



HAL
open science

EMiGRe: Unveiling why your recommendations are not what you expect

Herve-Madelein Attolou, Katerina Tzompanaki, Kostas Stefanidis, Dimitris Kotzinos

► **To cite this version:**

Herve-Madelein Attolou, Katerina Tzompanaki, Kostas Stefanidis, Dimitris Kotzinos. EMiGRe: Unveiling why your recommendations are not what you expect. IEEE 40th International Conference on Data Engineering, Jun 2024, Tempere, Finland. pp.419-423, 10.1007/978-3-031-62362-2_37. hal-04732438

HAL Id: hal-04732438

<https://hal.science/hal-04732438v1>

Submitted on 11 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

EMiGRe: Unveiling why your recommendations are not what you expect.

Herve-Madelein Attolou¹, Katerina Tzompanaki¹, Kostas Stefanidis², and
Dimitris Kotzinos¹

¹ ETIS, CY Cergy Paris University, ENSEA, CNRS UMR8051

² Tampere University

Abstract. This demonstration showcases *EMiGRe*, a system tailored for computing explanations for *missing* recommendations in a graph-based recommendation system. The users can interact with the system through our intuitive visualization interface to navigate the graph, select their missing recommendations, choose their preferred explanation mode (add or remove user actions), and finally consume the explanations in textual or graphical form. n. Throughout, they are guided through important steps of *EMiGRe*³ and useful statistics that clarify the scenario.

Keywords: Explanations · Graph-based Recommendation Systems · Why-Not questions · Explainable AI.

Acknowledgments This work is partially supported by the National French Agency, under the EXPIDA - ANR-22-CE23-0017 project.

1 Introduction

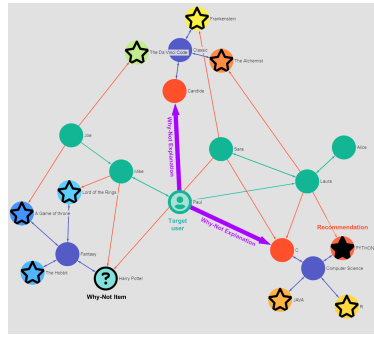
Recommendation Systems (RS) allow users to discover personalized content in vast data collections. Graph recommendation systems modelize entities (users, items, categories, etc) as nodes and relationships (interactions) as edges. They rely on the number and types of connections among the nodes in order to produce the final recommendation list for a user of the system. In this way, users can more effectively explore the data space and find interesting items. Recommendations are frequently coupled with *explanations* - pieces of information that can justify the recommendations. In our graph-based RS, explanations can be modeled as (existing or absent) user-rooted actions (edges) that are responsible for the recommendation. Depending on whether the final recipient is a final user or a system developer, explanations can enhance the trust of the system, or help them debug it in case of non-desired behavior.

In this demonstration, we demonstrate *EMiGRe* [1], a *Why-Not* explanation system for graph-based recommenders. *Why-Not* explanations answer why the system failed to return an expected item in the recommendation list, and are extremely pertinent during the testing phase of the RS, advertisement campaigns, or fairness assessment.

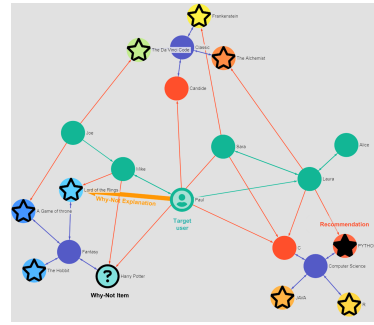
³ The source code for this demonstration is available here https://git.cyu.fr/hattolou/emigre_icwe2024

As a use case, refer to Fig. 1. The example illustrates the case of Paul, who wonders why they are not recommended the book ‘Harry Potter’. *EMiGR*e provides an *actionable Why-Not* explanation in the form of a set of edges (user actions) to be removed (Fig. 1(a)) or to be added (Fig. 1(b)) to the graph. More concretely, Fig. 1a provides the Why-Not explanation composed by two *past actions of the user*, and can be read as “Had you not interacted with ‘Candide’ and ‘C’, your top recommendation would be ‘Harry Potter’.”. Alternatively, Fig. 1b provides the Why-Not explanation composed by one *possible, future action* that the user can perform, and can be read as “Had you interacted with ‘Lord of the Rings’, your top recommendation would be ‘Harry Potter’.”.

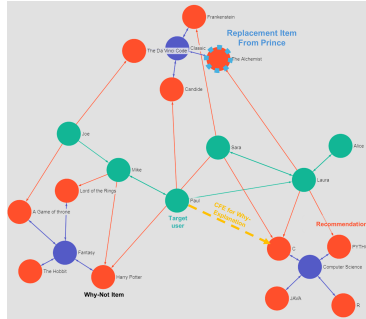
Related work (for instance [2]) has addressed mainly the *Why* Explanation problem. However, as we discuss in [1], Why and Why-Not Explanations are not equivalent. In Fig. 1c we can see that the Why explanation for the top-recommendation (‘Python’) provided by [2] is ‘Had you not interacted with ‘C’, you would have been recommended ‘The Alchemist’’. It is evident that this is not a correct explanation for not obtaining ‘Harry Potter’.



(a) Why-Not explanation (purple edges) in remove mode.



(b) Why-Not explanation (orange edge) in add mode.



(c) Why-Explanation for Python (yellow edge).

Fig. 1: A graph book recommendation system with users (green), books (red), and book categories (purple). Paul is the target user, and nodes with stars denote the recommendation items, with the full star (*Python*) being the top recommendation. Paul wonders “Why-Not *Harry Potter* (node with a question mark)?”.

During this demonstration, the participants will have the opportunity to understand the problem of *Why-Not explanations* and how this differs from the *Why* counterpart. They will be guided through the steps of *EMiGRe*, while interacting with the system through an interactive and intuitive interface. Finally, they will be able to appreciate the nature of the ‘actionable’ explanations, by applying (removing or adding) the proposed actions in the *Why-Not* explanations. The demonstration is built on real-life and synthetic datasets.

2 System Description

EMiGRe is a *Why-Not* explanation framework tailored for graph-based recommenders. It uses the popular Personalized Page Rank algorithm [5] but can be adapted to other user-defined functions that compute the importance of a node for another node monotonically in the number of edges that connect the two nodes. In more detail, *EMiGRe* receives in the input the graph-based RS, a user, and an item that is not the top-1 recommendation for the user, to serve as the *Why-Not* (aka missing) recommendation. *EMiGRe* computes counterfactual-like explanations for the missing recommendation, in two modes. In the *Add (Remove)* mode, it proposes the edges to be added (*removed*) so that the missing recommendation appears at the top of the list. As multiple explanations may exist, *EMiGRe* additionally may operate in a computation-time or explanation-size optimization mode, returning either fast or short explanations. More details can be found in the research paper [1].

Implementation-wise, the back-end implements *EMiGRe* in Python ⁴ and uses Flask for the REST API. The front end is implemented with React and the Sigma.js library to manage and display the graph.

3 Demonstration

Attendees will be exposed to the *Why-Not* explanation problem in recommendations through predefined use cases designed on two real-world and synthetic datasets. They will also have the possibility to freely interact with the platform, defining their own scenarios.

The first use-case is an ‘online’ scenario on a small graph inspired by the Goodreads dataset [4]. The second use-case is an ‘offline’, simulated scenario featuring the Food.com [3] dataset. The selected scenarios showcase the different outcomes of the algorithm in both *Remove* and *Add* modes. We highlight scenarios where (i) the smallest (one edge) explanation is available, (ii) when a more complex one is needed (multiple changes in the graph), and (iii) when a *Why-Not* explanation cannot be found.

⁴ <https://git.cyu.fr/hattolou/why-not-explainable-graph-recommender>

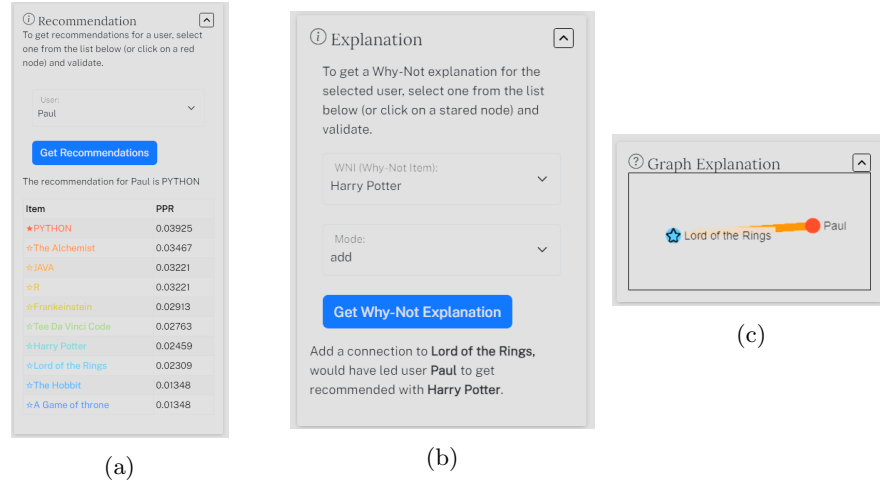


Fig. 2: Different panels of the demonstration interface. (a) Recommendation configuration and results panel. (b) Why-Not configuration and Text Explanation panel. (c) Graph Why-Not Explanation panel.

3.1 Demonstration Scenario Example

First, the attendee is presented with a visualization of the dataset as a labeled graph with edges describing the relationship between items, users, and categories (as Fig.1, ignoring the explanations). The graph is also going to be used in the later steps to display the follow-up recommendations and explanations graphically, but we exclude it in the figures to enhance the readability of the individual panels. In Fig. 2a, the attendee can select a target user via the recommendations panel (or by double-clicking the node in the original graph). Then, the personalized recommendations are displayed on the graph using the same heat-map color palette as on the result table on the recommendations panel (color corresponding to the Personalized Page Rank score (PPR) of the item). In Fig. 2b, the attendee can select the Why-Not item and the explanation mode (Add or Remove) via the explanations panel, where they also receive a text representation of the produced explanation. Finally, Fig. 2c displays the graph visualization of the explanation.

References

1. Attolou, H.M., Tzompanaki, K., Stefanidis, K., Kotzinos, D.: Why-not explainable graph recommender. In: ICDE (2024)
2. Ghazimatin, A., Balalau, O., Roy, R.S., Weikum, G.: PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. WSDM (2020)
3. Li, S.: Food.com recipes and interactions (2019), <https://www.kaggle.com/dsv/783630>
4. Wan, M., Misra, R., Nakashole, N., McAuley, J.J.: Fine-grained spoiler detection from large-scale review corpora. In: ACL (2019)
5. Zhang, H., Lofgren, P., Goel, A.: Approximate Personalized PageRank on Dynamic Graphs. In: SIGKDD. ACM (2016)