CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Do calibrated recommendations affect explanations? A study on post-hoc adjustments
Authors	Atauchi, Paul Dany Flores;Zanon, André Levi;da Rocha, Leonardo Chaves Dutra;Manzato, Marcelo Garcia
Publication date	2025
Original Citation	Atauchi, P. D. F., Zanon, A. L., da Rocha, L. C. D. and Manzato, M. G. (2025) 'Do calibrated recommendations affect explanations? A study on post-hoc adjustments', Journal on Interactive Systems, 16(1), pp.441-460. https://doi.org/10.5753/jis.2025.5563
Type of publication	Article (peer reviewed)
Link to publisher's version	10.5753/jis.2025.5563
Rights	© 2025, Brazilian Computing Society. This work is licensed under a Creative Commons Attribution 4.0 International License. - https://creativecommons.org/licenses/by/4.0/
Download date	2026-11-02 22:16:32
Item downloaded from	https://hdl.handle.net/10468/17725

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Journal on Interactive Systems, 2025, 16:1, doi: 10.5753/jis.2025.5563
This work is licensed under a Creative Commons Attribution 4.0 International License.

Do Calibrated Recommendations Affect Explanations? A Study on Post-Hoc Adjustments

Paul Dany Flores Atauchi [**Universidade de São Paulo** | ***paul.atauchi@usp.br***]
André Levi Zanon [**University College Cork** | ***andre.zanon@insight-centre.org***]
Leonardo Chaves Dutra da Rocha [**Universidade Federal de São João del-Rei** | ***lcrocha@ufsj.edu.br***]
Marcelo Garcia Manzato [**Universidade de São Paulo** | ***mmanzato@icmc.usp.br***]

Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Avenida Trabalhador São-carlense, 400, Centro, São Carlos, SP, 13566-590, Brazil.

Received: 15 February 2025 • **Accepted:** 23 June 2025 • **Published:** 28 June 2025

Abstract: Recommender systems generate suggestions by identifying relationships among past interactions, user similarities, and item metadata. Recently, there has been an increased focus on evaluating recommendations based not only on accuracy but also on aspects like transparency and calibration. Transparency is important, as explanations can enhance user trust and persuasion, while calibration aligns users' interests with recommendation lists, improving fairness and reducing popularity bias. Traditionally, calibration and explanation are applied in post-processing. Our study investigates two key research gaps: (1) the impact of graph embeddings in model-agnostic knowledge graph explanations, exploring their under-researched potential compared to syntactic approaches to produce meaningful explanations; and (2) the effect of calibration on recommendation explanations, assessing whether calibrated recommendation reordering influences the outcomes of explanation algorithms. We evaluate the quality of explanations using a set of metrics, such as diversity, which measures how well different interests of the user are covered; popularity, which assesses how well explanations avoid favoring already popular items; and recency, which examines the inclusion of recently interacted items. Our findings demonstrate that graph embedding methods are effective in generating high-quality explanations using these offline explanation metrics, and that post-hoc knowledge graph explanation algorithms are robust to calibration changes.

Keywords: Recommender Systems, Calibration, Explanation, Graph Embeddings

1 Introduction

Recommender systems generate suggestions by analyzing user interactions and/or incorporating side information, such as domain knowledge, contextual data, and item metadata, to enhance accuracy [Aggarwal, 2016; Ricci *et al*., 2022]. However, beyond improving accuracy, aspects such as fairness, diversity, serendipity, novelty, and coverage also play a crucial role in enhancing user experience. These aspects are particularly relevant in addressing the long-tail problem in recommender systems, where a small number of items are highly popular across users while most items receive few interactions [Kaminskas and Bridge, 2016]. In addition, to capture nonlinear relationships among these data sources, recommendation algorithms have become more complex and are often considered black boxes, meaning that users cannot understand why a particular suggestion was provided by the system [Tintarev and Masthoff, 2015].

To address the problem of fairness in recommender systems, Steck [2018] introduced the concept of calibration in which a recommendation list is considered calibrated when the user's interacted item list and the user's recommended item list have the same distribution of classes considering an attribute. Considering an example in a movie domain, if a user interacted with 30% of comedy movies, 10% of adventure movies, and 60% of action movies, it is expected that the user would receive a similar proportion of recommended items.

Since calibration ensures that recommendations reflect the user's preferences in a balanced way, it can also influence how users perceive and understand explanations in recommender systems.

In particular, generating explanations for recommendations has gained significant attention in the literature [Balog and Radlinski, 2020; Rana *et al*., 2022], as explanations can enhance transparency, trust, efficacy, persuasion, and user satisfaction [Tintarev and Masthoff, 2015]. Despite the efforts to explain and generate fair recommendations intrinsically in the recommendation algorithms, both methods are mainly approached as post-processing methods, which means that one can impact the other. However, no existing work has systematically analyzed whether and how calibration influences the explanations generated for recommendations. Consequently, the main objective of this research is:

Objective: To investigate the effects of calibrated recommendations in explanations algorithms for recommender systems.

To understand how calibration impacts explanations in recommender systems, it is important to include in our analysis a comparison among the main approaches for generating explanations available in the literature. Currently, strategies for generating explanations for RSs can be broadly categorized into two approaches: (1) model-intrinsic and (2) model-agnostic, also known as post-hoc approaches [Zhang and Chen, 2020; Rudin, 2019]. Intrinsic models aim to generate explanations along with the recommendation itself,

presenting the reasons why an interacted item is related to the recommended one [Xu *et al*., 2023]. Model-agnostic or post-hoc methods, on the other hand, use an algorithm independent of the recommendation process to establish relationships between interacted and recommended items and, therefore, do not rely on the recommendation algorithm [Rana *et al*., 2022]. Model-agnostic methods are generally enriched by external sources, such as Knowledge Graphs (KGs) – where nodes represent items and their related attributes, and edges represent the semantic relationships between nodes – or user items' reviews [Cao *et al*., 2024].

Post-hoc KG algorithms predominantly rely on syntactic approaches, where the relevance of paths connecting interacted and recommended items is measured by the number of associated links. However, this measure may not fully capture the semantic relationships within the data. In contrast, graph embedding-based approaches [Balloccu *et al*., 2022; Li and Yang, 2022] can generate semantic representations of paths between nodes of recommended and interacted items by projecting them into a vector space. Despite the potential impact that the choice of the graph embedding algorithm may have on the quality of recommendation explanations [Peng *et al*., 2023], we found no studies in the literature that evaluate this aspect.

Thus, we aimed to fill this gap in the literature by comparing the impact of different graph embedding algorithms on the explanation quality in post-hoc explanation algorithms for recommender systems, as well as by comparing them with syntactic approaches. Therefore, the specific objectives were divided into two Research Questions (RQ). The first is:

RQ1: What is the impact of different types of graph embedding algorithms on the quality of explanations generated for recommender systems?

To answer this research question, we implemented a model-agnostic (post-hoc) explanation algorithm using three graph embedding models: one bilinear and two translational. These vector representations of graph nodes and edges were generated using three state-of-the-art strategies (i.e., TransE [Lin *et al*., 2015] and RotatE [Sun *et al*., 2019] as translational models, and ComplEx [Trouillon *et al*., 2016] as a bilinear model). The embeddings obtained from these algorithms were combined to generate representations of both the user and the graph paths connecting interacted and recommended items. The path whose latent space representation is most similar to the user representation is selected as the explanation.

In the experimental analyses, we ran the post hoc explanations on six different collaborative filtering recommender systems. To validate explanations, we used explanation metrics proposed by Balloccu *et al*. [2022] that assess their quality, which is characterized by three aspects: measuring the recency of interacted items, measuring the popularity of attributes, and also measuring the diversity of the attributes connecting interacted items to the recommended item.

Additionally, we compared graph embedding strategies with three syntactic approaches (ExpLOD [Musto *et al*., 2016], ExpLOD version 2 (ExpLOD v2) [Musto *et al*., 2019], and the Proposed Property-based Explanation Model (PEM) [Du *et al*., 2022]). The goal of this analysis is to answer our second research question:

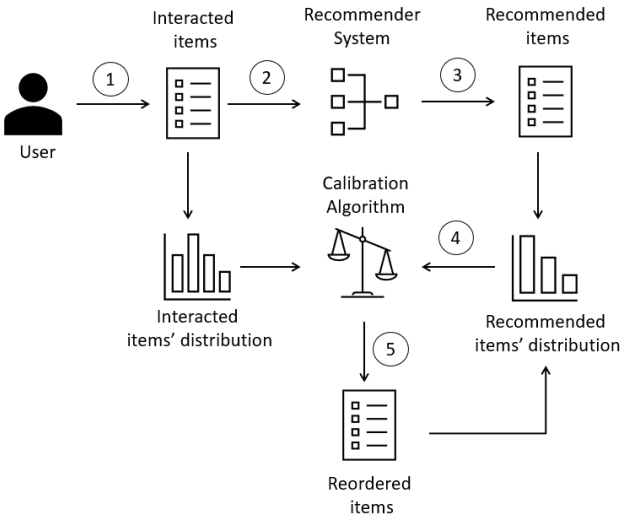


Figure 1. Post processing calibration of recommendations.

RQ2: Are graph embedding-based strategies actually capable of generating better explanations in recommender systems compared to syntactic approaches?

Similarly to explanation in recommender systems, calibration can be performed at different stages. The pre-processing step involves adjusting or omitting parts of the original dataset [da Silva and Durão, 2023]. During recommendation, calibration can be achieved by modifying the loss function or objective function of the recommendation algorithm [Souza and Manzato, 2024a]. Finally, as a post-processing step, it consists of reordering the recommended items to better align with the user's item attribute distribution [Steck, 2018; da Silva and Durão, 2025].

Figure 1 illustrate the calibration process, where, on step 1 a user interacts with items that are then inputted on step 2 to a recommender system. The user-interacted items follow a distribution based on a class of items (i.e., genres of music and movies). Then, step 3 represents the generation of recommendations by the recommender. This list also has a distribution of classes. The recommended and interacted item lists are then inputted in step 4 to a calibration algorithm that will be responsible for remembering the recommendations in step 5 in order to align the recommendation class distribution to the interacted item distribution. Once this alignment achieves its maximum similarity between distributions, the process is finalized.

In this context, while calibration ensures a balance between the user's preferences and recommendation distributions, and explainability fosters transparency and trust in recommender systems, both methods are commonly used in post-processing steps. Since most recommender systems do not account for these factors, the stacking of post-hoc methods could facilitate the integration of explainability and calibration algorithms. Consequently, our Research Question 3 (RQ3) is:

RQ3: What is the impact of calibration strategies on the explanations of recommendation systems?

To answer this research question, we implemented the calibration strategy proposed by Steck [2018] to reorder the recommendations of five recommender systems using knowledge graph (KG) attributes, based on the same six collaborative filtering algorithms used to address RQ1 and RQ2. Then, we applied three syntactic post-hoc explainable algorithms and the proposed semantic algorithm. We compared the explanation quality metrics with and without calibration to analyze whether reordering the items based on the user profile affected the explanations.

Therefore, the main contributions of this work are:

- The development of a comprehensive approach for evaluating explanation generation strategies for recommender systems based on knowledge graphs.
- A comparative analysis between graph embedding strategies and syntactic algorithms in terms of explanation quality in recommender systems.
- A comparative analysis of the impact of different graph embedding strategies on the quality of explanations generated for recommender systems;
- A comparative analysis of the effect of calibration on the different post-hoc explainable algorithms.

Focusing first on RQ1, our results clearly show that bilinear models, which can represent more complex relationships between nodes and edges, positively impact explanation quality metrics. Regarding RQ2, we observed that while syntactic methods prioritize the recency of items and the popularity of attributes chosen in explanations, embedding-based strategies balance the trade-off between attribute popularity and diversity in the explanations shown to users. Finally, in regard to RQ3, we verified that post hoc explanation algorithms are robust to calibration methods, meaning that calibrating recommendations had a low impact on the results of RQ1 and RQ2 and that post hoc explainable algorithms can be used stacked to calibration strategies.

The paper is structured as follows: Section 2 reviews related work on explainable, knowledge graph-based recommender system algorithms approaches and calibration strategies. Section 3 details the evaluation approach, focusing on reproducibility, and Section 4 details post-processing calibration and explanation algorithms, including their experimental setup and metrics. Section 5 discusses the results and answers to the three research questions leveraged in this section. Section 6 summarizes the findings.

2 Background and Related Work

2.1 Post Hoc KG Explanation Algorithms

The concept of Knowledge Graphs is defined by Paulheim [2016] with four main characteristics: (1) describe real-world entities and their relations in a graph structure; (2) define classes and relations of entities (3) allows potential interrelating entities and (4) cover different domains. More formally, we define KG as:

Definition: $KG = (V, E)$, where V is the set of nodes that represents real-world entities and E are edges that create triple facts. Triples, denoted by (v_h, e, v_t) where v_h and v_t are nodes in V and e is a directed edge contained in E connecting the two entities of the real-world nodes [Guo *et al.*, 2022].

This definition provides the structural basis for generating explanations by the models. In particular, it supports the construction of explanation path that connect user-interacted items to recommended items through shared attributes (see Section 4.1).

Considering a KG of places, a city such as Paris could be represented by a node connected to its tourist attraction, the Tower Eiffel, which would be another node in the KG. An edge 'has attraction' that represents the semantic relation between both real-world entities connects them, creating a triple $(Paris, has_attraction, Tower_Eiffel)$ that represents a connection between two nodes in a graph. Particularly, in recommender systems, an explanation represents a path between one or more interacted items and a recommended item.

Since KGs provide structured metadata on items, they have been used to generate accurate and explainable recommendations in various recommendation architectures. Explanations in KGs are generated by associating interacted and recommended items through shared attributes. In the literature, there are two types of post-hoc or model-agnostic explanatory algorithms using KGs: one in which recommendations are reordered based on the best explanations for a recommended item and another in which only the explanations are generated [Rana *et al*., 2022].

Considering model-agnostic reordering algorithms using KGs, Balloccu *et al*. [2022] employed three optimization metrics—recency of interacted items, popularity, and diversity of attributes extracted from KG explanation paths—to reorder recommendations. Meanwhile, Zanon *et al*. [2022] reordered recommendations by evaluating the relevance of attributes extracted from explanation paths, comparing the frequency of attribute associations with interacted items and the item catalog. Additionally, Hada *et al*. [2021] generated explanations through aspect extraction and sentiment analysis, enhancing recommendation accuracy by incorporating textual reviews as a regularizer for the recommendation algorithm. However, in these studies, the proposed approaches evaluated explanations exclusively based on metrics such as accuracy and diversity.

Regarding post-hoc KG architectures for generating explanations, Musto *et al*. [2016] introduced an algorithm called ExpLOD, which leverages KG explanations based on a bipartite graph that connects interacted and recommended items through shared attributes. Explanations are ranked using an adaptation of the Term Frequency-Inverse Document Frequency (TF-IDF) metric, where item nodes are documents, and attribute nodes are terms. This work was evaluated by comparing the proposed explanation with information extracted from the KG in an online experiment, where the method improved user perception regarding explanation objectives. Musto *et al*. [2019] extended ExpLOD Musto *et al*. [2016] by incorporating broader attributes

from the KG hierarchy. The same online experiment was conducted, but ExpLOD explanations were compared with the newly proposed version. Users preferred explanations with broader attributes when analyzed from the perspective of explanation objectives. More recently, Du *et al*. [2022] proposed the Property-based Explanation Model (PEM), a scoring function that ranks attributes based on their connections with interacted items and the complete item catalog. PEM outperformed the second version of ExpLOD in online experiments, establishing itself as the state-of-the-art for model-agnostic KG-based explanatory algorithms. However, these approaches are syntactic and do not inherently consider the KG structure and path to generate explanations. To address this gap, Zanon *et al*. [2024] introduced a model-agnostic explanation algorithm using KG vector representations, comparing it with syntactic approaches through offline explanation quality metrics. However, different ways of generating graph vector representations were not explored, as only a single model was used to generate the KG embeddings in the explanation algorithm.

The evaluation of explanations has also gained attention, as explanations are primarily assessed through online user studies, which are time-consuming and costly to validate strategies. In Coba *et al*. [2022], offline metrics were implemented to evaluate explainable recommendations, assessing the robustness of explanatory algorithms by measuring the number of items that can be explained to users and the number of user interactions related to explanations. On the other hand, some studies also consider the diversity and relevance of attributes displayed in explanations [Balloccu *et al*., 2022; Souza and Manzato, 2022], although the relationship between such metrics and online tests remains undefined. Moreover, offline metrics are not standardized, and studies evaluating explanations through user studies often lack quantitative assessment.

Thus, explainable recommender systems often do not evaluate explanations both quantitatively and qualitatively. While model-agnostic KG-based reordering models contribute with accuracy and diversity metrics, model-agnostic KG explanations are evaluated through online user studies, which are costly and limited by the number of participants. As a result, they are not extensively assessed through offline evaluation.

2.2 Calibration Strategies

Recommender systems research has traditionally focused on enhancing the accuracy of these systems. However, given their significant roles in shopping, entertainment, socialization, and education, it is crucial to go beyond accuracy. They should also emphasize other important qualities, such as diversity in the recommendations, fairness by ensuring all items have a similar opportunity to be recommended, serendipity to surprise users with unexpected good suggestions, and trustworthiness by strengthening system security [Kaminskas and Bridge, 2016; Wang *et al*., 2024].

The concept of calibration was extracted from the field of machine learning for the classification task, and it is used to solve the class imbalance problem. In general, miscalibration is associated with popularity bias where only a few sets of items that are interacted with by many users are recommended, affecting aspects such as diversity and fairness [Abdollahpouri *et al*., 2020]. According to Steck [2018], calibration is defined as:

> **Definition: A classification algorithm is considered calibrated when the predicted class distributions align with the actual data distributions. Specifically, in recommendations, calibration is achieved when the user's various interests are represented in the recommended list in their correct proportions [Steck, 2018].**

In the literature, numerous studies focus on calibrating recommender systems. An initial approach by Steck [2018] introduced a post-processing step that re-ranks recommendations using the maximal-marginal relevance algorithm and employs the Kullback-Leibler (KL) divergence as a calibration metric. The findings demonstrated that while the alignment between the genres of movies that users interacted with and those recommended improved after re-ranking, but at the cost of reduced accuracy.

Building on the proposal by Steck [2018], several adaptations and modifications of the algorithm have been developed to better align users' interests in recommended lists or to enhance the trade-off between accuracy and calibration. For instance, Naghiaei *et al*. [2024] introduced an algorithm aimed at addressing users' needs beyond accuracy metrics. Instead of focusing solely on the genre miscalibration as in [Steck, 2018], their approach emphasized novelty, coverage, surprise, and redundancy in user rankings. Conversely, da Silva and Durão [2023] proposed a new trade-off function that incorporates user bias to better align with users' tendencies. Meanwhile, Souza and Manzato [2024b] developed a processing approach for calibrating recommendations by modifying the Bayesian Personalized Ranking based on Matrix Factorization (BPR-MF) algorithm. They integrated the KL divergence into the cost function to minimize both error and divergence.

Several studies have explored calibration in recommender systems from a user-centric perspective. For example, Abdollahpouri *et al*. [2020] examined the relationship between calibration and popularity bias and fairness, finding that users with a stronger interest in popular items are more prone to miscalibration. Similarly, Lin *et al*. [2020] conducted experiments to understand the causes of miscalibration in collaborative filtering algorithms, concluding that, beyond item popularity, three other factors influence calibration: (1) category-wise user profile entropy, (2) the number of categories, and (3) the size of item categories. Conversely, Alves *et al*. [2024] carried out an online experiment to assess whether users could perceive fairness in calibrated recommendations. An A/B test was conducted, comparing groups that received different forms of calibrated recommendations with a control group that received recommendations without post-processing. The results indicated that calibration did not alter the perception of recommendations despite accuracy trade-offs reported in other studies. However, the study also found that users did not perceive the fairness of recommendations unless it was explicitly explained to them.

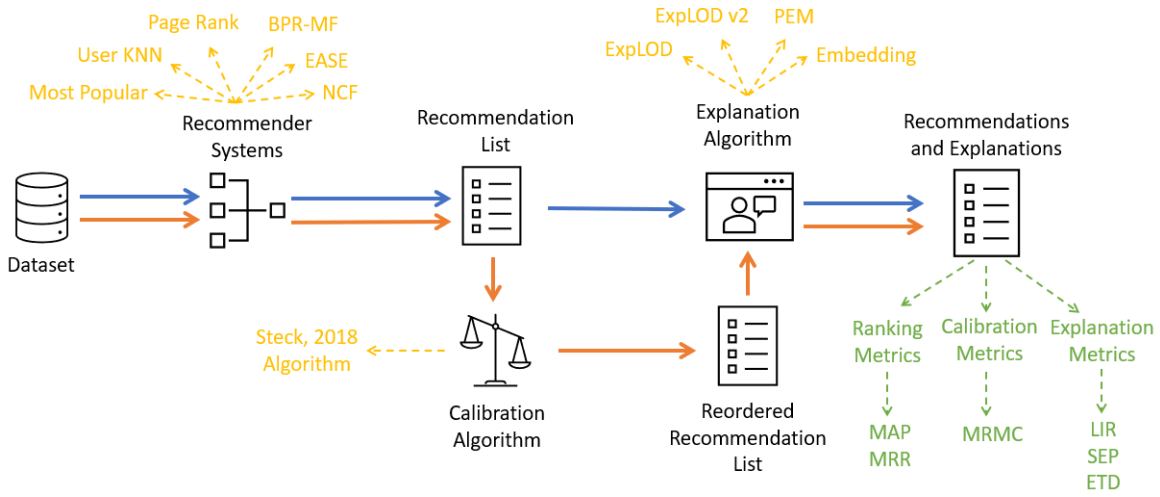


Figure 2. Evaluation Approach: The **blue** and **orange** lines depict the input and output flows of the modules. Dashed lines indicate algorithms (in **yellow**) or metrics (in **green**) within each module.

In this context, research on calibration in recommender systems can be broadly categorized into two subgroups: (1) studies aiming to improve the alignment of user interests in recommendation lists while maintaining accuracy and (2) studies exploring user perceptions and the causes of miscalibration in collaborative filtering algorithms. Despite these efforts, the effect of calibration strategies on other post-processing steps, such as explanation algorithms, remains largely unexplored. Our work addresses this gap by investigating whether calibration strategies influence the quality of explanations in recommendation systems.

3 Proposed Evaluation Approach

One of the key contributions of this paper is the evaluation approach for evaluating explanations, as illustrated in Figure 2. The **blue** line depicts the data flow used to address RQ1 and RQ2, focusing on evaluating and comparing model-agnostic post-hoc KG-based explanations using syntactic and semantic approaches.

Conversely, the **orange** line indicates the flow used to address RQ3. Before generating explanations for the recommendations produced by the Recommender System module, a calibration strategy proposed by Steck [2018] reorders the suggestions using two KG attributes: genre and award received. For instance, in a movie context, by the end of the calibration algorithm module phase, if the distribution of interacted genres is 20% drama, 50% action, and 30% comedy, the distribution of reordered recommended items will more closely align with these proportions compared to the initial list. The answer to RQ3 is then derived from comparing explanations generated with and without the Calibration Algorithm module.

The first two research questions focus on comparing the impact of different embedding algorithms on explanations and evaluating how these explanations compare to syntactic approaches. To address RQ1 and RQ2, the blue flow in the figure is followed. First, six recommender systems are executed using two datasets to generate recommendation lists for all users. Then, three syntactic explanation methods and one embedding-based model are used to generate explanations for each recommendation.

For RQ3, we aim to analyze the impact of calibration approaches on explanations. As shown by the orange flow in Figure 2, this process involves two stacked post-processing steps. First, the recommender systems generate suggestions. Then, a calibration reordering is applied to align the distribution of KG attributes between interacted and recommended items. This reordering takes place before the execution of the explanation algorithms.

In the context of movies, a KG can include various edge types such as genres, awards received, actors, and directors. The calibration algorithm operates by taking as input (1) the user's interacted items, (2) the suggested items generated by a recommendation algorithm, (3) a specific edge type, and (4) the KG itself. Each user's profile will display a distribution of attribute nodes linked to a chosen edge type. For instance, if we focus on the genre edge type, the distribution might show that 20% of interacted movie nodes are linked to drama, 50% to action, and 30% to comedy. The calibration algorithm then adjusts the order of the suggested items to match this distribution in relation to the chosen edge type.

In order to answer RQ1 and analyze how different graph embedding algorithms impact the generation of model-agnostic explanations in Recommender Systems (RS), we used the evaluation approach of Figure 3 that generates explanations for recommendations using graph embeddings produced by the TransE [Lin *et al.*, 2015], ComplEx [Trouillon *et al.*, 2016], and RotatE [Sun *et al.*, 2019] algorithms. The selected explanation is the path with the highest similarity between two embeddings: the path embedding and the user embedding. The path embedding is computed as the sum of the embeddings of the nodes and edges that connect one or more interacted item nodes to a recommended item node. Meanwhile, the user embedding is obtained by summing the embeddings of the interacted item nodes. In Section 4.1, we provide a detailed explanation of the embedding-based strategies.

To answer RQ2, we compare graph embedding-based approaches with three state-of-the-art algorithms for syntactic explanations in Recommender Systems (RS). Our goal

is to determine whether embedding-based approaches outperform syntactic approaches. The implemented syntactic algorithms are: ExpLOD [Musto *et al*., 2016], ExpLOD version 2 (ExpLOD v2) [Musto *et al*., 2019], and the Proposed Property-based Explanation Model (PEM) [Du *et al*., 2022]. All three use strategies to balance the number of references that attribute nodes have between interacted and recommended items to select the most relevant path for an explanation. In Section 4.2, we provide a detailed explanation of these approaches. In Section 4.3, we describe the metrics proposed by Balloccu *et al*. [2022] that assess explanation quality by measuring attribute diversity, attribute popularity, and the recency of items within explanations.

The evaluated strategies are post-hoc and, therefore, independent of the RS. In our evaluation, we consider seven recommendation algorithms based on different approaches:

- **Most Popular** [Cremonesi *et al*., 2010] for non-personalized recommendations. It recommends the most popular items that were not interacted by the user.;
- **Personalized PageRank algorithm** [Musto *et al*., 2016] augmented with the Wikidata graph for graph-based recommendations. This algorithm recommends items based on random walks from user-interacted item nodes. The weights for the random walks performed by the algorithm 80% to interacted items and 20% to all remaining nodes;
- **User-KNN** [Resnick *et al*., 1994] for neighborhood-based. This algorithm performs cosine similarity between users to recommend items that a similar user has interacted with, but the user that will receive the recommendation has not. The parameter K was set to the square root of the total number of users;
- **Embarrassingly Shallow AutoEncoder (EASE)** [Steck, 2019] and **Bayesian Personalized Ranking Matrix Factorization (BPR-MF)** [Rendle *et al*., 2009] for non-neural algorithms. While the EASE algorithm uses a linear auto-encoder architecture to generate suggestions, the BPR-MF is an optimization method for Matrix Factorization (MF) that learns from implicit feedback using a pairwise ranking approach. The model optimizes a criterion where a recommender system learns by preferring an interacted item over a non-interacted item for a given user. The parameter lambda for the EASE algorithm was set to 500 in concordance with the original paper. The parameter of embedding size for BPR-MF was set to 32;
- **Neural Collaborative Filtering (NCF)** [He *et al*., 2017] for neural network-based architectures. This algorithm is an ensemble of an Artificial Neural Network and Matrix Factorization. For both algorithms the user and item embedding was set to 32, with four layers of 64, 32, 16, 8 neurons, running for 10 epochs and a batch size of 256 samples. A negative sampling was also used were for each positive sample on the train set, 4 negative samples were added based on unseen items. For testing a leave-one-out evaluation was conducted, as in the original paper, therefore, the last interaction of every user along with 100 items that were not interacted are on the training set.

We used the CaseRecommender library [da Costa *et al*., 2018] to implement the Most Popular, User-KNN, and BPR-MF algorithms. The implementations of the other recommendation algorithms, along with the explanation algorithms, metrics, and queries used to extract triples for constructing the KG, are available in an open-source repository[1], which is one of our contributions in this work. The split for all algorithms, excluding the NCF, which is based on a leave-one-out evaluation, was 80% for training and 20% for testing.

For evaluations, we considered the MovieLens Latest (ml-latest-small) [Harper and Konstan, 2015] and LastFM-2k [Cantador *et al*., 2011] datasets to generate explanations for the Top-5 recommendations of six RS algorithms for all users. For the highest-ranked items from each recommendation algorithm, all seven (four embedding-based and three syntactic) model-agnostic explanation algorithms were executed to obtain explanation quality metrics. Taking into account the reproducibility guidelines in RS proposed by Ferrari Dacrema *et al*. [2021] and the robustness in evaluating the explanations highlighted in Tchuente *et al*. [2024], and to ensure a rigorous evaluation, we applied six RS algorithms from different families, using 90% of the dataset for training and 10% for testing to generate explanations.

We used a Wikidata Linked Open Data (LOD) KG to extract data for the domains of movies and musical artists to implement all evaluated explanation and calibration algorithms. Items where no data was found on the LOD was removed from the datasets. The processed data retained 99% of the original interactions for the MovieLens dataset and 89% for the LastFM dataset. A summary of the MovieLens and LastFM dataset statistics before and after preprocessing, as well as KG details, is available in Table 1. The data extraction software and obtained triples from the KG for the domains are available in this manuscript's open-source repository.

Table 1. Statistics of the original and processed datasets, as well as KG information regarding the number of entities, triples, and edges.

		MovieLens	LastFM
Original Dataset	users	610	1,892
	items	9,724	17,632
	interactions	100,836	92,834
Generated Dataset	users	610	1,875
	items	9,517	11,641
	interactions	100,521	83,017
Wikidata KG	entities	78,703	34,297
	triples	295,787	134,197
	edge types	23	33

To address RQ3, we integrated the calibration method suggested by Steck [2018] within our evaluation approach. We applied this calibration to two edge types in the KG for results from each recommendation algorithm, with the goal of aligning the distribution of user-interacted item nodes with those of recommended item nodes. Following calibration, we executed the explanation algorithms to determine whether stacking these two post-processing methods would influence the quality of explanations in recommender systems. Our ex-

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

periment was conducted using the five collaborative filtering methods, which are inherently more susceptible to popularity bias from the MovieLens dataset. Therefore, for this experiment, we included the Most Popular, UserKNN, BPR-MF, EASE and NCF algorithms.

As explanation algorithms, we included the three syntactic baselines. The ComplEx [Trouillon *et al.*, 2016] algorithm was chosen to represent an embedding approach as it was the most effective embedding algorithm among TransE [Lin *et al.*, 2015], RotatE [Sun *et al.*, 2019], and ComplEx in accordance with the answers of RQ1 and RQ2. The edge types chosen to calibrate recommendations upon were genre and award received.

4 Materials and Methods

4.1 Embedding Explanation Algorithm

In the literature, generating explanations for recommendations using Knowledge Graphs (KGs) is often approached as a path-finding problem [Du *et al.*, 2022; Musto *et al.*, 2016, 2019]. This method involves identifying connections between user-interacted and recommended items through shared attributes. As defined in Section 2.1, the KG models entities and their relationships, forming the basis for both syntactic and semantic techniques by unifying user interactions and item metadata. Model-agnostic KG explanations aim to identify the most relevant **explanation path**, which is defined as a KG path connecting an interacted item to a recommended item.

> **Definition: An explanation path is represented as $argmax(\forall c \in C : agg(rel(n) \forall n \in c))$. Where, for each path c in the set of paths C that connect an interacted item to a recommended item, an aggregation function (agg) — such as mean or sum — is applied to the relevance rel of each node n within the path.**

Figure 3 illustrates the model-agnostic KG embedding approach. The nodes interacted with by the user in the KG are shown in blue, attribute nodes are represented in yellow, and the recommended item node is in red. The same colors apply to the vectors that represent the embeddings of these nodes. The black vector represents an embedding of the relations between the nodes, represented by an arrow of the same color in Figure 3.

Two embeddings are required to compute the relevance rel of an explanation path: the user embedding and the path embedding. The user embedding is calculated using the sum pooling of the embeddings of the interacted items. The path embedding, in turn, is obtained by sum pooling all embeddings of items, attributes, and relations along the path that connects an interacted item node to a recommended item node in the KG. Equations 1 and 2 define the computation for each embedding, where I represents the set of interacted item nodes, and P is the set of item, relation, and attribute nodes in a path. The $embedding$ method returns the embedding of the node passed as a parameter. The paths were extracted using Dijkstra's algorithm [Dijkstra, 2022].

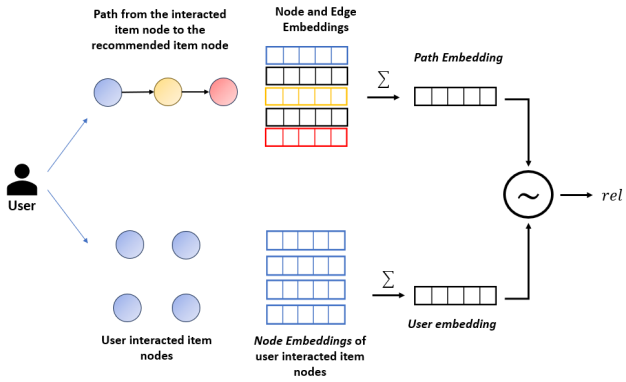


Figure 3. Structure of the proposed model. The **blue** nodes represent the items interacted with by the user, the yellow) nodes represent attribute nodes, and the **red**) node is the recommended item node. The same color scheme applies to the vectors generated by the embedding algorithms. Black vectors represent edge embeddings. The symbol $\sum$ represents a sum pooling operation, and $\sim$ is the cosine similarity between two embeddings, with rel being the output of the cosine similarity function.

$$embed(user) = \sum_{i \in I} embedding(i) \tag{1}$$

$$embed(path) = \sum_{n \in P} embedding(n) \tag{2}$$

The chosen explanation path is the one with the greatest similarity to the user's embedding among all the paths that connect at least one interacted item to the recommended item. In this sense, the cosine similarity of the user's embedding with the path embeddings computes this proximity. The maximum path length was set to 5, and the number of interacted items per explanation to 3, following the same configuration as the baselines. The similarity value is represented in Equation 3, where $embed(user)$ is the user's embedding and $embed(path)$ is the path's embedding.

$$sim(user, path) = \frac{embed(user).embed(path)}{||embed(user)||.||embed(path)||} \tag{3}$$

The selection of the explanation path adheres to Equation 4, where, for all explanation paths within the set $Paths$ that connect a node of an interacted item to a node of a recommended item through shared attributes, the path embedding with the highest cosine similarity to the user embedding is chosen.

$$argmax(\forall path \in Paths\ sim(user, path)) \tag{4}$$

Algorithm 1 is the pseudocode for the explanation algorithm based on embeddings and uses three parameters: the user's history items (pro_items), the recommended item to be explained (rec_item), and the trained graph embedding model ($model$). In line 2, the user's embedding is created by summing the embeddings of the interacted item nodes, which are returned by the graph embedding model.

Next, in line 3, all paths from an interacted item to the recommended item are obtained using Dijkstra's algorithm [Dijkstra, 2022]. Since in the proposed algorithm the user's embedding is compared to the path embeddings, lines 4 and 5 initialize variables that will store the current maximum similarity and the path embedding most similar to the user's.

Lines 6 to 14 represent the generation of path embeddings and their comparison with the user's embedding. Thus, for each path, the embedding is generated by summing the embeddings of the nodes that compose it. Therefore, $c.nodes()$ returns a list of nodes in the path c and $model(c.nodes())$ returns the embeddings of these nodes. Then, the cosine similarity between the user's and the path's embeddings is computed in line 8. When this value is greater than the current maximum, both the maximum value and the path corresponding to this maximum similarity are updated in lines 10 and 11. In line 14, the algorithm returns the path embedding with the greatest similarity to the user's embedding.

Algorithm 1 Embedding Explanation Generation

```
1:  function embed_expl(pro_items, rec_item, model)
2:      user_embed ← sum(model(pro_items))
3:      paths ← dijkstra(pro_items, rec_item)
4:      max ← −1
5:      max_path ← []
6:      for c in paths do
7:          path_embed ← sum(model(c.nodes()))
8:          sim ← cosine(path_embed, user_embed)
9:          if sim > max then
10:             max ← sim
11:             max_path ← c
12:         end if
13:     end for
14:     return max_path
15: end function
```

The embedding proposal was made by generating embeddings from the KGs extracted from the Wikidata LOD for the movie and artistic domains of the MovieLens and LastFM databases, respectively. The choice of embedding algorithms was made based on different families of algorithms. While TransE [Lin *et al*., 2015] and RotatE [Sun *et al*., 2019] use the translational approach, the ComplEX algorithm [Trouillon *et al*., 2016], in turn, is a bilinear algorithm. The TransE algorithm was compared to RotatE to verify whether the evolution of the state of the art in graph embeddings representation within a family of algorithms improves explanation quality metrics.

Translational models are based on the concept of cartesian coordinates [Cao *et al*., 2024; Zhang *et al*., 2019] where, considering a triple (h, r, t), where h and t are nodes in the KG and r is the relation that connects the two nodes, a linear transformation representing the distance among these elements generates the cost function to be minimized, so $h+r \approx t$. RotatE uses the distance equation $d_r(h, t) = ||h \circ r - t||$, whereas TransE uses the function $d_r(h, t) = ||h + r - t||$. The symbol $\circ$ denotes the element-wise product. Bilinear models, meanwhile, assess the similarity of h, r, and t using inner products and multiplicative operations, usually involving bilinear forms [Li and Yang, 2022].

For training the graph embedding models, parameter optimization was performed where the learning rate (λ) was varied between 0.1 and 0.001, and the batch size (B) for the parameter update was 128 and 256. The embedding size (K) was also optimized between 200 and 400. The use of negative sampling was also varied. When negative sampling was present, 10 negative samples per positive sample were generated for a batch. The number of epochs for training was fixed at 40 for all models. Since the ComplEX model [Trouillon *et al*., 2016] uses the Stochastic Gradient Descent algorithm with AdaGrad [Duchi *et al*., 2011] as an optimizer, the learning rate is adjusted during training.

The KG triples were divided into training, validation, and test sets in 0.8, 0.1, and 0.1 proportions, respectively. In the KG extracted from Wikidata for the MovieLens dataset, 235,466 triples were used for training, 29,434 for validation, and 29,433 for testing. For the KG extracted for LastFM, the training, validation, and test splits were 101,516, 12,690, and 12,689, respectively. The Pykeen library [Ali *et al*., 2021] was used to implement the graph embedding models. The model's accuracy is measured by the Hit Rate metric, which evaluates how well the model finds the node that completes a triple. Thus, given a node embedding h and a relation embedding r, the model must find the correct node embedding t corresponding to the triple (h, r, t) present in the graph. The Hit Rate metrics on the test set for the best parameter models for the KG embeddings extracted from Wikidata for MovieLens and LastFM are presented in Table 2.

The best models achieved the following parameters: For the MovieLens dataset, in the TransE model K was 200, λ was 0.001, B was 256, and negative sampling was not used; for the ComplEX model K was 400, B was 128, with negative sampling; in the RotatE model K was 200, λ was 0.001, B was 128, with negative sampling. For the LastFM dataset, in the TransE model K was 200, λ was 0.001, B was 256, and negative sampling was not used; for the ComplEX model K was 200, B was 128, without negative sampling; in the RotatE model K was 200, λ was 0.001, B was 128, with negative sampling.

Table 2. Test set metrics for different graph embedding algorithms for the KG of the MovieLens and LastFM datasets. $H@n$ is the Hit Rate metric of the model for correctly completing a triple considering the n nearest nodes.

		TransE	RotatE	ComplEX
MovieLens	H@1	0.0317	0.0982	0.0115
	H@3	0.0956	0.1595	0.0209
	H@5	0.1282	0.1927	0.0261
	H@10	0.1727	0.2418	0.0362
LastFM	H@1	0.0570	0.1852	0.0029
	H@3	0.1135	0.2725	0.0076
	H@5	0.1481	0.3154	0.0118
	H@10	0.1998	0.3735	0.0219

4.2 Syntactic Explanation Algorithms

Three syntactic algorithms were implemented for comparison with the four results obtained through the application of the graph embedding method in the evaluation approach described in Section 4.1. Unlike embedding-based methods, syntactic approaches use the occurrence of the attribute node related to the item node to determine its relevance.

The ExpLOD method [Musto *et al*., 2016] ranks properties in the Knowledge Graph (KG) using an adapted TF-IDF approach. The relevance of an attribute is determined by the

frequency of references to the attribute from both interacted and recommended items relative to the total references to the attribute across all items. Equation 5 illustrates the calculation for the relevance value of an attribute p, where n_{p,I_u} represents the number of links from the set of items interacted with by the user I_u to the attribute p. Similarly, n_{p,I_r} denotes the number of links connecting the attribute p to the recommended items I_r, and $IDF(p)$ stands for the Inverse Document Frequency of p, calculated as $\log(\frac{|C|}{n_{p,I_C}})$, where $|C|$ represents the total number of items in the catalog and n_{p,I_C} is the total number of items referencing the attribute p. The values α and β are weights and were set to 0.5 according to Musto *et al*. [2016].

$$explod(p, I_u, I_r) = (\alpha \frac{n_{p,I_u}}{|I_u|}) + (\beta \frac{n_{p,I_r}}{|I_r|}) * IDF(p) \quad (5)$$

Equation 6 shows the calculation for ranking attributes in ExpLOD v2 [Musto *et al*., 2019], which is very similar to its previous version but also encompasses broader attributes. For example, consider the film 'La La Land' from 2016, which is classified with the attribute 'romance' in the Wikidata KG. This classification implies that the film is also associated with broader attributes like 'interpersonal relationship' and 'love' since 'romance' is a subclass of these attributes. As a result, for broader KG attributes b that have subclasses, the relevance is the sum of Equation 5 for all attributes p_{bi} in the set $P_c(b)$ that are children of b, multiplied by the IDF of the broader class ($IDF(b)$). Therefore, the ExpLOD algorithms rank attributes that are popular among the set of interacted items but rare in the catalog items set.

$$explod(b, I_u, I_r) = \sum_{i=1}^{|P_c(b)|} explod(p_{bi}, I_u, I_r) * IDF(b) \quad (6)$$

The scoring mechanism used in the Property-based Explanation Model (PEM) is represented by Equation 7 and, unlike the ExpLOD algorithms, considers the number of interacted items that reference the attribute instead of the number of links. To score an attribute p, first, the number of interacted items that reference the attribute is considered, $|\overline{I(p, I_u)}|$, where I_u represents the set of items with which the user interacted. This value is then normalized by the total number of items the user interacted with, denoted by $|I_u|$.

Furthermore, the equation considers the number of items in the catalog C connected to the attribute $|\overline{I(p, C)}|$. Similar to the previous term, this value is normalized by the total number of items in the catalog, denoted by $|C|$. Finally, the logarithm of the total number of items in the catalog connected to the attribute $\log(|I(p, C)|)$ is calculated to amplify the importance of relatively rare attributes in the catalog.

$$score_pem(p, I_u, I_r, C) = \frac{|\overline{I(p, I_u)}|/|I_u|}{|\overline{I(p, C)}|/|C|} * \log(|I(p, C)|) \quad (7)$$

For all the algorithms, the path with the highest average attribute relevance is chosen as the explanation. Additionally, the maximum path length was set to five, and the maximum number of interacted items for a recommended item was three.

4.3 Explanation Metrics

In their study, Balloccu *et al*. [2022] conducted an online survey to explore what users perceive as quality explanations. The findings revealed that:

> **Definition:** A quality explanation is characterized by three key attributes: the recency of items connecting the interacted and recommended items, the popularity of the attributes linking these items, and the diversity of these attributes across various explanations.

The Linking Interaction Recency (LIR) metric measures the recency of the items interacted with by the user that form an explanation; the Shared Entity Popularity (SEP) measures the popularity of the attributes displayed in explanations for a single user, and the Explanation Type Diversity (ETD) measures the number of different attributes in the explanations. Thus, the metrics proposed by Balloccu *et al*. [2022] to evaluate explanation quality define that quality explanations find different paths (ETD), but use popular attributes (SEP) and connect recommended items with recently interacted items (LIR). All metrics range from 0 to 1, where 1 is the optimal value, except for ETD, which can be greater than 1 if the path has more than one attribute.

Equations 8 and 9 represent the metrics for LIR and SEP, respectively. These metrics are calculated based on the mean of normalized exponentially weighted moving average equations for each interacted item and attribute within an explanation.

For LIR, the values of the interacted items (p^i) are calculated using their respective timestamps (t^i), which are normalized using the min-max method to range between 0 and 1. The recursive nature of the function ensures that the value of a property i depends on $i-1$, with values ordered in ascending order of timestamps. Thus, $LIR(p^1, t^1)$ is equivalent to t_1. The parameter β is typically set to 0.3, as suggested in Balloccu *et al*. [2022]. Consequently, LIR assigns higher values to explanations that connect recommendations with more recently interacted items.

$$LIR(p^i, t^i) = (1-\beta) * LIR(p^{i-1}, t^{i-1}) + \beta * t^i \quad (8)$$

In Equation 9, the Shared Entity Popularity metric (SEP) quantifies the popularity of attributes considering the number v^i of item nodes connected to the attribute e^i. Min-max sorting and normalization are also applied to the number of references an attribute has to other nodes. Consequently, $SEP(e^1, v^1)$ corresponds to v_1, representing the attribute with the smallest number of references in the KG. High values of SEP indicate that the attributes in explanations are popular.

$$SEP(e^i, v^i) = (1-\beta) * SEP(e^{i-1}, v^{i-1}) + \beta * v^i \quad (9)$$

Finally, Equation 10 defines the Explanation Type Diversity metric (ETD), which quantifies the diversity of

attributes in explanations associated with recommendations. It calculates the ratio of the number of properties in the recommended list $|\omega_{L_u}|$ to the minimum between the length of the recommendation list k and the total number of possible attributes ω_L that could form an explanation. ETD provides insight into the variety of attributes presented in explanations and helps to assess whether the explanation algorithm tends to favor repetitive attributes. Higher values of ETD indicate a greater diversity of attributes.

$$ETD(S) = \frac{|\omega_{L_u}|}{min(k, |\omega_L|)} \tag{10}$$

4.4 Calibration Approach

Calibration in recommendation systems is a re-ranking method that aims to align the recommendation list with user preferences, i.e. minimize the divergence between them. To quantify this divergence, several methods such as Jensen-Shannon Divergence, Chi-Squared Test, Kullback-Leibler can be employed. In this work, we use Kullback-Leibler divergence because it is well adopted in literature [Steck, 2018; da Silva *et al.*, 2021; Alves *et al.*, 2024]. The equation 11 presents the mathematical formulation. Similar to Steck [2018], we set $\alpha = 0.01$ as a regularization term to prevent division by zero, and denote c for the category, i.e., edge type (genre, award received, and so on).

$$D_{\mathrm{KL}}(p \parallel q) = \sum_{c \in C} p(c|u) \log \left(\frac{p(c|u)}{\hat{q}(c|u)} \right) \tag{11}$$

$$\hat{q}(c|u) = (1 - \alpha) \cdot q(c|u) + \alpha \cdot p(c|u) \tag{12}$$

where $p(c|u)$ denotes the target distribution, while $\hat{q}(c|u)$ represents the approximating distribution.

On the other hand, Equation 13 defines the optimization function used to derive the optimal calibrated set of items, I_u^*, i.e., the re-ranked recommendation list for user u.

$$I_u^* = \max_{I_u} \Big((1 - \lambda) \cdot s(I_u) - \lambda \cdot D_{KL}(p, q(I_u)) \Big) \tag{13}$$

where λ is a trade-off parameter from 0 to 1 that balances accuracy and fairness. The term $s(I_u)$ represents the sum of scores for the recommended items for the user u, while $D_{KL}(p, q(I_u))$ measures the Kullback-Leibler divergence between the target distribution p and the approximating distribution $q(I_u)$.

4.5 Calibration Metrics

To evaluate the calibrated re-ranking list, we consider metrics to capture the relevance (accuracy) and fairness (calibration) of the recommendation. Achieving a balance between these two factors is essential to ensure that recommendations are accurate and aligned with user preferences.

To evaluate the recommended items' relevance, we use Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR), where higher values indicate better performance. MAP metric considers both the position and precision of recommended items. It is calculated using the formula in Equation 14, where U represents the set of users, and AP denotes the average precision for a specific user u. Average precision is calculated as shown in Equation 15. In this equation, R represents the number of relevant items for a user, N denotes the total number of items recommended, $P@k$ is the precision in position k, and rel_k is a binary indicator that specifies whether the item at position k is relevant or not.

$$MAP = \frac{1}{U} \sum_{u \in U} AP_u \tag{14}$$

$$AP_u = \frac{1}{R} \sum_{k=1}^{N} P@k \cdot rel_k \tag{15}$$

Mean Reciprocal Rank (MRR) measures the relevance of a recommended item by considering its position in the ranking. It assigns higher scores when relevant items appear earlier in the list. Equation 16 details the definition to calculate MRR.

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank_u} \tag{16}$$

where U denotes the set of users, and $rank_u$ represents the position of the first relevant item in the recommendation list for user u.

On the other hand, to assess calibration, we employ the Mean Rank Miscalibration (MRMC) metric proposed by da Silva *et al.* [2021]. Unlike relevance-based metrics, where higher values indicate better performance, lower MRMC values signify a closer alignment between recommended items and the user profile. MRMC quantifies the overall miscalibration by averaging the Rank Miscalibration (RMC) values, which, in turn, are derived from the mean Miscalibration (MC) values.

$$RMC_u = \frac{\sum_{k=1}^{N} MC(p, q(I@k))}{N} \tag{17}$$

$$MRMC = \frac{\sum_{u \in U} RMC_u}{|U|} \tag{18}$$

In Equation 17, the MC metric quantifies the divergence between the distributions p and $q(I@k)$, where $q(I@k)$ denotes the distribution at rank k. Consequently, the RMC_u metric indicates how well the recommendations for the user u align with their preferences. Finally, the MRMC metric (see Equation 18) represents the average of RMC across all users, providing a measure of the overall misalignment between the recommended items and user profiles.

5 Results

To address the research questions RQ1 and RQ2, we applied the syntactic algorithms ExpLOD, ExpLOD v2, and PEM, as well as the embedding approach using the TransE, RotatE, and ComplEX algorithms. This analysis was conducted for all users in the MovieLens and LastFM datasets, focusing on the top five items recommended by the algorithms Most Popular, BPR-MF, PageRank, UserKNN, EASE, and NCF,

Table 3. Table of LIR, ETD, and SEP metrics for explanations of the top five recommended items in the MovieLens dataset. For each recommendation algorithm (row) and each metric, values in **bold** indicate the highest, and underlined values indicate the lowest across the explanation algorithms (columns).

		TransE	RotatE	ComplEX	ExpLOD	ExpLOD v2	PEM
Most Popular	LIR	0.03147 ± 0.12	0.0275 ± 0.11	0.0234 ± 0.09	**0.0945 ± 0.15**	0.0834 ± 0.14	0.0320 ± 0.07
	ETD	0.6718 ± 0.21	0.6842 ± 0.20	**0.9537 ± 0.31**	0.5809 ± 0.19	0.5947 ± 0.19	0.9390 ± 0.12
	SEP	0.52104 ± 0.18	**0.6869 ± 0.13**	0.5625 ± 0.15	0.6322 ± 0.14	0.6107 ± 0.12	0.1430 ± 0.13
Page Rank	LIR	0.0310 ± 0.11	0.0312 ± 0.12	0.0241 ± 0.10	**0.0939 ± 0.15**	0.0872 ± 0.15	0.0323 ± 0.08
	ETD	0.7335 ± 0.21	0.6570 ± 0.22	0.9305 ± 0.31	0.5563 ± 0.20	0.6043 ± 0.19	**0.9389 ± 0.12**
	SEP	0.4662 ± 0.20	**0.7022 ± 0.15**	0.5557 ± 0.16	0.6250 ± 0.16	0.5606 ± 0.15	0.1095 ± 0.11
UserKNN	LIR	0.03327 ± 0.12	0.0352 ± 0.12	0.0234 ± 0.10	**0.1010 ± 0.14**	0.0914 ± 0.13	0.0322 ± 0.08
	ETD	0.7055 ± 0.24	0.6849 ± 0.22	**0.9859 ± 0.33**	0.6593 ± 0.19	0.6409 ± 0.19	0.9452 ± 0.11
	SEP	0.5202 ± 0.23	0.2311 ± 0.15	**0.6103 ± 0.16**	0.5803 ± 0.16	0.5275 ± 0.14	0.131 ± 0.12
BPR-MF	LIR	0.0408 ± 0.16	0.0298 ± 0.11	0.0244 ± 0.09	**0.1048 ± 0.14**	0.0945 ± 0.13	0.0306 ± 0.07
	ETD	0.6826 ± 0.26	0.6934 ± 0.24	**0.9855 ± 0.32**	0.6891 ± 0.19	0.6937 ± 0.19	0.9583 ± 0.10
	SEP	0.5755 ± 0.23	0.2151 ± 0.15	0.5013 ± 0.16	**0.6113 ± 0.14**	0.5428 ± 0.14	0.1400 ± 0.12
EASE	LIR	0.0312 ± 0.12	0.0295 ± 0.11	0.0209 ± 0.09	**0.1000 ± 0.14**	0.0912 ± 0.14	0.03193 ± 0.08
	ETD	0.7345 ± 0.24	0.6751 ± 0.23	**0.9724 ± 0.31**	0.6204 ± 0.20	0.6328 ± 0.19	0.9459 ± 0.11
	SEP	0.4952 ± 0.49	**0.6583 ± 0.17**	0.6089 ± 0.16	0.5743 ± 0.17	0.5274 ± 0.15	0.1350 ± 0.13
NCF	LIR	0.0320 ± 0.13	0.0244 ± 0.09	0.0199 ± 0.08	**0.1181 ± 0.13**	0.1035 ± 0.12	0.0380 ± 0.08
	ETD	0.6966 ± 0.29	0.8080 ± 0.25	**1.0100 ± 0.32**	0.8432 ± 0.16	0.8161 ± 0.16	0.9885 ± 0.05
	SEP	**0.6122 ± 0.22**	0.2511 ± 0.14	0.3955 ± 0.14	0.5868 ± 0.14	0.5350 ± 0.13	0.1613 ± 0.11

while considering the LIR, ETD, and SEP metrics. Additional results for other metrics, such as precision and diversity, along with examples of explanations generated by each algorithm, are available on Appendix 2 and Appendix 3, respectively.

Tables 3 and 4 show the mean and standard deviation of the SEP, ETD, and LIR explanation quality metrics for all users in the MovieLens and LastFM datasets, respectively, focusing on the top five items recommended by each algorithm. The first columns correspond to the recommendation algorithm and the quality metric. The subsequent three columns provide results for the proposed method using three different embedding algorithms, while the last three columns show results for the three syntactic methods. Values in bold are the highest among the algorithms, and underlined values are the lowest. We highlight the highest and lowest values to emphasize the trade-off between explanation objectives and quality attributes.

An important aspect of item attributes is that they follow a long-tail distribution, where only a few attribute nodes are linked to many item nodes in the KG [Ferraro, 2019]. As a result, including more attributes in explanations increases the likelihood of selecting a less popular attribute [Tintarev and Masthoff, 2015; Balloccu *et al*., 2022; Balog and Radlinski, 2020].

To address research question RQ3, we applied three syntactic algorithms: ExpLOD, ExpLOD v2, and PEM. Additionally, from the embedding-based approaches, we selected the ComplEX algorithm, as it outperformed both TransE and RotatE in our evaluations.

Our experiments were conducted using the MovieLens dataset and evaluated five recommender algorithms: Most Popular, BPR-MF, UserKNN, EASE, and NCF. To assess calibration, we used the MAP, MRR, and MRMC metrics, and to evaluate the quality of explanations, we employed the LIR, ETD, and SEP metrics. Figure 9 and 10 represent the result for relevance of recommendations, while Figure 11 represents the results for alignment of recommendations for user profile. These results provide insights into how effectively the models balance relevance and personalization across different categories and trade-offs. On the other hand, Figures 4, 5, 6, 7, 8 present the results related to the quality of the explanation in combination with the calibration of different categories in the best trade-offs (Table 5, which is provided in Appendix 1 - Calibration Trade-Offs) concerning Equation 13.

5.1 Answer to RQ1: What is the impact of different types of graph embedding algorithms on the quality of explanations generated for recommender systems?

To address RQ1 and differentiate the impact of various embedding algorithms on explanation generation, the first three columns of the tables present the explanation quality results for the model-agnostic method, utilizing the graph embedding algorithms TransE [Lin *et al*., 2015], RotatE [Sun *et al*., 2019], and ComplEX [Trouillon *et al*., 2016].

Regarding the TransE [Lin *et al*., 2015] and RotatE [Sun *et al*., 2019] methods, which belong to the family of translational graph embedding algorithms, the HitRate metrics in the training set of the graph embedding model in Table 2 demonstrate that the RotatE algorithm [Sun *et al*., 2019] represents an advancement over the TransE [Lin *et al*., 2015] translational model. In the LastFM dataset, this advancement in state-of-the-art techniques was reflected in the improved explanation quality metrics, with both ETD and SEP showing better results for the RotatE algorithm compared to TransE. However, this improvement was not observed in the MovieLens dataset, where enhancements in HitRate metrics for the same model type did not necessarily translate to better explanation quality metrics. Thus, even

Table 4. Table of LIR, ETD, and SEP metrics for explanations of the top five recommended items in the LastFM dataset. For each recommendation algorithm (row) and each metric, values in **bold** indicate the highest, and underlined values indicate the lowest across the explanation algorithms (columns).

		TransE	RotatE	ComplEX	ExpLOD	ExpLOD v2	PEM
Most Popular	LIR	0.0104 ± 0.06	0.0100 ± 0.06	0.0123 ± 0.08	0.0182 ± 0.09	$\mathbf{0.0189 \pm 0.11}$	0.0143 ± 0.08
	ETD	0.8247 ± 0.27	0.9319 ± 0.25	$\mathbf{1.1394 \pm 0.28}$	0.7023 ± 0.17	0.4927 ± 0.22	0.9212 ± 0.12
	SEP	0.5084 ± 0.22	0.6617 ± 0.21	0.5181 ± 0.16	0.7097 ± 0.17	$\mathbf{0.7537 \pm 0.23}$	0.1214 ± 0.08
Page Rank	LIR	0.0108 ± 0.07	0.008 ± 0.06	0.0125 ± 0.07	0.0191 ± 0.10	$\mathbf{0.0212 \pm 0.12}$	0.0134 ± 0.07
	ETD	0.7683 ± 0.25	0.8704 ± 0.30	$\mathbf{1.0769 \pm 0.32}$	0.6100 ± 0.20	0.5447 ± 0.19	0.9440 ± 0.10
	SEP	0.4820 ± 0.18	0.6359 ± 0.17	0.5022 ± 0.18	0.6501 ± 0.21	$\mathbf{0.7164 \pm 0.21}$	0.1209 ± 0.09
UserKNN	LIR	0.0102 ± 0.07	0.0090 ± 0.05	0.0117 ± 0.07	0.0183 ± 0.10	$\mathbf{0.0191 \pm 0.11}$	0.0158 ± 0.08
	ETD	0.7760 ± 0.26	0.8688 ± 0.31	$\mathbf{1.0624 \pm 0.32}$	0.5335 ± 0.20	0.5355 ± 0.18	0.9106 ± 0.14
	SEP	0.5071 ± 0.19	$\mathbf{0.6088 \pm 0.18}$	0.4540 ± 0.18	0.5288 ± 0.26	0.2810 ± 0.22	0.1417 ± 0.10
BPR-MF	LIR	0.0110 ± 0.06	0.0096 ± 0.06	0.0113 ± 0.07	0.0191 ± 0.10	$\mathbf{0.0204 \pm 0.11}$	0.0164 ± 0.08
	ETD	0.8403 ± 0.26	0.9219 ± 0.31	$\mathbf{1.1021 \pm 0.31}$	0.6145 ± 0.21	0.6196 ± 0.19	0.9450 ± 0.11
	SEP	0.5312 ± 0.18	0.6002 ± 0.17	0.5984 ± 0.18	0.5605 ± 0.23	$\mathbf{0.6302 \pm 0.19}$	0.1759 ± 0.12
EASE	LIR	0.0102 ± 0.07	0.0092 ± 0.05	0.0111 ± 0.07	0.0188 ± 0.11	$\mathbf{0.0194 \pm 0.11}$	0.0154 ± 0.08
	ETD	0.7934 ± 0.25	0.8753 ± 0.32	$\mathbf{1.0707 \pm 0.32}$	0.5474 ± 0.20	0.5585 ± 0.18	0.9246 ± 0.13
	SEP	0.5026 ± 0.19	$\mathbf{0.5870 \pm 0.19}$	0.4639 ± 0.18	0.5307 ± 0.25	0.2861 ± 0.21	0.1466 ± 0.10
NCF	LIR	0.0098 ± 0.06	0.0106 ± 0.06	0.0131 ± 0.07	$\mathbf{0.0182 \pm 0.09}$	0.0162 ± 0.08	0.0162 ± 0.08
	ETD	0.9409 ± 0.27	1.0318 ± 0.32	$\mathbf{1.2057 \pm 0.29}$	0.7775 ± 0.18	0.7867 ± 0.18	0.9589 ± 0.09
	SEP	0.6302 ± 0.17	$\mathbf{0.6509 \pm 0.16}$	0.6153 ± 0.17	0.5904 ± 0.19	0.5514 ± 0.20	0.2748 ± 0.15

simpler graph embedding models can produce vector representations that lead to quality explanations.

In this context, the ComplEX algorithm [Trouillon *et al*., 2016], from the family of bilinear graph embedding algorithms, achieved the most consistent results across both datasets. This algorithm notably attained ETD above 0.65 and SEP above 0.45 for all datasets and algorithms. This success is due to the use of bilinear forms in generating graph vector representations, which allows for the modeling of more complex patterns between nodes and edges. In contrast, translational models create embeddings by approximating vectors through translations, which limits their expressiveness.

Therefore, in answering RQ1, using an embedding model that captures more complex relationships between nodes and edges has led to improved explanation quality metrics.

5.2 Answer to RQ2: Are graph embedding-based strategies actually capable of generating better explanations in recommender systems compared to syntactic approaches?

To address RQ2 and compare the differences between syntactic and semantic approaches, we considered the models ExpLOD [Musto *et al*., 2016], ExpLOD v2 [Musto *et al*., 2019], and PEM [Du *et al*., 2022], as shown in the last three columns of Table 3 and Table 4. It becomes clear that, in both datasets, syntactic algorithms, particularly ExpLOD and ExpLOD v2, outperformed other algorithms regarding the LIR and SEP metrics. Conversely, embedding approaches excelled in the ETD diversity metric.

For example, with the RotatE algorithm, the SEP and ETD metrics performed better compared to syntactic algorithms when applied to the Most Popular and PageRank in the MovieLens dataset and UserKNN, BPR-MF, EASE, and NCF in LastFM. This demonstrates embedding methods' capacity to offer diverse explanatory paths across recommendations while maintaining attribute popularity. However, embedding algorithms showed lower levels of LIR, likely due to their training methodology. While they find diverse paths for explanations, they only incorporate the interacted item in the pooling sum process to produce the explanation path and user embeddings. In contrast, syntactic methods prioritize interacted item nodes that connect strongly with attribute nodes in the KG to determine the most relevant path for explanation.

An exception is the PEM algorithm, which shows a trade-off between diversity and attribute popularity. Although it achieves high diversity, it exhibits the lowest SEP across metrics. This is because, unlike ExpLOD algorithms based on TF-IDF, PEM normalizes the number of items referencing an attribute against those in the item catalog. Given the catalog's size, PEM tends to present more diverse items.

Thus, syntactic algorithms rely on a trade-off between the connections of attribute nodes with item nodes to define explanation path relevance. These algorithms select explanations that feature attribute nodes linked to many interacted items by the user, but less so to the entire item set. Particularly, ExpLOD [Musto *et al*., 2016] and ExpLOD v2 [Musto *et al*., 2019] prioritize popularity, while PEM [Du *et al*., 2022] focuses on diversity. On the other hand, embedding models are trained to complete triples in graphs. For a node h and a relation r, these algorithms learn to identify the correct node t in the triple (h, r, t) in the KG. As a result, explanation algorithms utilizing vector representations tend to be balanced concerning SEP and ETD metrics.

Our findings for RQ2 reveal that syntactic methods are more influenced by popularity, as they select explanations based on the number of connections an attribute node has

with item nodes. In contrast, embedding-based methods choose explanation paths based on the similarity of vector representations of nodes and edges within the graph. Consequently, the popularity of attribute nodes among item nodes does not significantly affect the choice of explanation for a recommended item, making embedding-based methods more balanced and, therefore, superior to syntactic methods.

In conclusion, the response to RQ2 is that syntactic methods are influenced by popularity because they depend on the number of connections between attribute and item nodes to select explanations. Conversely, embedding-based methods utilize vector representation similarities, resulting in a more balanced and effective approach to selecting explanations.

5.3 Answer to RQ3: What is the impact of calibration strategies on the explanations of recommendation systems?

To address RQ3, we experimented with the combination of recommender algorithms, category calibration, and explanation algorithms. We used five recommender algorithms, the Most Popular, BPR-MF, UserKNN, EASE, and NCF. For category calibration, we adopted the approach proposed by Steck [2018], and considered genre and award-received from KG properties. To generate the explanations, we applied three syntactic explanation algorithms: ExpLOD [Musto *et al*., 2016], ExpLOD v2 [Musto *et al*., 2019], and PEM [Du *et al*., 2022], and ComplEX algorithm as the most representative from embedding approaches. This setup allowed us to analyze recommendations in terms of relevance, fairness, and explainability.

Analyzing the MAP metric for recommendation relevance under calibration (see Figure 9) reveals that MAP scores varied depending on based on trade-off values and the category calibration applied. Among the evaluated recommended algorithms (Most Popular, BPR-MF, UserKNN, and NCF), MAP scores remained stable or even improved relative to the baseline across several trade-offs, indicating that these models can effectively incorporate calibration without significantly compromising relevance. In Contrast, the EASE recommender algorithm did not show the same positive behavior, regardless of the category calibration used or trade-off values.

A similar trend is observed when analyzing the MRR metric, which measures the ranking position of the first relevant recommendation (see Figure 10). Consistent with MAP results, MRR scores for Most Popular, BPR-MF, UserKNN, and NCF remained stable or improved across multiple trade-offs and category calibrations. The strong correlation between MAP and MRR suggests that calibration primarily shifts ranking positions rather than disrupting the overall ranking order entirely.

In the case of EASE recommender algorithm, its MAP and MRR scores failed to outperform the baseline, indicating that EASE may be more sensitive to calibration constraints and have difficulty balancing relevance and fairness.

Moreover, the findings indicate that genre calibration has a minimal negative impact on relevance compared to award-received calibration and baseline.

Now, analyzing the MRMC metric, which measures calibration (see Figure 11), we observe that applying calibration with both genre and award received has a consistently positive impact across all recommender algorithms and tradeoff values. This result indicates that calibration effectively aligns recommendations with user profiles by reducing the divergence between the recommended items and user preferences.

Overall, these findings underscore the importance of incorporating calibration mechanisms in recommendation systems to ensure that recommendations remain both relevant and fair for users. The consistent reduction in MRMC values across all models suggests that calibration is an effective tool for mitigating recommendation biases, improving user personalization and increasing user confidence in the system.

To understand how calibration affects explanation quality, we evaluated its impact on five recommender algorithms (Most Popular, BPR-MF, UserKNN, EASE, and NCF). We measured three explanation quality metrics (LIR, ETD, and SEP), and compared results between the baselines and calibrated versions (using genre and award-received calibration) for four explanation algorithms: ExpLOD, ExpLOD v2, PEM, and ComplEX.

Figure 4 presents the results for the Most Popular recommender. For LIR metric, no calibrated version outperformed the baseline. In contrast, the ETD metric shows in PEM algorithm with both genre and award-received calibration a better result than the baseline. Additionally, ExpLOD v2 and ComplEX showed improvements with genre calibration. Regarding the SEP metric, ExpLOD outperformed the baseline with award-received calibration, while both ExpLOD v2 and ComplEX outperformed the baseline when calibrated using either genre or award-received category.

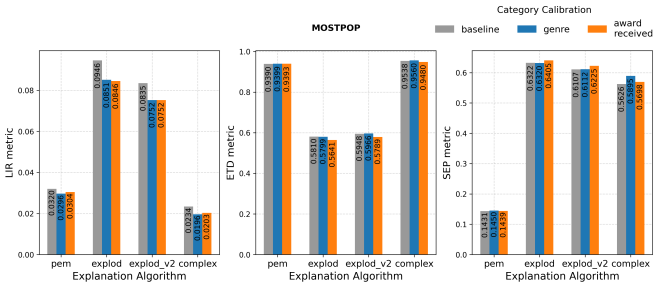


Figure 4. Measurement of Explanation Quality with Calibration – Results for the Most Popular Recommender Algorithm

For BPR-MF (see Figure 5), no calibrated versions surpassed the baseline in the LIR. However, the ETD metric reveals in PEM, ExpLOD, and ExpLOD v2 algorithms with genre calibration outperformed baseline, and ComplEX with both genre and award-received calibration outperformed baseline. In terms of SEP metric, the PEM algorithm showed that calibration leads to notable improvement with both genre and award-received, while ExpLOD and ExpLOD v2 performed well with award-received calibration, and ComplEX with genre calibration presented a significant improvement over the baseline.

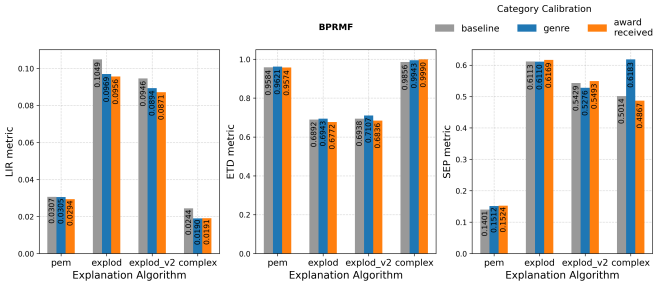


Figure 5. Measurement of Explanation Quality with Calibration – Results for the BPR-MF Recommender Algorithm

For the UserKNN recommender (see Figure 6), no calibrated version surpassed the baseline in LIR metric. However, for the ETD metric, PEM with genre calibration outperformed the baseline, as do ExpLOD and ExpLOD v2 when calibrated with genre. Furthermore, ComplEX outperformed the baseline with genre and award-received calibration. Regarding the SEP metric, both PEM and ExpLOD v2 showed improvements over the baseline with calibration based on both genre and award-received calibration.

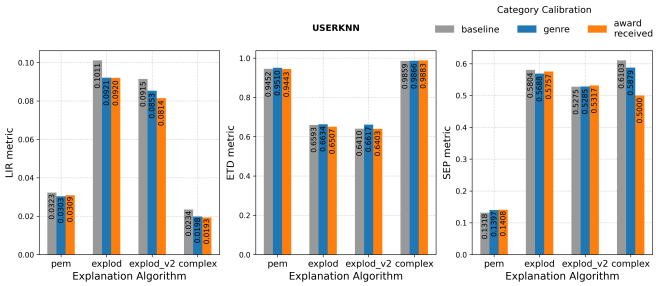


Figure 6. Measurement of Explanation Quality with Calibration – Results for the UserKNN Recommender Algorithm

For EASE (see Figure 7), CompLEX with genre calibration outperformed the baseline in the LIR metric. In the ETD metric, both PEM and ComplEX outperformed the baseline using either genre and award-received calibration, while ExpLOD v2 outperformed the baseline with genre calibration. In the SEP metric, PEM with award-received calibration outperformed the baseline, ExpLOD showed improvements with both genre and award-received calibration, and ComplEX with genre calibration also surpassed the baseline.

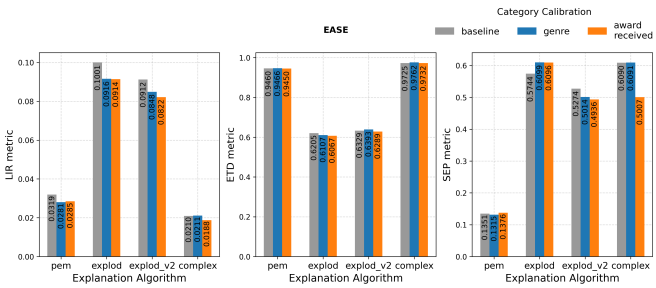


Figure 7. Measurement of Explanation Quality with Calibration – Results for the EASE Recommender Algorithm

Finally, for NCF (see Figure 8), no calibrated version surpassed the baseline in LIR. However, for ETD metric, PEM with genre calibration outperformed the baseline, and ExpLOD v2 surpassed the baseline with both genre and award-based calibration. Additionally, ComplEX outperformed the baseline with award-received calibration. In the SEP, PEM showed improvements over the baseline with both genre and award-received calibration, and ComplEX also outperformed the baseline with award-received calibration.

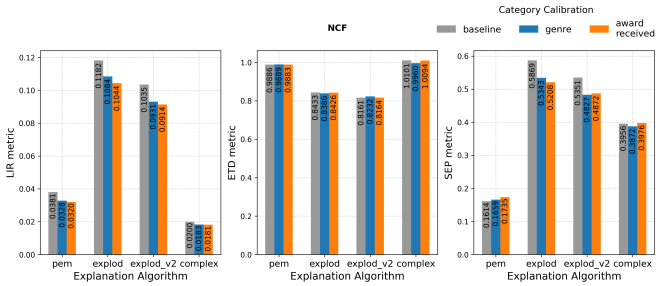


Figure 8. Measurement of Explanation Quality with Calibration – Results for the NCF Recommender Algorithm

Overall, these results suggest that while calibration does not improve the LIR metric (which reflects the recency of user interactions) across most recommender algorithms, it does improve explanation quality by promoting Diversity (ETD) and relevance (SEP).

Our findings for RQ3 indicate that category calibration enables recommendation systems to achieve two benefits: maintaining relevance and aligning recommendations with user preferences, while also enhancing explanation quality by increasing diversity and relevance. This improvement acts as a mechanism to mitigate bias, boost personalization, and finally foster user trust in the recommender system.

In response to RQ3, our findings indicate that category calibration significantly reduces bias while preserving recommendation relevance, although some recommender systems (such as EASE) appear more sensitive to calibration constraints in terms of relevance. Overall, calibration enhances explanation quality by boosting both diversity and relevance, thereby improving personalization and mitigating bias.

6 Limitations

The evaluation approach introduced in this study systematically compares explanation strategies and quantifies the impact of calibration in explanation quality, yet several limitations warrant discussion.

First, the method relies on a knowledge graph derived from Wikidata and implicitly assumes that this graph is both accurate and complete. In practice, Wikidata can exhibit sparsity, outdated links, or noise, which may lower the quality of generated explanations, particularly for items or attributes with limited or inconsistent representation.

Second, explanation quality is assessed exclusively through offline metrics. Although these metrics facilitate controlled, head-to-head comparisons, they do not necessarily reflect how real users perceive relevance or interpretability. Incorporating user studies would therefore provide valuable complementary insight.

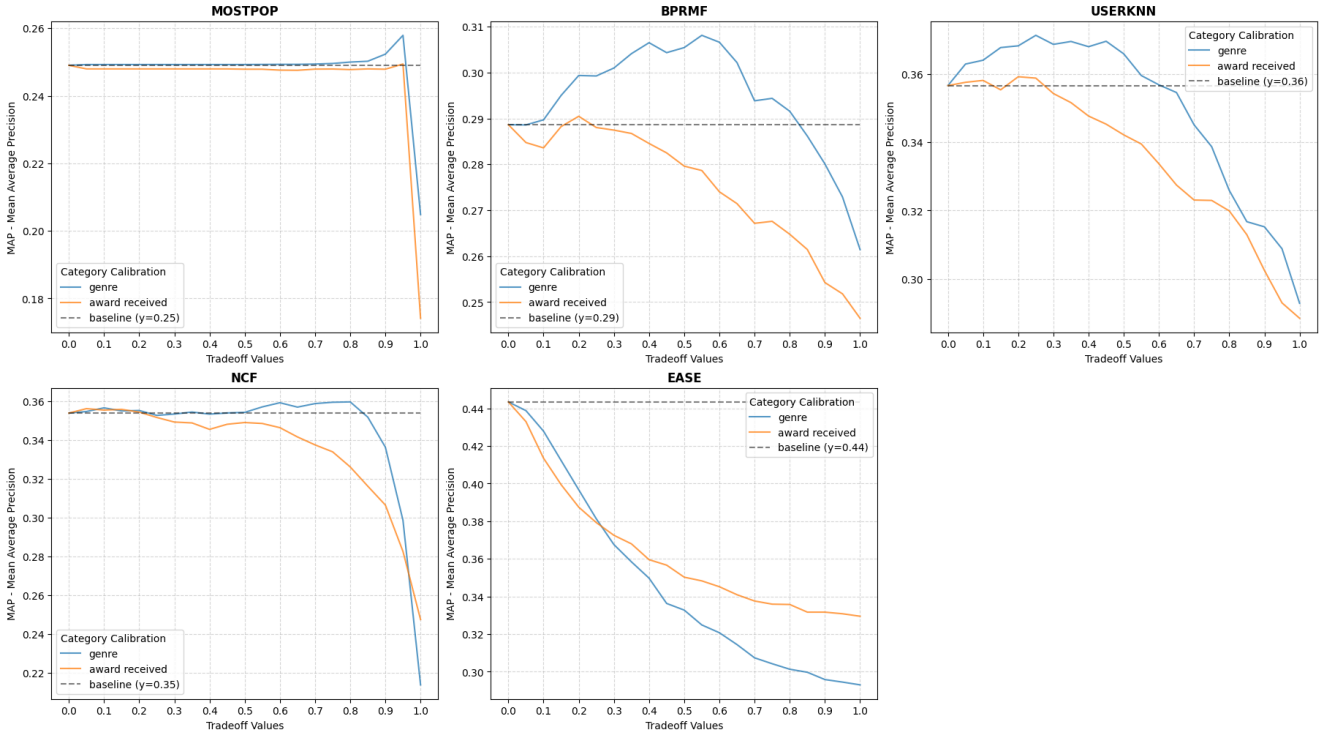


Figure 9. MAP results for recommender algorithms (Most Popular, BPR-MF, UserKNN, NCF, and EASE), comparing the baseline (uncalibrated) and calibrated versions across different trade-offs using genre and award-received categories

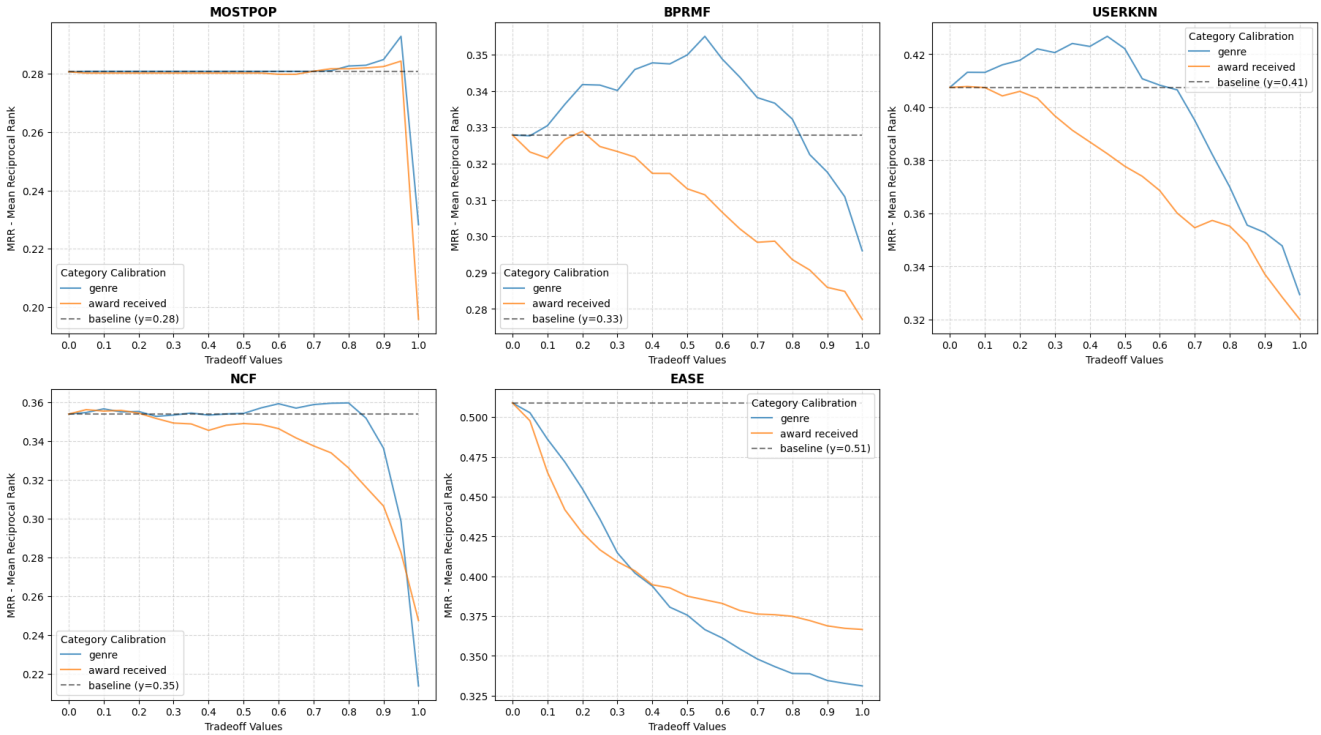


Figure 10. MRR results for recommender algorithms (Most Popular, BPR-MF, UserKNN, NCF, and EASE), comparing the baseline (uncalibrated) and calibrated versions under different trade-offs, considering genre and award-received categories

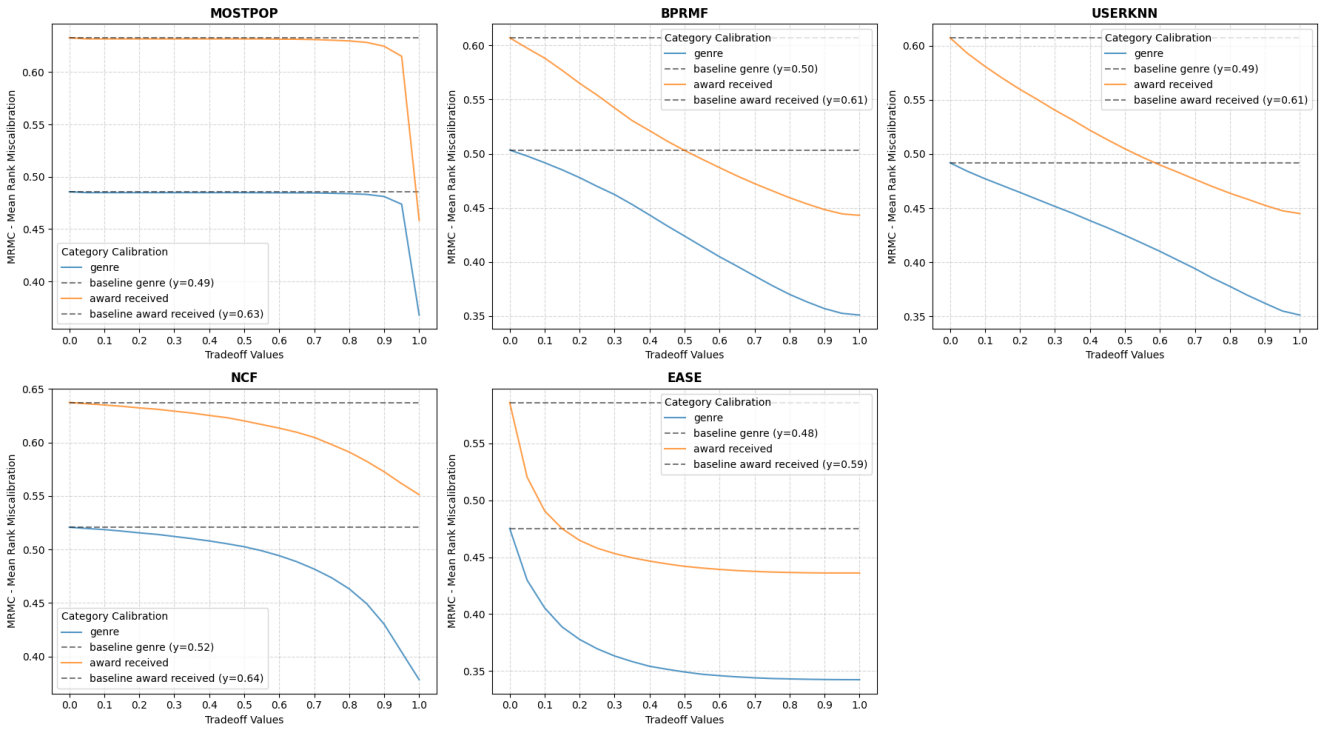


Figure 11. MRMC results for recommender algorithms (Most Popular, BPR-MF, UserKNN, NCF, and EASE) are compared between the baseline (uncalibrated) and calibrated versions across different trade-offs, considering genre and award-received categories

Third, the experiments are conducted on two benchmark datasets (MovieLens and LastFM) and a limited set of knowledge-graph attributes. Extending the evaluation to additional datasets and domains would strengthen the external validity of the findings.

Finally, the embedding models employed in the study were trained on static snapshots of the knowledge graph. When the graph evolves rapidly, frequent retraining may become necessary, which can impede scalability. Addressing this limitation represents a promising avenue for future research.

7 Conclusions

This work presents a comparative and reproducible approach to analyzing the impact of different graph embedding algorithms on generating quality explanations in model-agnostic recommendation systems (RS) using knowledge graphs (KGs). By employing three explainability metrics—the recency of items and the diversity and popularity of attributes—each algorithm's explanations were evaluated across two datasets and six recommendation systems. The findings indicate that embedding-based approaches better balance the popularity and diversity of attributes compared to syntactic approaches. Additionally, training metrics of various embedding methods do not necessarily correlate with improvements in explanation quality metrics.

In addition, we also investigated the effect of calibration to explanation in recommender systems. We employed the method proposed by Steck [2018] to align the user preferences of interacted and recommended items based on two KG attributes: genres and awards received. The results revealed that explanations were robust to the changes in the recommendation list made by the calibration algorithm. As a result, calibration strategies can be used along with explanation algorithms in recommender systems.

For future work, we will seek to investigate if generative models could assist in multi-parameter optimization and incorporate all three explanation quality metrics. This is important because syntactic explanation algorithms typically prioritize either attribute popularity or diversity while embedding models do not emphasize the recency of items. In addition, we can also investigate the perception of explanation to calibrated recommendations under the user perspective in order to further understand the effects of stacked post-processing steps with a user centric-evaluation.

Declarations

Acknowledgements

The authors acknowledge CAPES, CNPq, Fapesp, AWS, and Fapemig for their funding and support of this research. This publication also 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. ChatGPT was used for paper proofreading.

Funding

This research was funded by CAPES, CNPq, Fapesp, AWS, and Fapemig

Authors' Contributions

PA: Formal analysis, Investigation, Methodology, Software, Resources, Validation, Visualization, Writing - original draft AZ: Conceptualization, Data curation, Formal analysis, Investigation,

Methodology, Software, Resources, Validation, Visualization, Writing - original draft LR and MM: Conceptualization, Formal Analysis, Funding acquisition, Methodology, Project administration, Supervision, Writing – review & editing. All authors read and approved the final manuscript.

Competing interests

The authors declare they have no competing interests.

Availability of data and materials

The datasets generated and/or analyzed during the current study are available in `https://github.com/andlzanon/lod-personalized-recommender` where the development for RQ1 and RQ2 were developed. To answer RQ3 regarding the effect of calibration in explanations the results and implementation are in the repository `https://gitlab.com/paul.atauchi/lod-personalized-recommender/-/tree/effect_calibration_explanation`.

Appendix 1 - Calibration Trade-Offs

Table 5. Configurations of calibration trade-offs with high score for MAP, MRR and MRMC combination metrics.

Recommender algorithm - Category	Score	Trade-off
Most Popular - Genre	0.027019	0.95
Most Popular - Award Received	0.012894	0.95
BPR-MF - Genre	0.091179	0.55
BPR-MF - Award Received	0.030948	0.45
UserKNN - Genre	0.062192	0.45
UserKNN - Award Received	0.026600	0.25
NCF - Genre	0.040256	0.80
NCF - Award Received	0.005608	0.15
EASE - Genre	0.011905	0.05
EASE - Award Received	0.011113	0.05

Appendix 2 - Recommender Systems Ranking Metrics

Table 6. Top-5 recommendation ranking metrics for MovieLens datasets on the 90/10 split used to extract the explanation quality metrics. **Bold** values are the highest for a dataset. The metrics used were: Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) are ranking accuracy metrics and Aggregate Diversity (AGG-DIV), Gini Index (Gini), Entropy and Catalog Coverage (Coverage) are beyond accuracy metrics.

Metric	MostPop	BPR-MF	PageRank	UserKNN	EASE	NCF
NDCG	0.31702	0.384273	0.402638	0.46454	**0.556552**	0.43216
MAP	0.254989	0.296985	0.318773	0.374087	**0.47013**	0.33274
AGG-DIV	46	**428**	131	248	317	-
Gini	0.99864	**0.98342**	0.99758	0.99088	0.98905	-
Entropy	1.18239	**2.26747**	1.4077	2.0344	2.1021	-
Coverage	0.00502	**0.04672**	0.01430	0.02707	0.03461	-

Table 7. Top-5 recommendation ranking metrics for LastFM datasets on the 90/10 split used to extract the explanation quality metrics. **Bold** values are the highest for a dataset. The metrics used were: Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) are ranking accuracy metrics and Aggregate Diversity (AGG-DIV), Gini Index (Gini), Entropy and Catalog Coverage (Coverage) are beyond accuracy metrics.

Metric	MostPop	BPRMF	PageRank	UserKNN	EASE	NCF
NDCG	0.15498	0.268951	0.274804	0.367126	**0.380198**	0.355325
MAP	0.11344	0.201658	0.220805	0.292648	**0.312495**	0.271229
AGG-DIV	20	**402**	191	401	386	-
Gini	0.99920	**0.99269**	0.99805	0.99190	0.99293	-
Entropy	0.98064	**1.98878**	1.39999	2.05101	1.99260	-
Coverage	0.00180	**0.03633**	0.01726	0.03624	0.034891	-

Appendix 3 - Explanations Examples

Table 8. Explanations of the top-5 items recommended by the EASE algorithm for each explanation algorithm. The symbol -> represents an edge and; indicates that multiple items are related to the attribute in sequence.

	TransE
1	The Terminator ->Terminator ->Terminator 2 Judgment Day
2	Alien ->Alien ->Aliens
3	The Green Mile ->United States of America ->Men in Black
4	Hook ->United States of America ->Die Hard
5	JFK ->Washington, D.C. ->True Lies
	Rotate
1	American Beauty ->drama ->Terminator 2 Judgment Day
2	Alien ->Alien ->Aliens
3	Toys ->comedy film ->Men in Black
4	L.A. Confidential ->film based on a novel ->Die Hard
5	Toys ->comedy film ->True Lies
	ComplEX
1	Crocodile Dundee ->Australia ->The Thin Red Line ->Academy Award for Best Sound Mixing ->Terminator 2 Judgment Day
2	Crocodile Dundee ->Australia ->The Thin Red Line ->20th Century Studios ->Aliens
3	Dances with Wolves ->United States of America ->Men in Black
4	Crocodile Dundee ->Australia ->Welcome to Woop Woop ->20th Century Studios ->Die Hard
5	Mission Impossible ->London ->Lock, Stock and Two Smoking Barrels ->MTV Movie Award for Best Action Sequence ->True Lies
	ExpLOD
1	Independence Day; Platoon; Schindler's List ->Academy Award for Best Sound Mixing ->Terminator 2 Judgment Day
2	Independence Day; Platoon; Schindler's List ->Academy Award for Best Sound Mixing ->Aliens
3	Independence Day; Rob Roy; Henry V ->action film ->Men in Black
4	Independence Day; Platoon; Schindler's List ->Academy Award for Best Sound Mixing ->Die Hard
5	Independence Day; Ghostbusters; Batman Returns ->Academy Award for Best Visual Effects ->True Lies
	ExpLOD v2
1	Independence Day; Seven; The Silence of the Lambs ->thriller film ->Terminator 2 Judgment Day
2	Independence Day; Seven; The Silence of the Lambs ->thriller film ->Aliens
3	Independence Day; Rob Roy; Henry V ->action film ->Men in Black
4	Independence Day; Rob Roy; Henry V ->action film ->Die Hard
5	Independence Day; Seven; The Silence of the Lambs ->thriller film ->True Lies
	PEM
1	The Terminator; The Abyss; The 13th Warrior ->William Wisher ->Terminator 2 Judgment Day
2	Independence Day; Back to the Future; Groundhog Day ->Hugo Award for Best Dramatic Presentation ->Aliens
3	Howard the Duck; Indiana Jones and the Last Crusade; Star Wars Episode V – The Empire Strikes Back ->George Lucas ->Men in Black
4	Tombstone; Basic Instinct; RoboCop ->Frank J. Urioste ->Die Hard
5	Groundhog Day; The Wedding Singer; Wayne's World ->MTV Movie Award for Best Comedic Performance ->True Lies

References

Abdollahpouri, H., Mansoury, M., Burke, R., and Mobasher, B. (2020). The connection between popularity bias, calibration, and fairness in recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 726–731, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3383313.3418487.

Aggarwal, C. C. (2016). *An Introduction to Recommender Systems*, pages 1–28. Springer International Publishing, Cham. DOI: https://doi.org/10.1007/978-3-319-29659-3_1.

Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., Sharifzadeh, S., Tresp, V., and Lehmann, J. (2021). Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings. *Journal of Machine Learning Research*, 22(82):1–6.

Alves, G., Jannach, D., Ferrari De Souza, R., and Manzato, M. G. (2024). User perception of fairness-calibrated recommendations. In *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '24, page 78–88, New York,

NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3627043.3659558.

Balloccu, G., Boratto, L., Fenu, G., and Marras, M. (2022). 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*, SIGIR '22, page 646–656, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3477495.3532041.

Balog, K. and Radlinski, F. (2020). Measuring recommendation explanation quality: The conflicting goals of explanations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 329–338, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3397271.3401032.

Cantador, I., Brusilovsky, P., and Kuflik, T. (2011). Hetrec '11: Proceedings of the 2nd international workshop on information heterogeneity and fusion in recommender systems. New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/2043932.2044016.

Cao, J., Fang, J., Meng, Z., and Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Comput. Surv.*, 56(6). DOI: https://doi.org/10.1145/3643806.

Coba, L., Confalonieri, R., and Zanker, M. (2022). Recoxplainer: A library for development and offline evaluation of explainable recommender systems. *IEEE Computational Intelligence Magazine*, 17(1):46–58. DOI: https://doi.org/10.1109/MCI.2021.3129958.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, page 39–46, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/1864708.1864721.

da Costa, A., Fressato, E., Neto, F., Manzato, M., and Campello, R. (2018). Case recommender: a flexible and extensible python framework for recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 494–495, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3240323.3241611.

da Silva, D. C. and Durão, F. A. (2023). Introducing a framework and a decision protocol to calibrated recommender systems. *Applied Intelligence*, 53(19):22044–22072. DOI: https://doi.org/10.1007/s10489-023-04681-7.

da Silva, D. C. and Durão, F. A. (2025). Benchmarking fairness measures for calibrated recommendation systems on movies domain. *Expert Systems with Applications*, page 126380. DOI: https://doi.org/10.1016/j.eswa.2025.126380.

da Silva, D. C., Manzato, M. G., and Durão, F. A. (2021). Exploiting personalized calibration and metrics for fairness recommendation. *Expert Systems with Applications*, 181:115112. DOI: https://doi.org/10.1016/j.eswa.2021.115112.

Dijkstra, E. W. (2022). *A Note on Two Problems in Connexion with Graphs*, page 287–290. Association for Computing Machinery, New York, NY, USA, 1 edition. DOI: https://doi.org/10.1145/3544585.3544600.

Du, Y., Ranwez, S., Sutton-Charani, N., and Ranwez, V. (2022). Post-hoc recommendation explanations through an efficient exploitation of the dbpedia category hierarchy. *Knowledge-Based Systems*, 245:108560. DOI: https://doi.org/10.1016/j.knosys.2022.108560.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Ferrari Dacrema, M., Boglio, S., Cremonesi, P., and Jannach, D. (2021). A troubling analysis of reproducibility and progress in recommender systems research. *ACM Trans. Inf. Syst.*, 39(2). DOI: https://doi.org/10.1145/3434185.

Ferraro, A. (2019). Music cold-start and long-tail recommendation: bias in deep representations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 586–590, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3298689.3347052.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., and He, Q. (2022). A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568. DOI: https://doi.org/10.1109/TKDE.2020.3028705.

Hada, D. V., M., V., and Shevade, S. K. (2021). Rexplug: Explainable recommendation using plug-and-play language model. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 81–91, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3404835.3462939.

Harper, F. M. and Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4). DOI: https://doi.org/10.1145/2827872.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 173–182, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee. DOI: https://doi.org/10.1145/3038912.3052569.

Kaminskas, M. and Bridge, D. (2016). Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7(1). DOI: https://doi.org/10.1145/2926720.

Li, J. and Yang, Y. (2022). Star: Knowledge graph embedding by scaling, translation and rotation. In *International Conference on AI and Mobile Services*, pages 31–45. Springer. DOI: https://doi.org/10.1007/978-3-031-23504-7_3.

Lin, K., Sonboli, N., Mobasher, B., and Burke, R. (2020). Calibration in collaborative filtering recommender systems: a user-centered analysis. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, HT '20, page 197–206, New York,

NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3372923.3404793.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29. DOI: https://doi.org/10.1609/aaai.v29i1.9491.

Musto, C., Narducci, F., Lops, P., De Gemmis, M., and Semeraro, G. (2016). Explod: A framework for explaining recommendations based on the linked open data cloud. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 151–154, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/2959100.2959173.

Musto, C., Narducci, F., Lops, P., de Gemmis, M., and Semeraro, G. (2019). Linked open data-based explanations for transparent recommender systems. *International Journal of Human-Computer Studies*, 121:93–107. DOI: https://doi.org/10.1016/j.ijhcs.2018.03.003.

Naghiaei, M., Dehghan, M., Rahmani, H. A., Azizi, J., and Aliannejadi, M. (2024). Personalized beyond-accuracy calibration in recommendation. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '24, page 107–116, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3664190.3672507.

Paulheim, H. (2016). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508. DOI: https://doi.org/10.3233/SW-160218.

Peng, C., Xia, F., Naseriparsa, M., and Osborne, F. (2023). Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, pages 1–32. DOI: https://doi.org/10.1007/s10462-023-10465-9.

Rana, A., D'Addio, R. M., Manzato, M. G., and Bridge, D. (2022). Extended recommendation-by-explanation. *User Modeling and User-Adapted Interaction*, 32(1-2):91–131. DOI: https://doi.org/10.1007/s11257-021-09317-4.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 452–461, Arlington, Virginia, USA. AUAI Press. DOI: https://doi.org/10.48550/arXiv.1205.2618.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, page 175–186, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/192844.192905.

Ricci, F., Rokach, L., and Shapira, B. (2022). *Recommender Systems: Techniques, Applications, and Challenges*, pages 1–35. Springer US, New York, NY. DOI: https://doi.org/10.1007/978-1-0716-2197-4_1.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215. DOI: https://doi.org/10.1038/s42256-019-0048-x.

Souza, L. S. d. and Manzato, M. G. (2022). Aspect-based summarization: An approach with different levels of details to explain recommendations. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, WebMedia '22, page 202–210, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3539637.3557002.

Souza, R. and Manzato, M. (2024a). Uma abordagem em etapa de processamento para redução do viés de popularidade. In *Proceedings of the 30th Brazilian Symposium on Multimedia and the Web*, pages 310–317, Porto Alegre, RS, Brasil. SBC. DOI: https://doi.org/10.5753/webmedia.2024.241542.

Souza, R. and Manzato, M. (2024b). Uma abordagem em etapa de processamento para redução do viés de popularidade. In *Proceedings of the 30th Brazilian Symposium on Multimedia and the Web*, pages 310–317, Porto Alegre, RS, Brasil. SBC. DOI: https://doi.org/10.5753/webmedia.2024.241542.

Steck, H. (2018). Calibrated recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 154–162, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3240323.3240372.

Steck, H. (2019). Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, WWW '19, page 3251–3257, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3308558.3313710.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*. DOI: https://doi.org/10.48550/arXiv.1902.10197.

Tchuente, D., Lonlac, J., and Kamsu-Foguem, B. (2024). A methodological and theoretical framework for implementing explainable artificial intelligence (xai) in business applications. *Computers in Industry*, 155:104044. DOI: https://doi.org/10.1016/j.compind.2023.104044.

Tintarev, N. and Masthoff, J. (2015). Explaining recommendations: Design and evaluation. In *Recommender systems handbook*, pages 353–382. Springer. DOI: 10.1007/978-1-4899-7637-6.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA. PMLR. DOI: https://doi.org/10.5555/3045390.3045609.

Wang, S., Zhang, X., Wang, Y., and Ricci, F. (2024). Trustworthy recommender systems. *ACM Trans. Intell. Syst. Technol.*, 15(4). DOI: https://doi.org/10.1145/3627826.

Xu, Z., Zeng, H., Tan, J., Fu, Z., Zhang, Y., and Ai, Q. (2023). A reusable model-agnostic framework for faithfully explainable recommendation and system scrutability. *ACM Trans. Inf. Syst.*, 42(1). DOI: https://doi.org/10.1145/3605357.

Zanon, A. L., da Rocha, L. C. D., and Manzato, M. G. (2022). Balancing the trade-off between accu-

racy and diversity in recommender systems with personalized explanations based on linked open data. *Knowledge-Based Systems*, 252:109333. DOI: https://doi.org/10.1016/j.knosys.2022.109333.

Zanon, A. L., da Rocha, L. C. D., and Manzato, M. G. (2024). Model-agnostic knowledge graph embedding explanations for recommender systems. In *World Conference on Explainable Artificial Intelligence*, pages 3–27. Springer. DOI: https://doi.org/10.1007/978-3-031-63797-1_1.

Zhang, S., Tay, Y., Yao, L., and Liu, Q. (2019). Quaternion knowledge graph embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.. DOI: https://doi.org/10.5555/3454287.3454533.

Zhang, Y. and Chen, X. (2020). Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101. DOI: https://doi.org/10.1561/1500000066.