Multi-Objective Optimization of Explanation Metrics in Recommender Systems with LLMs

1st André Levi Zanon
Computer Science Department
University College Cork
Cork, Ireland
andre.zanon@insight-centre.org

2nd Leonardo Chaves Dutra da Rocha
Computer Science Department
Universidade Federal de São João del-Rei
São João del Rei, Minas Gerais, Brasil
lcrocha@ufsj.edu.br

3rd Marcelo Garcia Manzato
Inst. De Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos, São Paulo, Brasil
mmanzato@icmc.usp.br

2025 IEEE 37th International Conference on Tools with Artificial Intelligence (ICTAI) | 979-8-3315-4919-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICTAI66417.2025.00027

***Abstract*—Post-hoc Knowledge Graph (KG) explanation algorithms in Recommender Systems (RSs) identify the most relevant paths connecting interacted and recommended items based on shared attributes. These algorithms are categorized into syntactic and semantic approaches; however, neither can simultaneously optimize explanation quality metrics, which measure the recency of interacted items, diversity of attributes, and popularity of attributes shown across explanations. This study explores the use of Large Language Models (LLMs) to jointly optimize these metrics by presenting possible explanation paths for recommended items. The study evaluates three LLMs against both syntactic and semantic algorithms across two datasets and six RSs, finding that LLMs can improve explanation quality.**

***Index Terms*—Recommender Systems, Explainability, Recommendation explanation, Large Language Models**

I. Introduction

Explanations in Recommender Systems (RSs) increase transparency, trust, effectiveness, persuasiveness, efficiency, and satisfaction [1] by providing the reason why a RS generated a suggestion based on users' interactions and their underlying relations with items and metadata [2], [3].

There are two primary approaches to explain recommendations: model-intrinsic (explanations from within the model) and model-agnostic or post-hoc (explanations generated after the recommendation process) [2]. In that regard, Knowledge Graphs (KGs) have gained attention for representing semantic relationships between interacted and recommended items by modeling items and attributes as nodes, and their relations as connecting edges [4].

In KGs, an explanation path connects a recommended item to an interacted item through shared attributes. For example, the explanation "Since you watched Raiders of the Lost Ark, an action movie, you might like The Lord of the Rings: Fellowship of the Ring" uses the genre "action" as the connecting attribute node, between two item nodes.

Post-hoc explainable KG algorithms choose the most relevant paths between interacted and recommended items using either syntactic or semantic methods [5]. Syntactic methods assess attribute co-occurrence between interacted and recommended items relative to their presence in the entire dataset, while semantic methods embed the graph in a latent space to compute relevance scores.

The quality of explanations depends on three main aspects: attribute popularity, attribute diversity among explanations, and the recency of the interacted items on explanation paths [6]. Syntactic and semantic approaches struggle to balance these factors, since they generate explanations for each item sequentially, without considering the set of all possible explanation paths [7].

Large Language Models (LLMs) have the potential to enhance RSs for data augmentation, re-ranking, conversational recommender systems, and explainability [8]–[11]. In this paper, we propose the use LLMs to choose explanation paths for multi-objective optimization of the three explanation quality metrics.

Initially, to verify how the prompt could impact explanation metrics, we investigated the following research question: **(RQ1)**: *How do explanation paths chosen by LLMs compare in explanation quality metrics with syntactic and semantic approaches?* The main objective of this RQ was to compare the explanation quality metrics obtained from the proposed LLM approach with the syntactic and semantic baselines. To answer **(RQ1)**, we evaluated our method and compared it to four explainable algorithms (three syntactic baselines and one semantic baseline) with the offline explanation quality metrics proposed in [6].

Based on these results, a second research question that arises is: **(RQ2)** is: *Can LLM provide a balance between the three explanation quality metrics?* To answer this question, we used the Multi-Attribute Utility Theory (MAUT) from Game Theory to output the option that best balanced all the different explanation quality metrics. Our experiments show that LLMs, by considering all explanation paths for all recommendations, can achieve more balanced values on the explanation quality metrics compared to the baselines.

II. Related Work

A. Post-Hoc Explanations

Model-agnostic KG approaches generate explanations by choosing an attribute that connects interacted and recommended items that form a path. The works of [12]–[14] generated syntactic explanations by ranking attributes using scoring functions.

2375-0197/25/$31.00 ©2025 IEEE
DOI 10.1109/ICTAI66417.2025.00027

In [13], [14] this function considers the number of links between attributes with recommended and interacted items along with the attribute's Inverse Document Frequency (IDF) among all items on the KG. In contrast, [12] calculated a relevance score for each attribute by dividing the number of interacted items containing it by the total number of items with the same attribute. A logarithmic function was applied to penalize rare attributes and rank the most relevant ones. However, in all these studies, the evaluation of explanations was made solely on online experiments, limiting the evaluation to the number of participants.

More recently, [15] developed a post-hoc explainable algorithm using graph embedding algorithms and created two embeddings: one representing the user as the sum of the user's interacted items embeddings and the other representing paths on the KG, as the sum of the path's node and edge embeddings. The explanation chosen by the algorithm is the path with maximum similarity to the user's embedding.

In [15], the works of [12]–[14] were reproduced for offline evaluation using three explanation quality properties defined in [6]. These metrics assess (i) the quality of explanations by measuring the recency of interacted items to recommended items, (ii) the popularity of attributes, and (iii) diversity across multiple explanations. The study found that syntactic approaches often struggle to balance popularity and diversity metrics, while the embedding approach did not perform well on the item recency metric.

B. Large Language Model Explanations

LLMs have demonstrated the potential to enhance various aspects of RSs, including data augmentation, diversification through re-ranking, conversational recommendation and also to generate explanations [8]–[10], [16].

In [17], an LLM was fine-tuned as a surrogate model to replicate RS recommendations, aiming to provide accurate and explainable outputs. Another approach, introduced in [18], jointly optimized both recommendations and user review-based explanations by incorporating an LLM loss term into the Matrix Factorization (MF) training loss. The evaluation made using metrics such as BLEU and ROUGE demonstrated the effectiveness of LLMs for multi-objective optimization.

Despite these advances, current LLM approaches for generating explanations still exhibit limitations. Evaluations often focus primarily on ranking metrics, with limited consideration of explanation quality in user studies [3]. In such cases, explanations are evaluated with anecdotal examples [19]; as a result, there is also a lack of generalization to these approaches.

III. Materials and Methods

A. Problem Formulation and Evaluation Framework

We propose the use of LLMs for multi-objective optimization of explanation metrics. For empirical evaluation, we applied the pipeline shown in Figure 1. We used the MovieLens Latest (ml-latest-small) [20] and LastFM (hetrec2011-lastfm-2k) [21] datasets. Six RSs generated top-5 recommendations for each user, with explanations constructed by extracting a KG from Wikidata[1]. Considering the findings of [19] that explanations are typically assessed using only a few anecdotal examples, we evaluated explanations for all users across both datasets, comparing our LLM-based method with state-of-the-art syntactic and semantic KG post-hoc baselines. Explanation quality was assessed using the metrics proposed in [6].

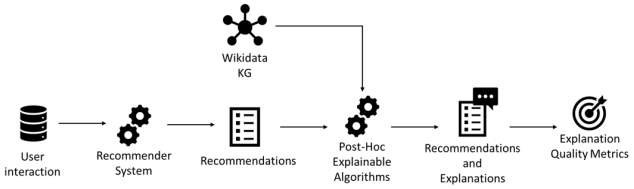

Fig. 1. Schema of the methodology to generate explanations.

KG syntactic and semantic post-hoc explainable algorithms identify the most relevant path in the graph connecting interacted items to recommended items via shared attributes. It can be formally represented as $\forall r \in R \quad argmax(\forall i \in I, \forall p \in P_{i,r} : rel(p))$. Here, I is the set of interacted items, r is a recommended item, $P_{i,r}$ is the set of possible paths, and $rel(p)$ scores the relevance of each path. The path with the highest score is selected for each recommendation.

The main limitation of such approaches is that generating the optimal solution for individual items does not necessarily result in the best overall solution when evaluating the explanations for a set of items. LLMs can generate explanations for an entire set of recommended items. Unlike traditional methods that treat each item separately, LLMs use prompts that incorporate all relevant paths ($P_{i,r}$). Consequently it is formalized as $\forall r \in R, \forall i \in I \quad LLM(prompt(P_{i,r}))$, to collectively determine the most pertinent explanations for all recommendations.

B. KG and Recommendation Generation

Two KGs were extracted from Wikidata for the movie and artist domains and used as input for the syntactic, semantic, and LLM post-hoc explainable algorithms. All descriptive information, statistical data, and identification numbers were removed to obtain data for both datasets, as they did not co-occur in other items. After data extraction, only interactions in the datasets with items containing information in the KG were retained.

For both datasets, the interactions were binarized to represent implicit feedback. Users with five or fewer interactions were also removed because the explanations relied on the metadata information of items. Table I displays the number of entities, triples, and edge types extracted for the KGs of the MovieLens and LastFM datasets, as well as the total number of users, items, and interactions before (in the "Original Dataset" row) and after (in the "Processed Dataset" row) these pre-processing steps. The MovieLens dataset retained 99% of the original interactions, while the LastFM dataset retained 89%.

Inspired by the reproducibility guidelines proposed in [22] and [23], we utilized six different RSs, allocating 90% of

[1]https://www.wikidata.org/

TABLE I
INFORMATION ON DATASETS AND THE WIKIDATA KG EXTRACTED.

		MovieLens		LastFM	
		Original	Processed	Original	Processed
Dataset	users	610	610	1,892	1,875
	items	9,724	9,517	17,632	11,641
	size	100,836	100,521	92,834	83,017
KG	entities	78,703		34,297	
	triples	29,5787		134,197	
	edges	23		33	

the dataset for training and 10% for testing. Every user in the dataset received an explanation for each of the top-5 recommendations. The source code used for all steps shown in Figure 1 is available online[2]. The repository includes the processed datasets and KGs, outputs from the RSs, prompts and explanation paths generated by each explanation algorithm, the trained models, and results of all algorithms.

The algorithms used were: Most Popular (MostPop) [24], Personalized Page Rank algorithm (Page Rank) [14], User K-Nearest Neighbours (UserKNN) [25], Embarrassingly Shallow AutoEncoder (EASE) [26], Bayesian Personalized Ranking Matrix Factorization (BPR-MF) [27], and Neural Collaborative Filtering (NCF) [28]. The authors implemented the EASE, NCF, and PageRank algorithms, while others were implemented using the CaseRecommender library [29]. NCF was developed with a leave-one-out evaluation as in [28].

C. Syntactic Baselines

Syntactic post-hoc KG explanation algorithms use bipartite graphs, where user-interacted items are connected to recommended items via shared attributes. In Figure 2, for instance, blue nodes represent interacted items, yellow nodes recommended items, and green nodes are shared attributes.

The main goal of syntactic approaches is to find the most relevant path connecting interacted and recommended items, where relevance is based on the number of connections an attribute has to the interacted or recommended items compared to its prevalence in the dataset. We used three syntactic baselines.

ExpLOD, as introduced in [13], adapts TF-IDF to rank these connecting attributes. The score for each property p linking a user's interacted items (I_u) to recommended items (I_r) is computed as follows:

$$explod(p, I_u, I_r) = (\alpha \frac{n_{p,I_u}}{|I_u|}) + (\beta \frac{n_{p,I_r}}{|I_r|}) * IDF(p) \quad (1)$$

Here, n_{p,I_u} and n_{p,I_r} represent the frequencies of attribute p in I_u and I_r, respectively. The IDF is defined as $IDF(p) = \log \frac{|N|}{n_{p,N}}$, where N is the set of items in the dataset and $n_{p,N}$ is the number of items in which p appears in N. The values for α and β are 0.5 as in [13].

On example of Figure 2, "Tom Hanks" is the most relevant explanatory attribute linking "Saving Private Ryan" to previously watched movies by ExpLOD because all attribute nodes are connected by two links to interacted items, but the actor node has a higher IDF than the other two attribute nodes.

[2]https://github.com/andlzanon/lod-personalized-recommender

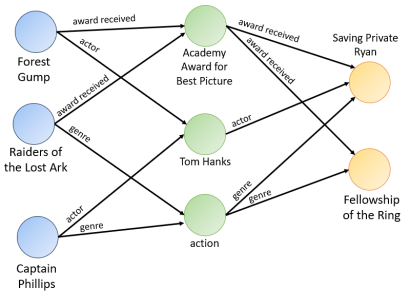

Fig. 2. Example of bipartite graph connecting interacted item nodes (in blue), with recommended item nodes (in yellow), by attribute nodes (in green).

An evolution of ExpLOD, named **ExpLOD v2** [30], expands the attribute consideration by including broader KG properties. For a broader property b, the score is the sum of ExpLOD scores over all its child properties ($P_c(b)$), weighted by the IDF of b:

$$explod(b, I_u, I_r) = \sum_{i=1}^{|P_c(b)|} explod(p_i, I_u, I_r) * IDF(b) \quad (2)$$

The **Property-based Explanation Model (PEM)**, proposed in [12], also uses bipartite property popularity, but instead of link counts, it uses the number of interacted nodes per attribute. It replaces the IDF penalty of ExpLOD with a logarithmic function. It also includes broader properties:

$$score\_pem(p, I_u, I_r) = \frac{|\overline{I(p, I_u)}|/|I_u|}{|\overline{I(p, C)}|/|C|} * \log(|I(p, C)|) \quad (3)$$

Term $|\overline{I(p, I_u)}|$ is the count of items (directly or indirectly) linked to p in I_u and C is the set of all catalog items.

Online experiments [30], [12] have shown that ExpLOD v2 surpasses ExpLOD, and PEM further outperforms ExpLOD v2 regarding user perception.

D. Semantic Baseline

In syntactic algorithms, attributes of a bipartite graph are scored based on their frequency and connections to items. In contrast, semantic post-hoc KG algorithms generate explanations by leveraging latent space representations of nodes and edges generated by a KG embedding model.

The **semantic baseline** employed was proposed in [15] and was not restricted to bipartite graphs. Instead, it creates user and path embeddings to select the most relevant explanation. The process involves: (i) generating KG node and edge embeddings with a KG embedding model (ii) computing a user embedding, (iii) extracting all paths from interacted items to the recommended item, (iv) computing embeddings for each path, (v) measuring similarity between the user and

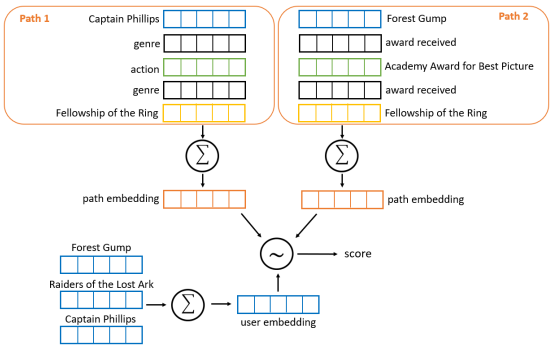

Fig. 3. Representation of the semantic baseline from [15].

path embeddings, and (vi) selecting the path with the highest similarity as the explanation.

This process is illustrated in Figure 3, where blue vector arrays represent interacted items, yellow represents recommended items, green indicates attributes, and black denotes the edges. The symbol $\sum$ represents sum pooling, and $\sim$ indicates cosine similarity.

To generate node and edge embeddings of the Wikidata KG, we employed two translational models (RotatE [31], TransE [32]) and a bilinear model (ComplEX [33]). Dijkstra's algorithm [34] was used to extract all paths of length less than or equal to 5 from interacted to recommended items.

The user embedding is computed using sum pooling over the embeddings of the interacted item nodes (as shown in Equation 4). Each path embedding is obtained by summing the embeddings of all nodes and edges along the path (Equation 5). The selected explanation corresponds to the path whose embedding has the highest similarity to the user embedding (as in Equation 6).

$$user\_embed = \sum_{i \in I_u} embedding(i) \tag{4}$$

$$path\_embed = \forall p \in Path(I_u, r), \sum_{n \in p} embedding(n) \tag{5}$$

$$argmax(\forall pe \in path\_embed, sim(user\_embed, pe)) \tag{6}$$

The term $user\_embed$ represents the user embedding, I_u denotes the set of interacted items, and $embedding(i)$ returns the embedding of node i. The set $Path(I_u, r)$ are paths that connect interacted items to the recommended item r, sim is the cosine similarity function.

TABLE II
KG EMBEDDINGS MODELS PARAMETERS

	MovieLens				LastFM			
	K	λ	B	neg	K	λ	B	neg
TransE	200	0.001	256	No	200	0.001	256	No
RotatE	300	0.001	258	No	300	0.001	258	No
RotatE-o	200	0.001	128	Yes	200	0.001	128	Yes
ComplEX	400	-	128	Yes	200	-	128	No

Parameter tuning of KG embedding models was performed as follows: the learning rate (λ) was chosen from the discrete values 0.1 and 0.001; batch sizes (B) were selected as either 128 or 256; the embedding dimension (K) was set to 200 or 400; and negative sampling was also tested. The training was conducted for 40 epochs. For ComplEX, the AdaGrad optimizer dynamically adjusted λ.

KG embedding models are trained on triple completion, where a triple is (h, r, t), representing a head node h, relation r, and tail node t. Based on h and r, the model predicts tail node t. The data splits for training the embedding models were 0.8/0.1/0.1 for the training, validation, and test sets, respectively. All models were implemented in Pykeen [35] and evaluated using the Hit Rate metric.

Table II presents the optimal parameters for the best-performing models, including the learning rate (λ), batch size (B), embedding dimension (K), and the use of negative sampling (neg). Two RotatE variants are shown: "RotatE-o" refers to a version with fine-tuned parameters for enhanced performance, while the standard "RotatE" follows the original settings from [31] and was trained for 50 epochs.

E. Metrics

Post-hoc explanation algorithms find the most relevant path connecting one or more interacted items to a recommended item. To evaluate such explanation paths, we used three metrics proposed in [6] that identified the main properties of quality explanations to be: (1) connecting recently interacted items to the recommended item, (2) using popular attributes for connecting items, and (3) diversifying paths when explaining multiple recommendations.

Both the Linking Interaction Recency (LIR) and Shared Entity Popularity (SEP) metrics are calculated as the mean of the normalized exponentially weighted moving average for a series of items or attributes within an explanation. The LIR metric evaluates the recency of user-interacted items, while the SEP metric assesses the popularity of attributes used in explanations.

The general form of these metrics is shown below (in Equation 7 for LIR and in Equation 8 for SEP), where β is set to 0.3 as in [6], and values are min-max normalized to range between 0 and 1, with higher values meaning more recently interacted items and popular attributes are displayed in explanations.

For MovieLens, LIR's t^i denotes the interaction timestamps of items, sorted in ascending order, where i is the index of item p. For LastFM, which records user, artist, and play count triples, the LIR metric instead uses t^i as the play count as a recency proxy. For SEP, v^i is the number of item nodes associated with each attribute e^i, also sorted in ascending order; the function is recursive, and the base case is when i is 1. In such cases $LIR(p^1, t^1) = t^1$ and $SEP(e^1, v^1) = v^1$.

$$LIR(p^i, t^i) = (1 - \beta) * LIR(p^{i-1}, t^{i-1}) + \beta * t^i \tag{7}$$

$$SEP(e^i, v^i) = (1 - \beta) * SEP(e^{i-1}, v^{i-1}) + \beta * v^i \tag{8}$$

The Explanation Type Diversity (ETD) metric (Equation 9) measures the variety of attributes within a set of explanations. It is defined as the ratio between the number of distinct attributes in the user's explanation set $|\omega_{L_u}|$ and the minimum of the explanation set length k or the total number of possible attributes $|\omega_L|$. Higher ETD values correspond to more diverse explanations.

$$ETD(S) = \frac{|\omega_{L_u}|}{min(k, |\omega_L|)} \tag{9}$$

We evaluated three syntactic baselines, three semantic baselines (using different graph embedding algorithms), and our proposed LLM approach (using three different LLMs) with respect to LIR, SEP, and ETD metrics. To assess the overall algorithm performance and determine which method best balances these explanation metrics, we applied Multi-Attribute Utility Theory (MAUT), a decision-making framework that ranks options based on a weighted sum of utilities for several criteria. The MAUT score for each algorithm is calculated as $\forall o \in O\ maut\_score(o) = \sum_{c \in C_o} utility(c) * weight_c$.

Here, (O) is the set of candidate algorithms, (C_o) is the set of shared criteria (metrics LIR, SEP, ETD); $weight_c$ is the assigned importance for each metric, and $(utility(c))$ is the normalized utility function value for each criterion. In our evaluation, all criteria were assigned equal weights, and min-max normalization was used as the utility function. The algorithm with highest MAUT is the most balanced across the criteria.

IV. Proposed LLM Approach

Traditional syntactic and semantic post-hoc KG explainable algorithms generate explanations for each recommendation independently, following a greedy strategy in which an explanation is found for each recommended item sequentially. However, such approaches have difficulty optimizing all explanation quality metrics, as syntactic methods struggle with the trade-off between popularity and diversity of attributes, and semantic methods do not prioritize recency during path selection [15].

Our proposed model, illustrated in Figure 4, utilizes LLMs to contextually optimize the explanation selection. The prompt is built based on two parameters: (i) the number of possible paths from interacted items to a recommended item (extracted using the Dijkstra algorithm [34]) and (ii) the number of the users' most recently interacted items. The task of the LLM is to output the most relevant path for each recommendation. Unlike previous methods, this approach allows the LLM to consider the broader context of all paths between interacted and recommended items when selecting explanations.

$$LLM(prompt(I_u, Paths(I_u\, I_r))) \tag{10}$$

Formally, the paths are chosen based on Equation 10, where $Paths(I_u, I_r)$ are paths from the set of interacted items I_u to the recommended items I_r, both provided in the prompt.

Three different LLMs were used: GPT-3.5, GPT-4.0 mini, and LLama 3-70B. We used the OpenAI API (https://openai.com/api/) to run the GPT models and Groq API (https://groq.com/) to run the Llama model. The LLM parameters were chosen empirically. In the ChatGPT models, the temperature was set to 0.2 and nucleus sampling to 1. For LLama, the temperature was set to 0.6, and nucleus sampling was set to 1. The temperature was not set to 0 because our tests showed that it reduced the expressiveness of LLMs, leading to repetitive explanations.

Prompt engineering was adjusted according to the model. GPT models performed best with one-shot prompts, which included an example, whereas LLama 3-70B responded better to zero-shot prompts because it hallucinated when an example was included. All prompts and explanations generated by the LLMs for all users are available in the project source code repository.

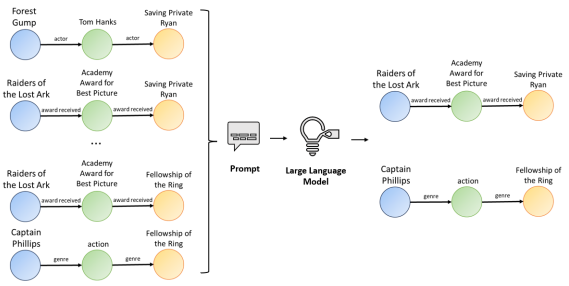

Fig. 4. Schema of the proposed LLM approach.

V. Results

A. Parameter Optimization

To address **RQ1** and compare our LLM-based approach (Section IV) against the baselines in Sections III-C and III-D, we first optimized the two key parameters of the LLM algorithm in Equation 10: (1) the number of the most recently interacted items shown in the prompt, and (2) the number of explanation paths per recommendation included in the input. Five configurations were tested: 50/50, 50/40, 30/30, 50/30, and 30/40, where the first and second numbers correspond to the number of interacted items and explanation paths, respectively.

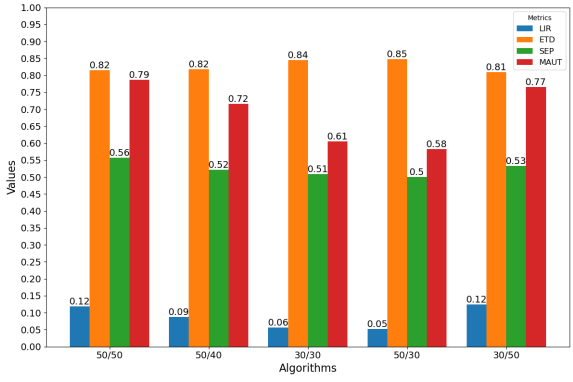

Fig. 5. Parameter optimization result metrics for the EASE top-5 recommendations and the GPT-3.5 LLM on the MovieLens dataset.

TABLE III

EXPLANATION QUALITY METRICS RESULTS FOR THE MOVIELENS DATASET FOR TOP-5 RECOMMENDATIONS. THE HIGHEST VALUES FOR THE SUBSET OF ALGORITHMS WITHIN THE SAME FAMILY ARE IN **BOLD** AND IN *italic* ARE THE HIGHEST VALUES FOR A RS AND A METRIC.

		Syntactic			Semantic				LLM		
		ExpLOD	ExpLOD v2	PEM	TransE	RotatE-o	RotatE	ComplEX	GPT-3.5	GPT-4o mini	Llama 3-70B
Most Popular	LIR	**0.0945 ± 0.15**	0.0834 ± 0.14	0.0320 ± 0.07	**0.0314 ± 0.12**	0.0275 ± 0.11	0.0300 ± 0.11	0.0234 ± 0.09	0.0908 ± 0.17	***0.1321 ± 0.18***	0.1200 ± 0.20
	ETD	0.5809 ± 0.19	0.5947 ± 0.19	***0.9390 ± 0.12***	0.6718 ± 0.21	0.6842 ± 0.20	0.8045 ± 0.32	**0.9537 ± 0.31**	0.8459 ± 0.16	**0.9075 ± 0.12**	0.8767 ± 0.14
	SEP	**0.6322 ± 0.14**	0.6107 ± 0.12	0.1430 ± 0.13	0.5210 ± 0.18	***0.6869 ± 0.13***	0.5392 ± 0.15	0.5625 ± 0.15	0.4638 ± 0.16	0.4479 ± 0.15	**0.4860 ± 0.17**
Page Rank	LIR	**0.0939 ± 0.15**	0.0872 ± 0.15	0.0323 ± 0.08	0.0310 ± 0.11	**0.0312 ± 0.12**	0.0271 ± 0.10	0.0241 ± 0.10	0.0885 ± 0.17	***0.1369 ± 0.18***	0.1151 ± 0.19
	ETD	0.5563 ± 0.20	0.6043 ± 0.19	***0.9389 ± 0.12***	0.7335 ± 0.21	0.6570 ± 0.22	0.8496 ± 0.29	**0.9305 ± 0.31**	0.8348 ± 0.18	**0.9117 ± 0.12**	0.8751 ± 0.14
	SEP	**0.6250 ± 0.16**	0.5606 ± 0.15	0.1095 ± 0.11	0.4662 ± 0.20	***0.7022 ± 0.15***	0.5174 ± 0.15	0.5557 ± 0.16	0.4526 ± 0.17	0.3990 ± 0.16	**0.4969 ± 0.17**
UserKNN	LIR	**0.1010 ± 0.14**	0.0914 ± 0.13	0.0322 ± 0.08	0.0332 ± 0.12	**0.0352 ± 0.12**	0.0290 ± 0.10	0.0234 ± 0.10	0.0829 ± 0.16	***0.1285 ± 0.19***	0.1134 ± 0.19
	ETD	0.6593 ± 0.19	0.6409 ± 0.19	**0.9452 ± 0.11**	0.7055 ± 0.24	0.6849 ± 0.22	0.9721 ± 0.26	***0.9859 ± 0.33***	0.8236 ± 0.19	**0.9000 ± 0.13**	0.8695 ± 0.15
	SEP	**0.5803 ± 0.16**	0.5275 ± 0.14	0.1318 ± 0.12	0.5202 ± 0.23	0.2311 ± 0.15	0.4149 ± 0.14	***0.6103 ± 0.16***	0.3003 ± 0.15	**0.3811 ± 0.15**	0.3220 ± 0.14
BPR-MF	LIR	**0.1048 ± 0.14**	0.0945 ± 0.13	0.0306 ± 0.07	**0.0408 ± 0.16**	0.0298 ± 0.11	0.0311 ± 0.11	0.0244 ± 0.09	0.0780 ± 0.16	***0.1295 ± 0.17***	0.1101 ± 0.18
	ETD	0.6891 ± 0.19	0.6937 ± 0.19	**0.9583 ± 0.10**	0.6826 ± 0.26	0.6934 ± 0.24	0.9852 ± 0.28	***0.9855 ± 0.32***	0.8252 ± 0.18	**0.9055 ± 0.13**	0.8777 ± 0.14
	SEP	**0.6113 ± 0.14**	0.5428 ± 0.14	0.1400 ± 0.12	**0.5755 ± 0.23**	0.2151 ± 0.15	0.4851 ± 0.15	0.5013 ± 0.16	***0.6252 ± 0.18***	0.4437 ± 0.15	0.5546 ± 0.15
EASE	LIR	**0.1000 ± 0.14**	0.0912 ± 0.14	0.0319 ± 0.08	**0.0312 ± 0.12**	0.0295 ± 0.11	0.0268 ± 0.10	0.0209 ± 0.09	0.0875 ± 0.17	***0.1285 ± 0.18***	0.1124 ± 0.18
	ETD	0.6204 ± 0.20	0.6328 ± 0.19	**0.9459 ± 0.11**	0.7345 ± 0.24	0.6751 ± 0.23	0.9627 ± 0.24	***0.9724 ± 0.31***	0.8184 ± 0.18	**0.9003 ± 0.13**	0.8731 ± 0.14
	SEP	**0.5743 ± 0.17**	0.5274 ± 0.15	0.1350 ± 0.13	0.4952 ± 0.49	***0.6583 ± 0.17***	0.3936 ± 0.14	0.6089 ± 0.16	**0.5224 ± 0.18**	0.4256 ± 0.15	0.4623 ± 0.16
NCF	LIR	***0.1181 ± 0.13***	0.1035 ± 0.12	0.0380 ± 0.08	**0.0320 ± 0.13**	0.0244 ± 0.09	0.0285 ± 0.09	0.0199 ± 0.08	0.0557 ± 0.11	**0.1040 ± 0.15**	0.0986 ± 0.16
	ETD	0.8432 ± 0.16	0.8161 ± 0.16	**0.9885 ± 0.05**	0.6966 ± 0.29	0.8080 ± 0.25	***1.1563 ± 0.27***	1.0100 ± 0.32	0.7825 ± 0.19	**0.9191 ± 0.12**	0.8600 ± 0.15
	SEP	**0.5868 ± 0.14**	0.5350 ± 0.13	0.1613 ± 0.11	**0.6122 ± 0.22**	0.2511 ± 0.14	0.3655 ± 0.13	0.3955 ± 0.14	***0.6305 ± 0.18***	0.4003 ± 0.14	0.3381 ± 0.14

TABLE IV

EXPLANATION QUALITY METRICS RESULTS FOR THE LASTFM DATASET FOR TOP-5 RECOMMENDATIONS. THE HIGHEST VALUES FOR THE SUBSET OF ALGORITHMS WITHIN THE SAME FAMILY ARE IN **BOLD** AND IN *italic* ARE THE HIGHEST VALUES FOR A RS AND A METRIC.

		Syntactic			Semantic				LLM		
		ExpLOD	ExpLOD v2	PEM	TransE	RotatE-o	RotatE	ComplEX	GPT-3.5	GPT-4o mini	Llama 3-70B
Mais Popular	LIR	0.0182 ± 0.09	***0.0189 ± 0.11***	0.0143 ± 0.08	0.0104 ± 0.06	0.0100 ± 0.06	0.0092 ± 0.06	**0.0123 ± 0.08**	0.0090 ± 0.05	0.0093 ± 0.06	**0.0096 ± 0.06**
	ETD	0.7023 ± 0.17	0.4927 ± 0.22	**0.9212 ± 0.12**	0.8247 ± 0.27	0.9319 ± 0.25	0.9620 ± 0.36	**1.1394 ± 0.28**	***1.3137 ± 0.31***	1.1946 ± 0.28	1.1244 ± 0.28
	SEP	0.7097 ± 0.17	***0.7537 ± 0.23***	0.1214 ± 0.08	0.5084 ± 0.22	**0.6617 ± 0.21**	0.6377 ± 0.15	0.5181 ± 0.16	0.5638 ± 0.15	0.5577 ± 0.16	**0.6240 ± 0.14**
Page Rank	LIR	0.0191 ± 0.10	***0.0212 ± 0.12***	0.0134 ± 0.07	0.0108 ± 0.07	0.0088 ± 0.06	0.0093 ± 0.06	**0.0125 ± 0.07**	**0.0107 ± 0.06**	0.0085 ± 0.05	0.0095 ± 0.05
	ETD	0.6100 ± 0.20	0.5447 ± 0.19	**0.9440 ± 0.10**	0.7683 ± 0.25	0.8704 ± 0.30	0.8769 ± 0.40	**1.0769 ± 0.32**	1.2293 ± 0.32	***1.1365 ± 0.28***	1.1156 ± 0.28
	SEP	0.6501 ± 0.21	***0.7164 ± 0.21***	0.1209 ± 0.09	0.4820 ± 0.18	0.6359 ± 0.17	**0.6522 ± 0.19**	0.5022 ± 0.18	0.5399 ± 0.16	0.5311 ± 0.17	**0.5956 ± 0.15**
UserKNN	LIR	0.0183 ± 0.10	***0.0191 ± 0.11***	0.0158 ± 0.08	0.0102 ± 0.07	0.0090 ± 0.05	0.0076 ± 0.06	**0.0117 ± 0.07**	0.0081 ± 0.05	**0.0090 ± 0.06**	0.0078 ± 0.05
	ETD	0.5335 ± 0.20	0.5355 ± 0.18	**0.9106 ± 0.14**	0.7760 ± 0.26	0.8688 ± 0.31	0.8453 ± 0.36	**1.0624 ± 0.32**	***1.2107 ± 0.33***	1.1031 ± 0.29	1.1136 ± 0.29
	SEP	**0.5288 ± 0.26**	0.2810 ± 0.22	0.1417 ± 0.10	0.5071 ± 0.19	0.6088 ± 0.18	***0.6464 ± 0.20***	0.4540 ± 0.18	0.4643 ± 0.18	**0.5216 ± 0.17**	0.4603 ± 0.18
BPR-MF	LIR	0.0191 ± 0.10	***0.0204 ± 0.11***	0.0164 ± 0.08	0.0110 ± 0.06	0.0096 ± 0.06	0.0102 ± 0.06	**0.0113 ± 0.07**	0.0090 ± 0.05	**0.0099 ± 0.06**	0.0091 ± 0.06
	ETD	0.6145 ± 0.21	0.6196 ± 0.19	**0.9450 ± 0.11**	0.8403 ± 0.26	0.9219 ± 0.31	0.9148 ± 0.38	**1.1021 ± 0.31**	1.2628 ± 0.34	1.1624 ± 0.30	***1.1927 ± 0.31***
	SEP	0.5605 ± 0.23	**0.6302 ± 0.19**	0.1759 ± 0.12	0.5312 ± 0.18	0.6002 ± 0.17	***0.6332 ± 0.18***	0.5984 ± 0.18	0.5139 ± 0.17	**0.5411 ± 0.16**	0.5159 ± 0.17
EASE	LIR	0.0188 ± 0.11	***0.0194 ± 0.11***	0.0154 ± 0.08	0.0102 ± 0.07	0.0092 ± 0.05	0.0073 ± 0.05	**0.0111 ± 0.07**	**0.0100 ± 0.06**	0.0087 ± 0.05	0.0082 ± 0.05
	ETD	0.5474 ± 0.20	0.5585 ± 0.18	**0.9246 ± 0.13**	0.7934 ± 0.25	0.8753 ± 0.32	0.8581 ± 0.37	**1.0707 ± 0.32**	1.2220 ± 0.33	1.1123 ± 0.30	***1.1426 ± 0.29***
	SEP	**0.5307 ± 0.25**	0.2861 ± 0.21	0.1466 ± 0.10	0.5026 ± 0.19	0.5870 ± 0.19	***0.6182 ± 0.21***	0.4639 ± 0.18	0.4985 ± 0.18	**0.5574 ± 0.16**	0.4776 ± 0.17
NCF	LIR	***0.0182 ± 0.09***	0.0162 ± 0.08	0.0162 ± 0.08	0.0098 ± 0.06	0.0106 ± 0.06	0.0093 ± 0.06	**0.0131 ± 0.07**	**0.0124 ± 0.06**	0.0115 ± 0.06	0.1020 ± 0.07
	ETD	0.7775 ± 0.18	0.7867 ± 0.18	**0.9589 ± 0.09**	0.9409 ± 0.27	1.0318 ± 0.32	1.0776 ± 0.35	**1.2057 ± 0.29**	1.3720 ± 0.34	1.3064 ± 0.31	***1.3383 ± 0.33***
	SEP	**0.5904 ± 0.19**	0.5514 ± 0.20	0.2748 ± 0.15	0.6302 ± 0.17	***0.6509 ± 0.16***	0.6185 ± 0.17	0.6153 ± 0.17	0.5627 ± 0.15	**0.5786 ± 0.15**	0.5024 ± 0.16

As shown in Figure 5, we evaluated these settings using the EASE recommendation algorithm (chosen for its top accuracy performance compared to the other six RSs applied) and the GPT-3.5 LLM, that provides a conservative lower bound for LLM performance, since it is the oldest LLM model used.

The results revealed that reducing the number of input paths in the prompt, especially in the 30/30 and 50/30 configurations, resulted in lower LIR metric values. Providing more explanation paths (as in 30/50) improves LIR and keeps ETD competitive, even though SEP decreases slightly. All algorithms obtained similar ETD results, demonstrating that LLMs consistently generate diverse explanations.

We chose the parameter configuration with 50 interacted items and 40 explanation paths because it achieved one of the highest MAUT values for the LLM while keeping the number of paths and interacted items reasonable. This configuration represents a balanced trade-off across metrics and corresponds to the elbow point in the MAUT curve, beyond which further increases in parameters lead to diminishing returns.

B. Explanation Quality Metrics

Tables III and IV report LIR, SEP, and ETD values for the syntactic, semantic, and LLM-based explanation approaches across six RSs (Section III-B) on the MovieLens and LastFM datasets. The columns represent the explanation algorithms, and the rows present the metric values for each RS. The tables are organized into three sections: syntactic (ExpLOD, ExpLOD v2, and PEM), semantic (TransE, RotatE-o, RotatE, and ComplEX), and LLM (GPT-3.5, GPT-4.0 mini, LLama 3-70B). Metric values in **bold** are the highest within each section, while *italic* values are the highest across all algorithms.

Syntactic approaches. In the MovieLens dataset (Table III), ExpLOD and ExpLOD v2 produced similar results, with ExpLOD achieving the highest LIR and SEP among the syntactic algorithms, whereas ExpLOD v2 obtained higher path diversity (ETD). However, PEM achieved the overall highest ETD among the syntactic methods at the expense of a lower SEP. On LastFM (Table IV), ExpLOD v2 outperformed ExpLOD in terms of LIR and SEP, reversing the trend observed in MovieLens. PEM continued to dominate in ETD across all RSs, confirming the popularity-diversity trade-off noted in [7].

Semantic approaches. In MovieLens, TransE achieved the best LIR scores for the Most Popular, BPR-MF, EASE, and NCF. For ETD, ComplEX outperformed the other embedding models in terms of diversity in five of six RSs. The SEP results were mixed, with no single model consistently leading. On

TABLE V
MAUT VALUES FOR EXPLANATION QUALITY METRICS RESULTS ON THE MOVIELENS DATASET. THE HIGHEST VALUE FOR EACH RECOMMENDATION ALGORITHM ACROSS ALL METHODS IS IN **BOLD**.

	Syntactic			Semantic				LLM		
	ExpLOD	**ExpLOD v2**	**PEM**	**TransE**	**Rotate-o**	**Rotate**	**ComplEX**	**GPT-3.5**	**GPT-4o mini**	**Llama 3-70B**
Most Popular	0.5180	0.4831	0.3464	0.3375	0.4382	0.4628	0.5904	0.6401	**0.8121**	0.7707
Page Rank	0.4960	0.4819	0.3576	0.3753	0.4418	0.4939	0.5769	0.6261	**0.8058**	0.7645
UserKNN	0.5757	0.4908	0.3220	0.3641	0.1489	0.5348	0.6666	0.4821	**0.7573**	0.6379
BPR-MF	0.6031	0.5263	0.3244	0.3546	0.0818	0.5928	0.5815	0.6716	0.7872	**0.7895**
EASE	0.5249	0.4793	0.3422	0.3691	0.4116	0.5069	0.6351	0.6406	**0.7834**	0.7311
NCF	**0.7419**	0.6356	0.2731	0.3613	0.1599	0.5074	0.3936	0.5169	0.6163	0.5111

TABLE VI
MAUT VALUES FOR EXPLANATION QUALITY METRICS RESULTS ON THE LASTFM DATASET. THE HIGHEST VALUE FOR EACH RECOMMENDATION ALGORITHM ACROSS ALL METHODS IS IN **BOLD**.

	Syntactic			Semantic				LLM		
	ExpLOD	**ExpLOD v2**	**PEM**	**TransE**	**Rotate-o**	**Rotate**	**ComplEX**	**GPT-3.5**	**GPT-4o mini**	**Llama 3-70B**
Most Popular	**0.7034**	0.6667	0.3526	0.3874	0.4992	0.4699	0.5832	0.5666	0.5258	0.5919
Page Rank	0.6059	**0.6667**	0.3227	0.3736	0.4554	0.4804	0.5791	0.6276	0.5178	0.5902
UserKNN	0.5658	0.4263	0.4236	0.4395	0.5168	0.4868	**0.5887**	0.5619	0.5730	0.5150
BPR-MF	0.5765	**0.6671**	0.3871	0.4332	0.4846	0.5207	0.6262	0.5797	0.5746	0.5483
EASE	0.5889	0.4374	0.4104	0.4567	0.5270	0.4868	0.5887	**0.6581**	0.6113	0.5564
NCF	0.6130	0.5081	0.3623	0.4271	0.5270	0.4729	0.6848	**0.7048**	0.6495	0.6183

LastFM, semantic methods showed stable trends: ComplEX delivered the best ETD and achieved the highest LIR scores. However, SEP was highest for the RotatE variants. These results suggest that semantic models offer balanced trade-offs between diversity (ETD) and popularity (SEP), though with lower recency (LIR) compared to syntactic models.

LLM approaches. On MovieLens, GPT-4.0 mini achieved the highest LIR and SEP across all RSs, with competitive ETD performance, approaching that of PEM and ComplEX models. On LastFM, the LLM performance varied more by RS and metric. However, LLMs dominated syntactic and semantic methods on the ETD metric overall: one of GPT-3.5, GPT-4.0 mini, and LLama 3-70B achieved the top ETD scores for all RS. This can be attributed to the use of longer and more complex paths that differ from the bipartite graphs used in the baselines. These longer paths improved diversity but reduced attribute popularity (SEP), mirroring the trade-off seen in syntactic and semantic approaches.

Addressing **RQ1**, the LLMs demonstrated strong performance across datasets and metrics. They consistently achieved the highest LIR values and the best ETD scores, offering competitive and often superior explanation quality when compared to traditional syntactic and semantic methods.

C. Multi-Objective Optimization Analysis

Tables III and IV report the individual metric values, but the trade-offs between the metrics make direct comparisons difficult. To address **RQ2**, we applied the MAUT method to rank the explanation algorithms for each RS and dataset using LIR, SEP, and ETD as criteria. The results are shown in Tables V and VI. **Bold** values highlight the best-performing explanation algorithm per RS.

In the MovieLens dataset (Table V), GPT-4.0 mini achieved the highest MAUT scores for four RSs: Most Popular, Page Rank, UserKNN, and EASE. LLama 3-70B led for BPR-MF, and ExpLOD was best for NCF. These results show that LLMs, especially GPT-4.0 mini, offered the most balanced trade-off between popularity, diversity, and recency metrics, outperforming approaches in five out of six RSs.

In contrast, Table VI reveals a greater variability in the LastFM dataset. Syntactic methods (ExpLOD and ExpLOD v2) led to three RSs, ComplEX was best for UserKNN, and GPT-3.5 excelled in EASE and NCF. This confirms the sensitivity of the explanation quality to the underlying RS.

Therefore, to answer **RQ2**, LLMs demonstrated some ability to balance the three explanation quality metrics. Across both datasets, a generative model outperformed baselines in 7 out of 12 RS–dataset combinations, highlighting the effectiveness of LLMs in optimizing multiple explanation criteria simultaneously.

VI. CONCLUSIONS

Our study finds that while LLM-based approaches can outperform conventional syntactic and semantic methods in terms of explanation quality for post-hoc KG explanations in RSs, they come with computational challenges. Syntactic strategies remain lightweight, and semantic strategies require graph embedding model training, but LLMs are far more resource-intensive. We also observed that the relationship between the explanation path length and explanation metrics is complex because longer explanation paths on the LastFM dataset led to higher attribute diversity but lower attribute popularity. Future work should focus on fine-tuning LLMs with respect to explanation quality metrics and conducting user studies to assess improvement in user experience.

ACKNOWLEDGMENT

This publication has emanated from research conducted with the financial support of Research Ireland under Grant

number 12-RC-2289-P2, which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission. The authors also acknowledge CAPES, CNPq, Fapesp, Fapemig, INCT-TILD-IAR for their funding and support of this research.

REFERENCES

[1] N. Tintarev and J. Masthoff, "Explaining recommendations: Design and evaluation," *Recommender systems handbook*, pp. 353–382, 2015.

[2] Y. Zhang, X. Chen *et al.*, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends® in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020.

[3] I. Nunes and D. Jannach, "A systematic review and taxonomy of explanations in decision support and recommender systems," *User Modeling and User-Adapted Interaction*, vol. 27, pp. 393–444, 2017.

[4] Q. Zhang, Z. Xu, H. Liu, and Y. Tang, "Kgat-sr: Knowledge-enhanced graph attention network for session-based recommendation," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2021, pp. 1026–1033.

[5] N. Tiwary, S. A. M. Noah, F. Fauzi, and T. S. Yee, "A review of explainable recommender systems utilizing knowledge graphs and reinforcement learning," *IEEE Access*, 2024.

[6] G. Balloccu, L. Boratto, G. Fenu, and M. Marras, "Post processing recommender systems with knowledge graphs for recency, popularity, and diversity of explanations," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 646–656.

[7] A. L. Zanon, L. C. D. da Rocha, and M. G. Manzato, "Balancing the trade-off between accuracy and diversity in recommender systems with personalized explanations based on linked open data," *Knowledge-Based Systems*, vol. 252, p. 109333, 2022.

[8] W. Wei, X. Ren, J. Tang, Q. Wang, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang, "Llmrec: Large language models with graph augmentation for recommendation," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 806–815.

[9] D. Carraro and D. Bridge, "Enhancing recommendation diversity by re-ranking with large language models," *arXiv preprint arXiv:2401.11506*, 2024.

[10] D. Yang, F. Chen, and H. Fang, "Behavior alignment: A new perspective of evaluating llm-based conversational recommendation systems," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 2286–2290.

[11] S. Lubos, T. N. T. Tran, A. Felfernig, S. Polat Erdeniz, and V.-M. Le, "Llm-generated explanations for recommender systems," in *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, 2024, pp. 276–285.

[12] Y. Du, S. Ranwez, N. Sutton-Charani, and V. Ranwez, "Post-hoc recommendation explanations through an efficient exploitation of the dbpedia category hierarchy," *Knowledge-Based Systems*, vol. 245, p. 108560, 2022.

[13] C. Musto, F. Narducci, P. Lops, M. De Gemmis, and G. Semeraro, "Explod: a framework for explaining recommendations based on the linked open data cloud," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 151–154.

[14] C. Musto, G. Rossiello, M. de Gemmis, P. Lops, and G. Semeraro, "Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations," in *Proceedings of the 13th ACM conference on recommender systems*, 2019, pp. 383–387.

[15] A. L. Zanon, L. C. D. da Rocha, and M. G. Manzato, "Model-agnostic knowledge graph embedding explanations for recommender systems," in *World Conference on Explainable Artificial Intelligence*. Springer, 2024, pp. 3–27.

[16] J. Chen, Z. Liu, X. Huang, C. Wu, Q. Liu, G. Jiang, Y. Pu, Y. Lei, X. Chen, X. Wang *et al.*, "When large language models meet personalization: Perspectives of challenges and opportunities," *World Wide Web*, vol. 27, no. 4, p. 42, 2024.

[17] Y. Lei, J. Lian, J. Yao, X. Huang, D. Lian, and X. Xie, "Recexplainer: Aligning large language models for explaining recommendation models," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 1530–1541.

[18] Y. Peng, H. Chen, C.-S. Lin, G. Huang, J. Hu, H. Guo, B. Kong, S. Hu, X. Wu, and X. Wang, "Uncertainty-aware explainable recommendation with large language models," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–8.

[19] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert, "From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai," *ACM Comput. Surv.*, vol. 55, no. 13s, Jul. 2023.

[20] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[21] I. Cantador, P. Brusilovsky, and T. Kuflik, "2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011)," in *Proceedings of the 5th ACM conference on Recommender systems*, ser. RecSys 2011. New York, NY, USA: ACM, 2011.

[22] M. Ferrari Dacrema, S. Boglio, P. Cremonesi, and D. Jannach, "A troubling analysis of reproducibility and progress in recommender systems research," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 2, pp. 1–49, 2021.

[23] D. Tchuente, J. Lonlac, and B. Kamsu-Foguem, "A methodological and theoretical framework for implementing explainable artificial intelligence (xai) in business applications," *Computers in Industry*, vol. 155, p. 104044, 2024.

[24] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, ser. RecSys '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 39–46.

[25] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '94. New York, NY, USA: Association for Computing Machinery, 1994, p. 175–186.

[26] H. Steck, "Embarrassingly shallow autoencoders for sparse data," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 3251–3257.

[27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI '09. Arlington, Virginia, USA: AUAI Press, 2009, p. 452–461.

[28] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 173–182.

[29] A. da Costa, E. Fressato, F. Neto, M. Manzato, and R. Campello, "Case recommender: a flexible and extensible python framework for recommender systems," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 494–495.

[30] C. Musto, F. Narducci, P. Lops, M. de Gemmis, and G. Semeraro, "Linked open data-based explanations for transparent recommender systems," *International Journal of Human-Computer Studies*, vol. 121, pp. 93–107, 2019.

[31] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[32] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[33] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[34] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[35] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, and J. Lehmann, "PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings," *Journal of Machine Learning Research*, vol. 22, no. 82, pp. 1–6, 2021.