TRAIL Summer Workshop' 25
Project Proposal

Full Name of Team Leader	Julien ALBERT, PhD Student, UNamur
Project Title	Aligning Recommendation Explanations to User Preferences Using LLMs Fine-Tuned by Reinforcement Learning with AI Feedback
Profile of the Team Leader(s)	Yasmine Akaichi: PhD student (UNamur), thesis on reliable privacy-preserving AI for federated learning framework, expertise in business intelligence and artificial intelligence. Julien Albert: PhD student (UNamur), thesis on the use of XAI to address the filter bubble effect in recommender systems, expertise on recommender systems, XAI, UX research and design, human-centered research, HCI. Martin Balfroid: PhD student (UNamur), thesis on evaluating LLM-generated explanations to support the onboarding of new developers. Lluc Bono Rosselló: PhD student (ULB), thesis on complex systems in music and creativity, expertise in network science, complex systems and multi-agent simulations. Lucile Dierckx: PhD student (UCLouvain), thesis on neurosymbolic learning with a focus on learning rules in a neural framework and learning from complex logical constraints. Sédrick Stassin: PhD (UMONS), thesis on explainability for computer vision with a focus on the evaluation of explainability methods and dedicated methods to explain reliably vision-language transformers. Field expertise in XAI evaluation and methods, vision, and vision-language models. Vincent Stragier: Reasearch Assistant at the University of Mons, ISIA LAB. Working on health-related projects
Abstract	Recommender systems still fail to explain their recommendations in a transparent way, limiting user trust and acceptance. Building on our interactive web platform developed during the last workshop, this year's project investigates to what extent we can produce higher-quality, user-aligned explanations by aligning LLMs with Reinforcement Learning from AI feedback (RLAIF). These explanations will be evaluated against user-centered criteria, such as satisfaction, scrutability, and transparency. Our objective is to advance the explainability of recommender systems by aligning explanation strategies with human preferences at a low cost. All outcomes, including the evaluation tools, codebase, and trained models, will be released via the TRAIL Factory.
Project Objectives	*Why did this get recommended to me?* is a question we still ask ourselves every day. In an age where algorithms curate much of what we see, recommender systems still lack transparency. To address this, recent research has focused on making recommender systems more transparent and explainable. One promising approach is to leverage large language models (LLMs), which, thanks to their pre-training on vast text corpora, can produce clear and human-readable explanations. Different aspects around the *generation of textual explanations using LLMs, based on outputs of explainable recommender systems*, were explored during the 2023's and 2024's TRAIL Workshops. In 2023, participants compared classic template-based

	explanations to dynamic, LLM-based ones, experimenting with template rephrasing and knowledge graph integration. The results suggested that the LLM explanations are more nuanced and better alig ned with the user expectations, highlighting key limitations in existing recommendation methods. In 2024, efforts focused on developing a working prototype: a web-based interface that combined graph-based recommendation with LLM-enhanced explanations. Additional work explored the integration of several recommendation methods and LLM into the framework, as well as multiple LLM deployment, training, and prompting strategies. These foundations now enable targeted investigations on how the various components of the framework can influence user trust and interaction with the recommender system. This year's project builds on the existing framework to *explore a new direction*: enhancing explanation generation through fine-tuned LLMs. The goal is to assess how Reinforcement Learning with AI Feedback (RLAIF) can improve explanations with respect to user-centered criteria, such as Tintarev's goals [18]. We will therefore compare our methods with several explanation strategies using a user-based evaluation approach. The aim of this project is to improve user experience and trust, aligning with TRAIL's work-packages on AI-in-the-loop (WP1.2) and user trust (WP2.5). This research opens a new line of study beyond past projects, aiming to assess how fine-tuning and feedback-driven strategies can improve explanations in recommender systems.
Project Dataset	In this project, we use the MovieLens dataset, a widely recognized benchmark in the field of recommender systems. It is provided by the GroupLens research lab at the University of Minnesota and is publicly accessible at the following URL: https://grouplens.org/datasets/movielens/. The dataset contains rich user-item interaction data, including explicit movie ratings on a 1–5 scale, timestamps, and metadata such as movie titles, genres, and optional user demographic information (age, gender, occupation, and zip code) depending on the version. The data is available in various sizes (100K, 1M, 10M, 20M, 25M ratings), provided in CSV format, and is well-structured for rapid experimentation. MovieLens is particularly suited for research in explainable recommendations due to its clean schema, reproducibility, and the diversity of user preferences it captures. Its extensive use in academic literature also allows for meaningful comparisons with existing work. We used this dataset for our initial experiments, and moving forward, we plan to continue using it to assess both the performance and explainability of our system under realistic and reproducible conditions. Moreover, recommendation can be reframed as a sequential decision-making problem [6], where an agent learns which item to recommend next in order to maximize user satisfaction (e.g., rating or click probability). This is particularly relevant in reinforcement learning with logged data (a form of offline RL) and in simulated environments derived from MovieLens data. In such scenarios, the dataset is typically adapted to an RL-friendly format, where the state represents the user's interaction history, the action is the next item to recommend, and the reward is the rating or a derived utility function. Self-Instruct [20] is an iterative method for generating large, diverse datasets from a small pool of human-written instructions—e.g., "Explain why the system recommended this movie" in the context of movie recommendation explanations. The process involves three steps: (1) generating new instructions using a few-shot prompting strategy with a mix of initial human-written and LLM-generated instructions; (2) generating corresponding input-output pairs—starting from the input (e.g., the recommendation) to the output

	(e.g., the explanation); and (3) filtering out low-quality or redundant data using criteria such as text similarity and banned keywords.
Background Information	Recommender systems were developed to address the problem of information overload, a widespread phenomenon on the Internet affecting various fields. They regroup a set of information filtering techniques whose purpose is to propose a selection of items from a generally large corpora to a user [10]. These items are chosen based on the user's preferences and characteristics, deduced from the history of their interactions with items of the given corpus. However, users sometimes do not accept the suggested recommendations for diverse reasons, such as a lack of trust in the recommendation system or a lack of transparency in the recommendation process [18]. Thus, methods were developed to provide explanations alongside recommendations to improve user acceptance and, more generally, user experience. Explainability is, therefore, an important research concern in recommendation systems [24]. In this workshop, we focus on graph-based explainable recommendation techniques [11] and textual explanations. Large language models (LLMs) [25] are trained to predict the next 'word', which, surprisingly, makes them highly capable of performing a variety of natural language processing (NLP) tasks such as text completion, translation, or summarization. They are based on the transformer architecture [19], with the decoder-only architecture being the most dominant thanks to the abilities of the GPT family (e.g., GPT-4 [16], DeepSeek v3 [23], Llama 4 [14], etc.). Although they can be fine-tuned for more specialized tasks [4], they opened up a new paradigm in which adapting to a new task can simply consist of reformulating the prompt in a particular way [13], which is often referred to as prompt engineering. A popular technique is few-shot learning or in-context learning, which prepends the prompt with a few examples of solved tasks [5]. This versatility means that they can be used in a wide range of fields, including recommendation systems [12]. Although fine-tuning further specializes a model on a downstream task, it does not guarantee better alignment with non-functional requirements [17], which is what the Tintarev and Mashtoff's criteria are [18]. A solution is to add a reinforcement learning layer on top of fine-tuning, such as Reinforcement Learning from Human Feedback (RLHF), which involves fine-tuning a model by rewarding outputs that align with human preferences on non-functional requirements. Although RLHF [17] improves alignment with human preferences, its reliance on human annotations makes it impractical if you have limited resources. To address the reliance of RLHF on humans, Reinforcement Learning from AI Feedback (RLAIF) leverages LLM-as-a-Judge methodologies, which replace human annotations with judgments from larger models, to reduce annotation costs while still improving alignment [1]. A core component of RLAIF is knowledge distillation, where larger LLMs generate synthetic datasets to fine-tune smaller models iteratively [20, 22].
Bibliographic References	[1] Bai, Yuntao, et al. "Constitutional ai: Harmlessness from ai feedback." *arXiv preprint arXiv:2212.08073* (2022). [2] Balloccu, Giacomo, et al. "Hands on explainable recommender systems with knowledge graphs." *Proceedings of the 16th ACM Conference on Recommender Systems*. 2022. [3] Balog, Krisztian, and Filip Radlinski. "Measuring recommendation explanation quality: The conflicting goals of explanations." *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 2020.

[4] Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021).
[5] Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.
[6] Chen, Xiaocong, et al. "Intrinsically motivated reinforcement learning based recommendation with counterfactual data augmentation." *World Wide Web* 26.5 (2023): 3253-3274.
[7] Chiang, Wei-Lin, et al. "Chatbot arena: An open platform for evaluating llms by human preference." *Forty-first International Conference on Machine Learning*. 2024.
[8] Cropper, Andrew, and Sebastijan Dumančić. "Inductive logic programming at 30: a new introduction." *Journal of Artificial Intelligence Research* 74 (2022): 765-850.
[9] Cropper, Andrew, and Rolf Morel. "Learning programs by learning from failures." *Machine Learning* 110.4 (2021): 801-856.
[10] de Gemmis, Marco, Pasquale Lops, and Marco Polignano. "Recommender Systems, Basics of." *Encyclopedia of Social Network Analysis and Mining*. Springer, 2018. 2125-2137.
[11] Gao, Chen, et al. "A survey of graph neural networks for recommender systems: Challenges, methods, and directions." *ACM Transactions on Recommender Systems* 1.1 (2023): 1-51.
[12] Lin, Jianghao, et al. "How can recommender systems benefit from large language models: A survey." *ACM Transactions on Information Systems* 43.2 (2025): 1-47.
[13] Liu, Pengfei, et al. "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." *ACM computing surveys* 55.9 (2023): 1-35.
[14] Meta Llama. "meta-llama/Llama-4-Maverick-17B-128E." (2025). https://huggingface.co/meta-llama/Llama-4-Maverick-17B-128E
[15] Muggleton, Stephen. "Inverse entailment and Progol." *New generation computing* 13 (1995): 245-286.
[16] Achiam, Josh, et al. "Gpt-4 technical report." *arXiv preprint arXiv:2303.08774* (2023).
[17] Ouyang, Long, et al. "Training language models to follow instructions with human feedback." *Advances in neural information processing systems* 35 (2022): 27730-27744.
[18] Tintarev, Nava, and Judith Masthoff. "Explaining recommendations: Design and evaluation." *Recommender systems handbook*. Boston, MA: Springer US, 2015. 353-382.
[19] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
[20] Wang, Yizhong, et al. "Self-instruct: Aligning language models with self-generated instructions." *arXiv preprint arXiv:2212.10560* (2022).
[21] Xian, Yikun, et al. "Reinforcement knowledge graph reasoning for explainable recommendation." *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 2019.
[22] Xu, Can, et al. "WizardLM: Empowering large pre-trained language models to follow complex instructions." *The Twelfth International Conference on Learning Representations*. 2024.
[23] Yang, Chars. "DeepSeek v3 - Advanced AI & LLM Model Online". (2024). https://deepseekv3.org/
[24] Zhang, Yongfeng, and Xu Chen. "Explainable recommendation: A survey and new perspectives." *Foundations and Trends® in Information Retrieval* 14.1 (2020): 1-101.

	[25] Zhao, Wayne Xin, et al. "A survey of large language models." *arXiv preprint arXiv:2303.18223* 1.2 (2023).
Detailed Work Plan	**Work packages**: **WP1**: Implementation of the recommendation system **WP1.1**: Graph-based recommendation **WP1.2**: ILP-based recommendation **WP2**: Design and implementation of the explanation method based on LLMs Fine-Tuned by RLAIF **WP3**: Evaluation of the explanation method using a user-based approach **WP4**: Communication of the work done We plan to organize a one-week workshop in June to prepare the Trail Workshop 2025. The aim is to put in place the necessary technical elements, in particular the recommendation system (**WP1**), to enable us to concentrate on the RLAIF part of the workshop. During the workshop, the first day would be dedicated to introducing the project topic and objectives in order to share a common vision. Thanks to this, the implementation and experimentation work will be defined and allocated to the team's members, according to their expertise and their affinity. The first two-thirds of the workshop will be devoted to implementation (**WP2**), and the last third to experimentation, including user-based evaluation (**WP3**). After the workshop, the results would be valorized through a workshop publication. The material (documentation, source code and results) will be made available on the TRAIL Factory (**WP4**). Further technical details about the work packages are given below. **WP1.1**: For the technical description, as mentioned earlier, we are focusing on graph-based explainable recommendation techniques. Ballocu et al. [2] introduces in a tutorial basic concepts and some state-of-the-art methods. During the workshop, we plan to use the Policy-Guided Path Reasoning (PGPR) method proposed by Xian, et al. [21]. As explained by Ballocu et al. [2], PGPR is "based on the idea of training a reinforcement learning (RL) agent for finding paths between users and candidate items. During training, the agent starts from a user and learns to reach the correct items (i.e., the best recommendations), with high rewards. During inference, the agent directly walks to correct items for recommendation, without enumerating all the paths between users and items." **WP1.2**: In parallel to graph-based recommendation techniques like PGPR, we will evaluate the use of Inductive Logic Programming (ILP) as a complementary approach to extract symbolic rules that justify recommendations. ILP is a form of machine learning rooted in logic programming, where the goal is to learn interpretable rules from examples and background knowledge [8]. In the context of recommendation, user-item interactions, item attributes, and contextual metadata can be encoded as logical facts. From these, ILP systems such as Progol [15] or Popper [9] can infer rules such as *recommended(User, Movie) :- liked(User, Genre), has_genre(Movie, Genre)*. Such rules offer structured and semantically meaningful explanations that are easy to inspect and validate. These symbolic outputs can either be presented directly to users or translated into natural language via an LLM,

	providing a hybrid pipeline in which ILP ensures logical consistency and explainability, while LLMs enhance fluency and contextual richness. During the workshop, we plan to test the integration of ILP systems with our LLM-based explanation interface. **WP2**: We have not yet chosen the LLM models that will be used during the workshop, as new—and potentially better—models will likely be released by September. However, we will consider several factors when selecting them: (performance) benchmark results from the Chatbot Arena leaderboard (https://chat.lmsys.org/), based on an Elo-style evaluation method proposed by a team from Stanford [7]; (cost) pricing varies from pay-per-token, subscription-based access to cloud-based models or running bigger models in-house; (legal constraints) terms of use, as many companies prohibit using their model outputs to train other models. **WP3**: As mentioned in the project objectives, we plan to evaluate the explanations using a user-based approach. In particular, we will focus on satisfaction, scrutability and transparency as these properties are the most likely to provide the best overall picture of the quality of explanations [3]. To do so, we will integrate the explainable recommendation pipeline into a lightweight web application that will allow users to compare different types of explanations w.r.t. the above-mentioned properties, similarly to the Chatbot Arena platform [7] (available at: https://lmarena.ai/). This first prototype will allow us to conduct a first evaluation during the workshop. In a second phase, probably after the workshop, we also plan to combine the recommendation procedure and the LLM-enhanced explanation into a prototype that will take the form of a web application, allowing an end-user to interact freely and in quasi-real-time with the recommendation system and its explanations. Thanks to this prototype, we will conduct user testing to delve deeper into the assessment of the above-mentioned properties. Regarding the **resources needed**, specific computational resources may be required depending on the chosen models and the possibility of having to train or fine-tune some of them. Practically, we plan to investigate 2 options: the deployment of a model on the Lucia cluster provided by the CÉCI, and the use of a third-party API (potentially chargeable). Concerning the prototype deployment for user testing, the options considered are the TRAIL Factory and internal university resources. Those needs would be investigated before the workshop. We also plan to involve other workshop participants in user testing during the workshop. Finally, the most important is perhaps whiteboards, sticky notes, tea, and coffee!
Other Remarks	