 2410.02897 [cs.IR]. URL: https://arxiv.org/abs/2410.02897 (page 3).

[10] Jingwei Yi et al. “Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models”. 2024. arXiv: 2312.14197 [cs.CL]. URL: https://arxiv.org/abs/2312.14197 (page 3).

[11] Ingonyama. “Solving LLM Privacy with FHE”. Accessed: 2025-01-07. Aug. 2023. URL: https://medium.com/@ingonyama/solving-llm-privacy-with-fhe-3486de6ee228 (page 3).

[12] Paul Resnick et al. “GroupLens: an open architecture for collaborative filtering of netnews”. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. CSCW ’94. Chapel Hill, North Carolina, USA: Association for Computing Machinery, 1994, pp. 175–186. ISBN: 0897916891. DOI: 10.1145/192844.192905. URL: https://doi.org/10.1145/192844.192905 (pages 4, 11).

[13] Greg Linden, Brent Smith, and Jeremy York. “Amazon.com Recommendations: Item-to-Item Collaborative Filtering”. In: *IEEE Internet Computing* 7.1 (2003), pp. 76–80. DOI: 10.1109/MIC.2003.1167344 (page 4).

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. “Matrix Factorization Techniques for Recommender Systems”. In: *Computer* 42.8 (2009), pp. 30–37. DOI: 10.1109/MC.2009.263 (pages 4, 11).

[15] Paul Covington, Jay Adams, and Emre Sargin. “Deep Neural Networks for YouTube Recommendations”. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys ’16. Boston, Massachusetts, USA: Association for Computing Machinery, 2016, pp. 191–198. ISBN: 9781450340359. DOI: 10.1145/2959100.2959190. URL: https://doi.org/10.1145/2959100.2959190 (pages 4, 11).

[16] Xiangnan He et al. “Neural Collaborative Filtering”. 2017. arXiv: 1708.05031 [cs.IR]. URL: https://arxiv.org/abs/1708.05031 (pages 4, 11).

[17] Hai-Dang Kieu et al. “Keyword-driven Retrieval-Augmented Large Language Models for Cold-start User Recommendations”. 2024. arXiv: 2405.19612 [cs.IR]. URL: https://arxiv.org/abs/2405.19612 (pages 5, 8, 11, 12).

[18] Yunjia Xi et al. “Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models”. 2023. arXiv: 2306.10933 [cs.IR]. URL: https://arxiv.org/abs/2306.10933 (pages 5, 7, 11, 12).

[19] Jianghao Lin et al. “How Can Recommender Systems Benefit from Large Language Models: A Survey”. 2024. arXiv: 2306.05817 [cs.IR]. URL: https://arxiv.org/abs/2306.05817 (pages 6–8, 10).

[20] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond”. In: *Foundations and Trends in Information Retrieval* 3 (Jan. 2009), pp. 333–389. DOI: 10.1561/1500000019 (page 7).

[21] Wang-Cheng Kang et al. “Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction”. 2023. arXiv: 2305.06474 [cs.IR]. URL: https://arxiv.org/abs/2305.06474 (page 7).

[22] Yuwei Cao et al. “Aligning Large Language Models with Recommendation Knowledge”. 2024. arXiv: 2404.00245 [cs.IR]. URL: https://arxiv.org/abs/2404.00245 (page 7).

[23] Hugo Touvron et al. “LLaMA: Open and Efficient Foundation Language Models”. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971 (pages 7, 8).

[24] Zhaopeng Qiu et al. “U-BERT: Pre-training User Representations for Improved Recommendation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.5 (May 2021), pp. 4320–4327. DOI: 10.1609/aaai.v35i5.16557. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16557 (page 7).

[25] Junjie Zhang et al. “AgentCF: Collaborative Learning with Autonomous Language Agents for Recommender Systems”. 2023. arXiv: 2310.09233 [cs.IR]. URL: https://arxiv.org/abs/2310.09233 (pages 7, 8, 11, 12).

[26] Lei Wang et al. “User Behavior Simulation with Large Language Model based Agents”. 2024. arXiv: 2306.02552 [cs.IR]. URL: https://arxiv.org/abs/2306.02552 (pages 7, 8, 12).

[27] Lanling Xu et al. “Prompting Large Language Models for Recommender Systems: A Comprehensive Framework and Empirical Analysis”. In: *ArXiv* abs/2401.04997 (2024). URL: https://api.semanticscholar.org/CorpusID:266902921 (page 8).

[28] DeepSeek-AI et al. “DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model”. 2024. arXiv: 2405.04434 [cs.CL]. URL: https://arxiv.org/abs/2405.04434 (page 8).

[29] Yancheng Wang et al. “RecMind: Large Language Model Powered Agent For Recommendation”. 2024. arXiv: 2308.14296 [cs.IR]. URL: https://arxiv.org/abs/2308.14296 (pages 8, 10).

[30] Xiangnan He et al. “LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation”. 2020. arXiv: 2002.02126 [cs.IR]. URL: https://arxiv.org/abs/2002.02126 (page 11).