






# How to Surprisingly Consider Recommendations? A Knowledge-Graph-Based Approach Relying on Complex Network Metrics

Oliver Baumann<sup>1</sup><sup>a</sup>, Durgesh Nandini<sup>1</sup><sup>b</sup>, Anderson Rossanez<sup>2</sup><sup>c</sup>, Mirco Schoenfeld<sup>1</sup><sup>d</sup>  
and Julio Cesar dos Reis<sup>2</sup><sup>e</sup>

<sup>1</sup>University of Bayreuth, Bayreuth, Germany

<sup>2</sup>Institute of Computing, University of Campinas, Campinas, SP, Brazil

{oliver.baumann, durgesh.nandini, mirco.schoenfeld}@uni-bayreuth.de, {anderson.rossanez, jreis}@ic.unicamp.br

**Keywords:** Recommender Systems, Knowledge Graphs, Complex Network Metrics.


**Abstract:** Traditional recommendation proposals, including content-based and collaborative filtering, usually focus on similarity between items or users. Existing approaches lack ways of introducing unexpectedness into recommendations, prioritizing globally popular items over exposing users to unforeseen items. This investigation aims to design and evaluate a novel layer on top of recommender systems suited to incorporate relational information and rerank items with a user-defined degree of surprise. Surprise in recommender systems refers to the degree to which a recommendation deviates from the user's expectations, providing an unexpected yet reliable recommendation. We propose a knowledge graph-based recommender system by encoding user interactions on item catalogs. Our study explores whether network-level metrics on knowledge graphs (KGs) can influence the degree of surprise in recommendations. We hypothesize that surprisingness correlates with specific network metrics, treating user profiles as subgraphs within a larger catalog KG. The achieved solution reranks recommendations based on their impact on structural graph metrics. Our research contributes to optimizing recommendations to reflect the network-based metrics. We experimentally evaluate our approach on two datasets of LastFM listening histories and synthetic Netflix viewing profiles. We find that reranking items based on complex network metrics leads to a more unexpected and surprising composition of recommendation lists.


## 1 INTRODUCTION


Recommender Systems aim to offer a personalized view of large complex spaces, prioritizing items likely to interest the user by analyzing user preferences, historical behavior, and item characteristics (Felfernig and Burke, 2008). Recommendations can expose users to relevant items and expand their understanding of the catalog, regardless of whether in an e-commerce, media-streaming, or GLAM setting. The most popular approaches for recommender systems (RS) are collaborative filtering and content-based filtering (Schafer et al., 2007; Sarwar et al., 2001).


User-item recommendations are an important part of the discovery process of large collections. In content-based filtering, item characteristics are used to determine the similarity between items rated (viewed, listened, bought, etc.) by a user, and “unseen” items. Collaborative filtering, on the other hand, determines users similar to the target user and predicts ratings on unseen items by the target user.


Existing approaches have been shown to produce meaningful recommendations; the items they recommend tend to be expected and located in whatever portion of the catalog considered “mainstream”. These approaches do not consider the rich relations between items beyond the realm of similarity alone. We argue that users may profit from recommendations that include an element of surprise, as they may come in touch with concepts they have been unaware of. State-of-the-art commonly operationalizes surprise through auxiliary constructs such as novelty and diversity (Kaminskas and Bridge, 2016; Castells

<sup>a</sup> <https://orcid.org/0000-0003-4919-9033>

<sup>b</sup> <https://orcid.org/0000-0002-9416-8554>

<sup>c</sup> <https://orcid.org/0000-0001-7103-4281>

<sup>d</sup> <https://orcid.org/0000-0002-2843-3137>

<sup>e</sup> <https://orcid.org/0000-0002-9545-2098>

All authors contributed equally.

et al., 2021). We provide closed-form definitions for these terms in Section 4.3. We define “surprise” as the degree to which a recommendation deviates from the user’s expectations, introducing unexpectedness into the recommended items list while maintaining relevance to the user’s interests and preferences.

Knowledge Graph RS combine the capabilities of recommender systems and Knowledge Graphs (KGs) by incorporating and analyzing the structured representation of information in KGs. These systems leverage the interconnected nature of entities and their attributes within the KG to enhance the accuracy and relevance of recommendations. Using KGs, recommender systems can go beyond simple user-item interactions and incorporate a broader understanding of the relationships among items, users, and other entities. This allows for more sophisticated recommendation approaches that consider not only the user’s preferences. In this sense, contextual information encoded in KGs influences recommendation items. For example, in a movie recommendation scenario, a KG-based RS could consider not only the user’s past viewing history and ratings. It can consider, for instance, the genre of the movie, the actors and directors involved, and the relationships between movies based on shared themes or motifs.

In this study, we propose a layer on top of recommender systems, extending their functionality by a configurable degree of surprise. Our approach considers relational information among items encoded in KGs and suggests items with a user-defined degree of surprise relying on results generated by a recommender system. The main research question guiding our investigation is whether network metrics computed on the KG influence the degree of surprise within the recommendations. We propose leveraging the graph structure of KGs, employing complex network measurements (Rossanez et al., 2023) to encode entity relevance in a KG. Centrality measurements denote different meanings of relevance for graph nodes, bringing novelty aspects for analyses over KGs. Our assumption highlights that the “surprisingness” of recommendations is reflected in the network-level metrics of the KG, which provide a means to evaluate structural changes in KGs when recommendations are included.

Figure 1 provides a high-level overview of our approach. We construct KGs from two distinct catalogs: users’ listening events on the platform LastFM<sup>1</sup>, and TV shows and movies on Netflix<sup>2</sup>. User profiles for LastFM are available through the LFM-1b dataset (Schedl, 2016); for Netflix, we generate syn-

<sup>1</sup><https://www.last.fm/>

<sup>2</sup><https://www.netflix.com>

thetic profiles. Recommendations for these profiles are then generated through state-of-the-art recommender systems. Our work supports any RS, as we focus on reranking recommendations to surface surprising results. Consequently, a specific RS optimal for a particular use case can be selected. For each user profile, we determine the induced subgraph on the catalog-KG that includes all items the user interacted with and further entities that enrich the model. Then, for each user and each item in their recommendation list, we assess the impact of including that item and its KG-informed neighborhood on the user’s subgraph through pre-determined graph metrics. The original recommendation lists are then re-ranked according to their relative impact.

Our contributions are summarized as follows:

- Insert a configurable level of surprise to any recommender system by adding a layer of meta-analysis on obtained recommendations;
- Identify a network metric that correlates with different dimensions of surprise;
- Provide a comparative study regarding several network-level metrics for reranking recommendation results;

The remainder of this article is organized as follows: Section 2 discusses related work. Section 3 presents our proposal. Section 4 reports our experimental evaluation and its results. Section 5 discusses our findings. Section 6 wraps up our investigation and points out directions for future studies.

## 2 RELATED WORK

Joseph & Jiang (Joseph and Jiang, 2019) proposed a graph traversal algorithm along with a novel weighting scheme for cold-start content-based recommendation using named entities. Their approach computes the shortest distance between named entities over large KGs. Wang *et al.* (Wang et al., 2019) introduced the KG Attention Network (*KGAT*), which enhances the effectiveness of collaborative filtering in RS by effectively modeling the high-order connectivity between users, items, and entities within a KG. Their research investigated how different levels of connectivity, first-order, second-order, third-order, etc. impact the model’s effectiveness. They discussed the findings of using attention mechanisms and KG embeddings.

Hui *et al.* (Hui et al., 2022) presented *ReBKC*, an RS that uses auxiliary information such as historical user behavior and KGs to provide personalized

suggestions. Their investigation integrates KG embeddings and user-item interactions to address issues like sparse data and cold start. ReBKC suggests using KGs as heterogeneous networks to incorporate additional information to unify embeddings of user behavior and knowledge features. Their proposed algorithm employs collaborative filtering, enhanced by the rich semantic associations in KGs, to mine user preferences more deeply. The system learns from historical user interactions and multiple relationships within the KG.

Zhang *et al.* (Zhang et al., 2016) addressed the limitations of collaborative filtering in recommender systems by leveraging heterogeneous information in a knowledge base to improve the quality of recommendations. Their proposed framework – Collaborative Knowledge Base Embedding (*CKE*) – comprises three components to extract semantic representations from items’ structural, textual, and visual content. These components employ techniques such as heterogeneous network embedding, stacked denoising auto-encoders, and convolutional auto-encoders to extract textual and visual representations. It then jointly learns the latent representations in collaborative filtering and items’ semantic representations from the knowledge base. Kaminskas and Bridge (Kaminskas and Bridge, 2016) looked into the aspects of diversity, serendipity, novelty, and coverage. They explained that introducing surprise in RS can burst the “user filter bubble” by finding interesting items that the user might not have otherwise discovered.

Kotkov *et al.* (Kotkov et al., 2016) examined the concept of serendipity in the context of recommender systems. Their work discussed different approaches to measure and enhance serendipity in RS, including using algorithms that utilize uncommon similarity measures or adapt based on user feedback. Their investigation looked at the balance between accuracy and novelty in recommendations and explored offline and online evaluation strategies for assessing the effectiveness of RS in delivering serendipitous results. On the other hand, De Gemmis *et al.* (De Gemmis et al., 2015) proposed to produce serendipitous suggestions by utilizing the knowledge infusion process. Their investigation addressed the overspecialization issue in RS, proposing to enhance serendipity by suggesting surprising items. Their approach enriches a graph-based recommendation algorithm with background knowledge to uncover hidden correlations among items.

Baumann and Schoenfeld (Baumann and Schoenfeld, 2022) used a KG-based recommender system to evaluate recommendations’ diversity and novelty on a content- and network-level. Using subgraphs con-

structed from user profiles, they generated recommendations by favoring unpopular items in the catalog that exhibit a high distance from a user’s profile regarding content-based features. Apart from unexpectedness and diversity on a content level, they found this approach to result in a more fair degree distribution on the individual profile subgraphs.

There are some state-of-the-art approaches that address the problem of reranking recommended items, with their focus on bias mitigation or long-tailed problems. (Abdollahpouri et al., 2019) introduces a personalized diversification reranking approach to increase the representation of less popular items in recommendations and to address the problem of popularity bias. They achieve this by introducing a likelihood parameter that controls the popularity bias. (Liu et al., 2022) discusses reranking in multiple facets such as awareness, diversity, and edge reranking using neural networks. (Pei et al., 2019) propose a personalized reranking model for recommender systems by employing a self attention based transformer model that encodes information of all items in the list by modeling the global relationships between any pair of items in the entire list. However, to the best of our observation, none of the papers considered multiple metrics for reranking the recommendation list.

To the best of our knowledge, our present study is the first to apply complex network measurements to rerank the order of recommendation results. Our approach looks at the graph structure within the KG changes to compute the metrics for obtaining surprising recommendations.

### 3 KG-INFORMED RECOMMENDATION (RE-)RANKING

We propose a recommendation process as a two-step approach consisting of retrieval and ranking steps. In the retrieval step, recommendation candidates are determined by an existing RS. These candidates are ordered in the ranking step, and the top  $N$  elements are returned to the user. Figure 1 presents an overview of the proposed process.

This investigation treats the recommender system as a closed system over which we can not exert any influence. Our solution emphasizes and contributes to the ranking stage. We determine an item’s rank based on its impact on network metrics correlating with surprise; Section 4.2 presents a list of metrics investigated.

To evaluate such metrics, we construct KGs from

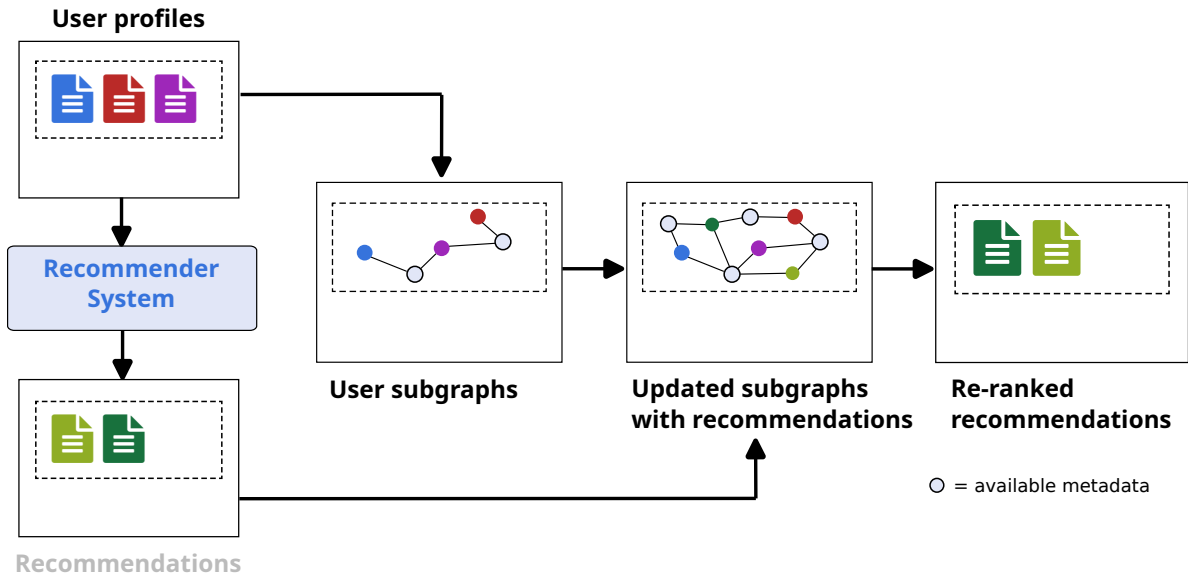


Figure 1: Overview of the proposed knowledge-graph informed recommendations. KGs are constructed for the item catalog and all user profiles. The latter serve as input to an arbitrary state-of-the-art RS, whose results are re-ranked according to the impact the items would have were they included in the original user profile.

datasets suited for the recommendation task (*cf.* Section 4.1). Two types of KGs are constructed. The first type is a KG representing the catalog, *i.e.*, containing the entire knowledge about the catalog. This includes the whole set of recommendable items and all the metadata describing them. The second type are user-profile KGs, which constitute subgraphs of the catalog KG and represent items users have already interacted with. These KGs are constructed based on TBox statements representing the domain of their datasets, *i.e.*, a conceptual model describing classes and properties that are aligned with the underlying domain. Therefore, it includes recommendable entities, additional entities, and heterogeneous relations.

The recommendable entities are evaluated by including them in the user-profile KGs. According to those existing in the catalog KG, a recommendable entity is included along with its relationships and further entities. From the updated user-profile KG, we compute complex network metrics (*cf.* Figure 2). The process is conducted for all the recommendable items and all available metrics. At the final stage, our solution provides a re-ranked recommendation list sorted according to each metric.

Where network metrics do not result in scalar values, but in distributions (*e.g.* betweenness), we calculate the Herfindahl-Hirshman-Index (HHI) (Hirschman, 1964) to obtain a single value representing the concentration of the network (*cf.* Schoenfeld and Pfeffer (Schoenfeld and Pfeffer, 2021)). Let  $s$  be the relative centrality score over all

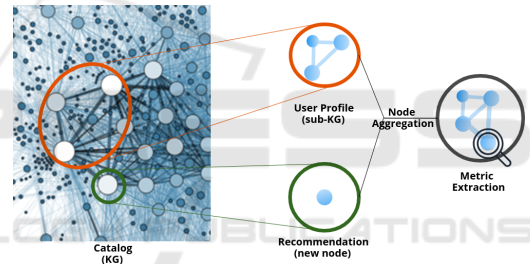


Figure 2: Obtaining KG-informed recommendations. The user profile is represented as a subgraph of the knowledge graph (sub-KG). A candidate recommendation node is selected from the catalog KG and integrated into the sub-KG along with relevant edges. Network metrics are then computed on the updated sub-KG.

vertices, and  $N$  the number of vertices, then the index and its normalized form are given by

$$HHI = \sum_{i=1}^N s^2 \quad (1)$$

$$HHI^* = \frac{HHI - 1/N}{1 - 1/N} \quad (2)$$

Values of  $HHI^*$  range in  $[0, 1]$ , with 0.0 corresponding to a balanced network with no monopolies and a value of 1.0 indicating a strongly centralized network.

Formally, let  $KG$  be a domain KG, consisting of concepts  $C$  and relations  $R$ . A user profile  $U = \{u_1, u_2, \dots, u_n\}$  is a subset  $U \subset C$  of concepts a user has interacted with. Each user profile constitutes an



induced subgraph  $SG \subset KG$  containing the history items and further concepts and relations.

For a recommender system  $RS$ , let  $I = \{i_1, i_2, \dots, i_n\}$  be the list of recommended concepts, ordered by a score assigned through  $RS$ . Then,  $N_{KG}[i_n]$  denotes the closed neighborhood of the vertex  $i_n$  in  $KG$  that corresponds to this recommendation. Let  $SG'$  denote the induced subgraph produced by including  $N_{KG}[i_n]$  in  $SG$ , *i.e.*, by adding a recommendation to the user subgraph.

Lastly, let  $m$  denote any graph-metric,  $m_{baseline} = m(SG)$  the value of this metric on the original user subgraph, and  $m_{update} = m(SG')$  its value after incorporating the recommendation in the subgraph. Then, our method for re-ranking recommendations is as follows:

1. Given a dataset, construct a KG  $KG$ ;
2. Given a user profile  $U$ , determine the subgraph  $SG$  of  $KG$  that contains all items in the user's profile and all relations and intermediate entities;
3. Given a set of items in  $U$ , determine recommendations using any RS;
4. For each recommended item  $i_n$ , obtain the metric  $m$  on the subgraph  $SG'$  including  $i_n$ ;
5. Re-rank all items according to their impact on the given computed metric;

Algorithm 1 provides a formal formulation of this approach.

## 4 EXPERIMENTAL EVALUATION

We evaluate our proposed approach on two distinct music- and movie-domain datasets. Proceeding according to our defined method (*cf.* Section 3), we obtain reranked recommendation lists for a set of users. As we focus our attention on “surprisingness” of recommendations, rather than measuring precision/accuracy, we turn to “beyond accuracy”-metrics commonly used in the evaluation of surprise and serendipity in RS, such as novelty and diversity (*cf.* (Castells et al., 2015; Ge et al., 2010)). The proposed system aims to introduce users to entirely new items that do not appear in their interaction history, thus rendering metrics such as precision and accuracy less meaningful.

To back up the insights obtained in this sense, we further measure the agreement of the re-ranked recommendation lists with those generated through the SOTA RS, which we treat as ground truth for “expectable” recommendations. Measuring normalized discounted cumulative gain (nDCG) on the two lists

```

Input:  $KG$  // Catalog KG
Input:  $SG$  // User profile subgraph
Input:  $RS$  // Recommender system
Input:  $m$  // A graph-metric
 $v_{user\_nodes} \leftarrow SG.get\_all\_nodes();$ 
 $I \leftarrow RS(v_{user\_nodes});$ 
 $I' \leftarrow \emptyset;$ 
foreach  $i \in I$  do
     $SG' \leftarrow SG.copy();$ 
     $SG'.add\_node(i);$ 
     $edge\_list \leftarrow \emptyset;$ 
    foreach  $neigh \in KG.get\_neighbors(i)$  do
        if  $neigh \in v_{user\_nodes}$  then
             $edges \leftarrow KG.get\_edges(i, neigh);$ 
             $edge\_list.insert(edges);$ 
        end
    end
     $SG'.add\_edges(edge\_list);$ 
     $metric\_value \leftarrow m(i, SG');$ 
     $I'.insert(metric\_value, i);$ 
end
 $recos \leftarrow sort(I', metric\_value);$ 
return  $recos;$ 

```

Algorithm 1: KG-informed recommendation.

allows us to identify whether the re-ranked variant deviates from the expectable recommendations.

Our study addresses the following specific research questions:

**RQ1.** Which network-level metrics correlate with key surprise elements such as novelty, unexpectedness, and novelty in recommendations?

**RQ2.** Can these metrics be used to introduce more surprise into state-of-the-art recommender systems?

### 4.1 Datasets

We report on data collection and curation for the two domains investigated.

#### 4.1.1 LastFM

**LFM-1b.** We base our analysis of recommendations for the music domain on the LFM-1b dataset (Schedl, 2016), which we enrich with two further datasets: acoustic features for a selection of tracks (the *CultMRS* dataset) curated by Zangerle *et al.* (Zangerle et al., 2020), and musical genres annotating a subset of tracks within LFM-1b, kindly provided by Schedl *et al.* (Schedl et al., 2020). The acoustic features contained in the dataset were re-

Table 1: Statistics of LFM-1b after merging with other datasets.

# listening events	379.754.730
# users	120.053
# artists	26.129
# tracks	282.011
# genres	2.137

trieved via the Spotify API<sup>3</sup> and serve as content-based features describing the nature of a track. Examples for these features are a track’s *tempo*, or *danceability*. After merging the three datasets, we are left with 379 million listening events (*cf.* Table 1).

**KG Construction.** From the merged LFM-1b dataset, we construct a KG consisting of artists, tracks, and genres. To model the relations among these entities, we use classes and properties provided by three different ontologies: FOAF<sup>4</sup>, Dublin Core<sup>5</sup> and Music Ontology<sup>6</sup> (Raimond et al., 2007); we define an auxiliary URI to identify entities from LastFM, <http://last.fm/lfm-resource>. For instance, a description of the track “Never Gonna Give You Up” by Rick Astley in Turtle syntax<sup>7</sup> would be:

```
lfmr:disco a mo:Genre ;
  dc:title "disco" .
lfmr:15160 a mo:MusicArtist ;
  foaf:name "Rick Astley" .
lfmr:t_4471632 a mo:Track ;
  dc:title "Never Gonna Give You Up" ;
  mo:genre lfmr:disco ;
  foaf:maker lfmr:15160 .
```

**Recommendations.** We sub-sample the listening events to 1000 users with at least 100 unique tracks in their profile. The mean number of tracks listened to is 1076 ( $\pm 1194$ ), with a median of 656 tracks. We use the Python library *Surprise* (Hug, 2022), which relies on explicit user-item ratings to determine the base recommendations. As our data contains implicit ratings as the number of times a track was listened to by a user, we follow the approach outlined in Kowald *et al.* (Kowald et al., 2021) and scale these play-counts into the range [1, 1000] using min-max-normalization; a user’s most-listened track will thus receive an explicit rating of 1000.

We evaluate six recommendation models provided by *Surprise*: *BaselineOnly*, which predicts a

<sup>3</sup><https://developer.spotify.com/documentation/web-api/reference/get-several-audio-features>

<sup>4</sup><http://xmlns.com/foaf/0.1/>

<sup>5</sup><http://purl.org/dc/elements/1.1/>

<sup>6</sup><http://purl.org/ontology/mo/>

<sup>7</sup><https://www.w3.org/TR/turtle/>

Table 2: Evaluation of prediction algorithms, sorted by increasing MAE.

Model	MAE
NMF	54.82
BaselineOnly	62.38
KNNWithZScore	65.74
KNNBaseline	67.01
KNNWithMeans	67.47
KNNBasic	71.10

baseline rating estimate from global averages and user/item deviations (*c.f.* (Koren, 2010)); *KNNBasic*, a user-based collaborative filtering approach using kNN; *KNNBaseline*, *KNNWithMeans*, and *KNNWithZScore*, extensions of the base kNN model taking into account baselines, mean ratings, and z-score normalized ratings, respectively; and *NMF*, a non-negative matrix factorization model. We use the default parameters provided by the library and employ cosine similarity as the distance measure for the kNN-based approaches.

Using 5-fold cross-validation, we evaluate each algorithm’s mean absolute error (MAE), and pick NMF as our final model; Table 2 presents MAE for all models.

We train NMF on the full data and retrieve rating predictions on the *anti-testset*, *i.e.*, on all items present in the training data that the user has not rated. The recommendation lists obtained this way are truncated to the top 100 items, sorted by descending predicted rating.

#### 4.1.2 Netflix

**Netflix Titles Dataset.** Our evaluation includes the domain of movies and TV shows. We considered the “Netflix titles” dataset, available on Kaggle<sup>8</sup>. This dataset provides a set of 8808 titles of movies and TV shows available on Netflix, along with their cast, directors, countries, release dates, ratings, and brief descriptions. All data is provided in a comma-separated value (CSV) file.

**KG Construction.** The catalog KG was created considering TBox statements representing properties and classes as provided in the CSV file. More specifically, the statements contain the type of each entry, which can be either a movie or a TV show. An actor acts on entries, and a director directs entries. Entries have an English title, a brief description, a country of origin, a rating, and a duration. All classes are of the

<sup>8</sup><https://www.kaggle.com/datasets/shivamb/netflix-shows>

rdf:Class type, and properties of the rdf:Property type.

The dataset provides no user data; therefore, we randomly generated 88 user profiles, ranging from a minimum of 5 to 55 entries, representing watched movies and TV shows. From this, user-profile KGs were generated using the same TBox as the catalog KG.

**Recommendations.** The recommendations were generated with the help of a state-of-the-art KG-based recommender system, KGAT (Wang et al., 2019). We used the same parameters for the configuration of graph convolutional layers, decay factors, and learning rates as the authors of the paper used for their evaluations.

The data set was cleaned up to obtain meaningful yet compact KGs. To this end, rdf:label-entities and nodes with a degree of 1 were removed. In addition, the rdf:Class and rdf:Property nodes were removed to prevent the knowledge graph from becoming too centralized. Certain entries in the KGs were labeled as recommendable items, *i.e.*, only movies and TV shows.

From user-profile KGs, the interactions on recommended items were registered and divided into training and test data using a 90/10 split, *i.e.*, 90% of the interactions of a user profile appear in the training set.

## 4.2 Experimental Procedure

For both datasets, we employed the following evaluation procedure:

1. Obtain base recommendations through SOTA model.
2. Rerank base recommendations according to graph metric (as outlined in Section 3); ranking proceeds in ascending and descending order of item relevance.
3. (LFM-1b only) For each metric and each sort order, measure Unexpectedness and Intra List Diversity using item features.
4. For each metric and each sort order, compare the reranked with the base lists using nDCG@10 via TREC\_EVAL<sup>9</sup>.

To emphasize that the proposed approach is independent of the underlying recommender system, we applied a different RS for each dataset: NMF for LFM-1b, and KGAT for the Netflix titles. The network metrics applied in this evaluation are the number

of nodes, number of edges, density, PageRank, average degree, {in,out}-degree, betweenness, and closeness centrality, summarized in Table 3. These metrics adhere to standard metrics in the field of social network analysis (Wasserman et al., 1994).

Table 3: Summary of network metrics considered in the experimental procedure.

Metric	Formula
# nodes	$N =  C $ , where $C$ is # entities
# edges	$E =  R $ , where $R$ is # relationships
Density	$\kappa = NE(E - 1)$
Degree centrality	$c_i^D = \sum_{j=1}^N \phi_{ij}$ where $\phi_{ij} = 1$ , if exists an edge, 0 if not;
In-degree centrality	$c_j^{ID} \sim c_i^D$ , incoming edges only
Out-degree centrality	$c_j^{OD} \sim c_i^D$ , outgoing edges only
Average degree	$\langle K \rangle = \frac{\sum_{i=1}^N c_i^D}{N}$
Betweenness centrality	$c_i^B = \sum_{j=1}^N \sum_{k=1, k \neq i, j}^N \frac{\eta_{jk}(i)}{\eta_{jk}}$ where $\eta_{jk}$ is # shortest paths from node $j$ to $k$ ; $\eta_{jk}(i)$ is # shortest paths from $j$ to $k$ containing $i$ .
Closeness centrality	$c_i^C = \frac{1}{\sum_{j=1, j \neq i}^N d(i, j)}$ where $d(i, j)$ is shortest distance for nodes $i$ to $j$ ; $d(i, i) = d(j, j) = 0$
PageRank	$p_i = \frac{q}{N} + (1 - q) \times \sum_j^M \frac{p_j^{(j)}}{c_j^{OD}}$ where $M$ is # nodes connected to $i$ . $c_j^{OD}$ is the out-degree of node $j$ , linked to $i$ ; $q$ is the damping factor

To evaluate Unexpectedness and Intra List Diversity, we represent each track as an 8D vector of acoustic features. The features we use are *danceability*, *energy*, *speechiness*, *acousticness*, *instrumentalness*, *liveness*, *valence* and *tempo*. In the original dataset, these features range in  $[0, 1]$ , except for tempo, which we scale into this range using min-max normalization following prior research (Zangerle et al., 2020; Kowald et al., 2021).

Intra List Diversity (ILD) measures the pairwise distance of all items in a recommendation list  $I$  w.r.t. a distance function  $d$  (c.f. (Castells et al., 2015)):

$$ILD(I) = \frac{1}{|I| \cdot (|I| - 1)} \sum_{i \in I} \sum_{j \in I} d(i, j) \quad (3)$$

We measure Unexpectedness on a user-profile level to determine how different a recommendation is from the user’s previous history. Essentially, this is the mean distance of each new item to each item the user has interacted with. Thus, for a user-profile  $H$ , a recommendation list  $I$  and a distance  $d$ , Unexpected-

<sup>9</sup>[https://trec.nist.gov/trec\\_eval/](https://trec.nist.gov/trec_eval/)

ness can be expressed as (cf. (Castells et al., 2015)):

$$Unexpectedness(R) = \frac{1}{|I| \cdot |H|} \sum_{i \in I} \sum_{h \in H} d(i, h) \quad (4)$$

We employed cosine distance between feature vectors as  $d$  for both ILD and Unexpectedness. In these measures and nDCG, we limit the recommendation lists to the top 10 items, in line with previous findings on users' searching behaviour (Jansen et al., 2000; Silverstein et al., 1999).

To further assess the rank-based dynamics underlying this reordering, we measure nDCG@10 for each re-ranking. The base recommendations serve as a ground truth of expectable recommendations for our purposes. Their ranking thus serves as the relevance judgment of items. High nDCG indicates that the same items are ranked highly in the base and re-ordered recommendations, whereas low nDCG indicates more perturbation in the second list. Our assumption is that low nDCG indicates that highly expectable items are ranked lower after reordering.

### 4.3 Experimental Results

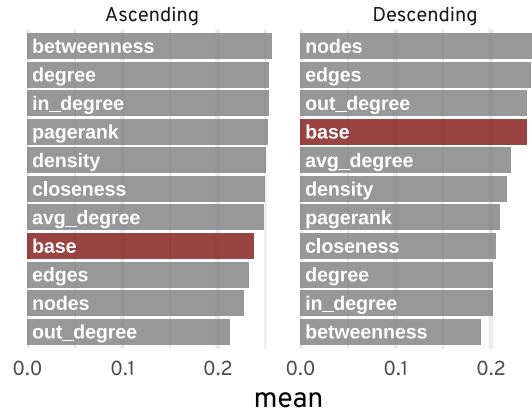
We present the results obtained from the experimental procedure for both datasets.

#### 4.3.1 LastFM

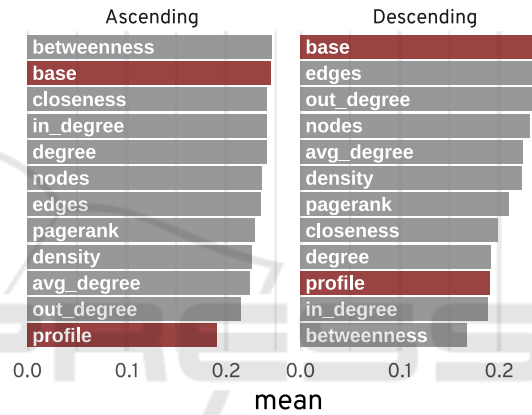
Before evaluating list perturbation, we first review the findings from measuring surprise on the reranked list of recommendations. Section 5 discusses obtained results and how they can be further interpreted from a network perspective.

We evaluate Unexpectedness and Intra List Diversity on all re-ranked recommendation lists and the two possible ranking orders. We include the measurements obtained on the original SOTA recommendations as a baseline; the users' mean profile diversity serves as a reference point for diversity. Figure 3 plots the mean measures against all metrics, split by ranking order. For Unexpectedness, sorting ascendingly by betweenness results in the largest deviations from the user's history, as shown in Figure 3a.

"Betweenness" in this case corresponds to the Herfindahl-Hirshman-Index (HHI) of the distribution. Sorting in ascending order thus places low index values at the top of the list, indicating a fairer distribution and, therefore, an overall less centralized network. The opposite holds for descending order, where favoring higher betweenness-indexes, and therefore more centralized networks, results in more expectable recommendations. We observe that increasing the number of nodes and edges in the users' subgraphs has the highest effect on Unexpectedness.



(a) Unexpectedness



(b) Diversity

Figure 3: Measuring surprise on feature-level for recommendations reranked by metric. For Unexpectedness (3a), the highlighted bars denote the comparison to the SOTA recommendations. For Diversity (3b), the highlighted bars denote the ILD of SOTA and original user profiles (*base* and *profile*, resp.).

Turning to Diversity, we first observed that users' listening behavior seems largely uniform, as indicated by the *profile* bars in Figure 3b. As the measure of ILD on the user profile is expressed as the mean pairwise distance between items in the list, a low distance on average indicates the presence of tracks with similar acoustic features.

We found that preferring a low betweenness index and thus decentralized networks results in diverse tracks being recommended for ascending sort order, whereas for the descending case, increasing the number of nodes, edges, and out-degree produces the most diverse lists out of our approaches but does not outperform the baseline recommendations. An interesting observation is that the base recommendations already contain very diverse items. This is in line with



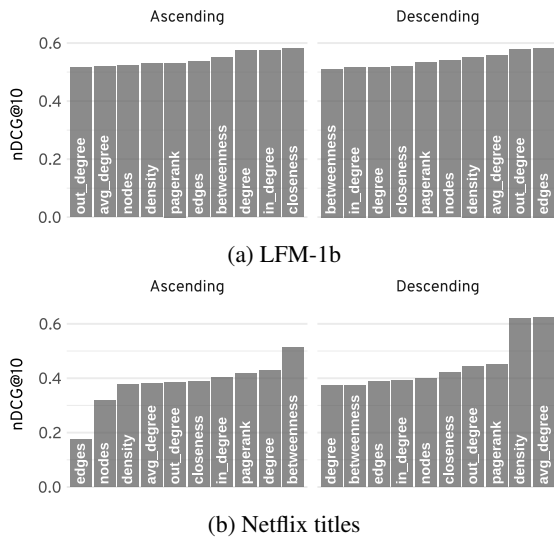


Figure 4: nDCG@10 for both datasets. Panels show nDCG scores obtained on recommendation lists ranked by metric in ascending and descending order.

previous findings of the underlying algorithm, NMF, being able to recommend items from the long-tail of user-item-interactions (Kowald et al., 2020).

To assess the extent to which re-ranked lists correspond with the original, expectable ranking, Figure 4a plots nDCG@10 for all metrics. We found that out-degree and betweenness, particularly, result in a high perturbation of ranks. We highlight that items considered relevant by an expectable RS are not ranked highly after optimizing for one of the network metrics.

### 4.3.2 Netflix

Unlike LastFM, the Netflix titles dataset provides no additional content-based features that would allow measuring Unexpectedness and Diversity. The analysis of this dataset is therefore based on the nDCG score, considering the initial 10 elements of the recommendation list. Figure 4b summarizes these results by metrics, both in ascending and descending orders of relevance.

Similarly to LastFM, we observe that betweenness centrality is the metric that introduces the most surprise into the list of recommendations obtained from the RS. To further illustrate this finding, Table 4 presents the first three recommendations offered by the RS, compared with those reranked by betweenness centrality in particular user-profile KGs.

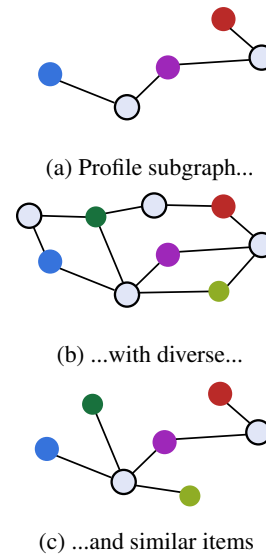


Figure 5: Illustration of the effect of recommendations on betweenness centrality in profile subgraphs. Diverse items being recommended open alternative paths in the resulting profile subgraph lowering the betweenness of all nodes (5b), whereas similar items tend to increase the betweenness of a few nodes (5c).

## 5 DISCUSSION

We observe that recommendations sorted by betweenness in ascending order of the associated HHI exhibit high Unexpectedness and Diversity. Ranking in this way favors nodes that result in a lower HHI, thus, revealing a more decentralized user subgraph. In such a KG, many paths among concepts exist, and there is low monopolization. The opposite holds for a highly centralized KG, in which a small number of concepts appear along many paths and carry high importance.

Figure 5 illustrates this effect, presenting an example user subgraph 5a and two extensions arising from incorporating more diverse (*cf.* Figure 5b) or more similar (*cf.* Figure 5c) recommendations. Interactions and recommendations are shown as solid colored circles; related concepts are light colors with an outline. Diverse items will likely be loosely connected to existing concepts the user is familiar with and bring along further related nodes, thus expanding the user’s exposure. Contrast this with the second example, where similar items are introduced that only exhibit relations to concepts familiar to the user. These examples illustrate the effect on the number of edges, nodes, and degree-related measures. In the diverse case, adding two recommendations results in four nodes and seven edges added to the graph versus two nodes and two edges for the case of similar items.

Table 4: Comparing the top three recommended items obtained from state-of-the-art recommender, against those re-ranked using betweenness centrality applied on a user-profile KG.

Dataset	SOTA recommender	Re-ranked (betweenness)
Netflix	Bakugan: Armored Alliance	Creeped Out
	The C Word	Black Mirror
	Weird Wonders of the World	Arthur Christmas
LFM-1b	Iron Maiden, The Talisman	Shakira, Spotlight
	Iron Maiden, When the Wild Wind Blows	Here We Go Magic, Make Up Your Mind
	Shakira, Spotlight	Here We Go Magic, Alone But Moving

Considering the results from evaluating nDCG, we observed that ranking by betweenness, node-/edge counts, or degree-based metrics yields lists with low-rank correlation compared to expectable recommendations.

Our study demonstrated that network-level metrics correlate with key surprise elements such as diversity and unexpectedness (**RQ1**). We found betweenness resulting in the most diverse and unexpected recommendations that rank expectable items lower than a state-of-the-art baseline. We showed that adding a KG-informed reranking model on top of an existing recommender system can thus introduce a level of surprise into user-item recommendations (**RQ2**).

Results highlighted that calculating betweenness may not be computationally feasible in constrained environments, especially on large profile subgraphs. Besides truncating user profiles to the most recent interactions as a solution in this case, our findings suggest that node-/edge counts or degree-based features are viable alternatives to betweenness.

We identify the Netflix dataset’s lack of rich content-based features, prohibiting a similar investigation of surprise-related measures as performed for the enriched LFM-1b dataset. Furthermore, a user study should evaluate the degree of surprise, as listening and viewing behaviors are governed by highly subjective user dynamics. We plan to address this in future studies by considering different baselines to compare our method’s results. Furthermore, although this study focused on exploring and comparing metrics for reranking a state-of-the-art baseline, the developed system is capable of generating recommendations without requiring a base model; this is also a subject for future studies.

Many user-profile KGs are sparse and not dense, especially when considering real-world user profile information on distinct scenarios and domains. The initial step of our approach, *i.e.*, the generation of recommendations, is affected by data sparsity similar to the underlying state-of-the-art baseline system. The reranking phase, especially the centrality measures,

requires a connected graph. If the employed KG is sparsely connected, limiting the KG to the largest connected component, or using metrics less reliant on connections, such as degree and node-/edge counts, is an approach to overcome this aspect. This also reinforces that the metric choice can influence the final results.

The catalog KGs employed in our study only contain intra-domain concepts (artists, music genres, directors, actors, etc.). However, KGs are well suited for linking cross-domain concepts, *e.g.*, tracks that appear in a movie’s score, or actors who are musicians. Not only does this result in a richer representation of domains, it also enables cross-domain recommendations. We defer an analysis of surprising recommendations in such settings to future work.

## 6 CONCLUSION

We still encounter open research challenges in how systems may deal with and benefit from surprise recommendations. This investigation designed a solution incorporating network-level metrics to introduce personalized yet unexpected recommendations to users. We evaluated the LastFM music and Netflix movies datasets to determine the extent to which Intra List Diversity, Unexpectedness, and comparison to nDCG, respectively, affect the degree of surprise in recommendations. We found that network-level metrics indeed influence the degree of surprise in recommendations. Our results demonstrated that betweenness centrality showed a stronger influence when reranking recommendations for surprise. Future work involves additional analysis of surprising recommendations and how content-based features from items can be combined with our designed approach.

## SUPPLEMENTAL MATERIAL

Source code and data for the experiments and evaluations conducted in this work are available at <https://github.com/baumann/kg-recommender>. The LFM-1b dataset is available at <http://www.cp.jku.at/datasets/LFM-1b/>, the CultMRS dataset at <https://zenodo.org/records/3477842>, and the Netflix titles dataset at <https://www.kaggle.com/datasets/shivamb/netflix-shows>.

## ACKNOWLEDGEMENTS

This article is the outcome of research conducted within the Africa Multiple Cluster of Excellence at the University of Bayreuth, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC 2052/1 – 390713894. This work is also supported by the ‘PIND/FAEPEX - “Programa de Incentivo a Novos Docentes da Unicamp” (#2560/23) and the São Paulo Research Foundation (FAPESP) (Grant #2022/15816-5)<sup>10</sup>

## REFERENCES

- Abdollahpouri, H., Burke, R., and Mobasher, B. (2019). Managing popularity bias in recommender systems with personalized re-ranking. *arXiv preprint arXiv:1901.07555*.
- Baumann, O. and Schoenfeld, M. (2022). Supporting Serendipitous Recommendations with Knowledge Graphs. In Tamine, L., Amigó, E., and Mothe, J., editors, *2<sup>nd</sup> Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2022)*, number 3178 in CEUR Workshop Proceedings, Aachen.
- Castells, P., Hurley, N., and Vargas, S. (2021). Novelty and diversity in recommender systems. In *Recommender systems handbook*, pages 603–646. Springer.
- Castells, P., Hurley, N. J., and Vargas, S. (2015). Novelty and Diversity in Recommender Systems. In Ricci, F., Rokach, L., and Shapira, B., editors, *Recommender Systems Handbook*, pages 881–918. Springer US, Boston, MA.
- De Gemmis, M., Lops, P., Semeraro, G., and Musto, C. (2015). An investigation on the serendipity problem in recommender systems. *Information Processing & Management*, 51(5):695–717.
- Felfernig, A. and Burke, R. (2008). Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th international conference on Electronic commerce*, pages 1–10.
- Ge, M., Delgado-Battenfeld, C., and Jannach, D. (2010). Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems, RecSys ’10*, pages 257–260, New York, NY, USA. Association for Computing Machinery.
- Hirschman, A. O. (1964). The paternity of an index. *The American Economic Review*, 54(5):761–762.
- Hug, N. (2022). NicolasHug/Surprise.
- Hui, B., Zhang, L., Zhou, X., Wen, X., and Nian, Y. (2022). Personalized recommendation system based on knowledge embedding and historical behavior. *Applied Intelligence*, pages 1–13.
- Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the web. *Information Processing & Management*, 36(2):207–227.
- Joseph, K. and Jiang, H. (2019). Content based news recommendation via shortest entity distance over knowledge graphs. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 690–699.
- Kaminskas, M. and Bridge, D. (2016). Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7(1):1–42.
- Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data*, 4(1):1–24.
- Kotkov, D., Wang, S., and Veijalainen, J. (2016). A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192.
- Kowald, D., Muellner, P., Zangerle, E., Bauer, C., Schedl, M., and Lex, E. (2021). Support the underground: Characteristics of beyond-mainstream music listeners. *EPJ Data Science*, 10(1).
- Kowald, D., Schedl, M., and Lex, E. (2020). The Unfairness of Popularity Bias in Music Recommendation: A Reproducibility Study. *Advances in Information Retrieval*, 12036:35–42.
- Liu, W., Xi, Y., Qin, J., Sun, F., Chen, B., Zhang, W., Zhang, R., and Tang, R. (2022). Neural re-ranking in multi-stage recommender systems: A review. *arXiv preprint arXiv:2202.06602*.
- Pei, C., Zhang, Y., Zhang, Y., Sun, F., Lin, X., Sun, H., Wu, J., Jiang, P., Ge, J., Ou, W., et al. (2019). Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM conference on recommender systems*, pages 3–11.
- Raimond, Y., Abdallah, S. A., Sandler, M. B., and Giason, F. (2007). The music ontology. In Dixon, S., Bainbridge, D., and Typke, R., editors, *Proceedings of the 8<sup>th</sup> International Conference on Music Information Retrieval, ISMIR 2007, Vienna, Austria, September 23-27, 2007*, pages 417–422. Austrian Computer Society.
- Rossanez, A., da Silva Torres, R., and dos Reis, J. C. (2023). Characterizing complex network properties of knowledge graphs. In *Proceedings of the 15th International Joint Conference on Knowledge Discovery*,

<sup>10</sup>The opinions expressed in this work do not necessarily reflect those of the funding agencies.

- Knowledge Engineering and Knowledge Management - KEOD*, pages 119–128. INSTICC, SciTePress.
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.
- Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*, pages 291–324. Springer.
- Schedl, M. (2016). The LFM-1b Dataset for Music Retrieval and Recommendation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM.
- Schedl, M., Mayr, M., and Knees, P. (2020). Music Tower Blocks: Multi-Faceted Exploration Interface for Web-Scale Music Access. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 388–392. Association for Computing Machinery, New York, NY, USA.
- Schoenfeld, M. and Pfeffer, J. (2021). Shortest path-based centrality metrics in attributed graphs with node-individual context constraints. *Social Networks*.
- Silverstein, C., Marais, H., Henzinger, M., and Moricz, M. (1999). Analysis of a very large web search engine query log. *ACM SIGIR Forum*, 33(1):6–12.
- Wang, X., He, X., Cao, Y., Liu, M., and Chua, T.-S. (2019). Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958.
- Wasserman, S., Faust, K., and Urbana-Champaign, S. U. o. I. W. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press.
- Zangerle, E., Pichl, M., and Schedl, M. (2020). User Models for Culture-Aware Music Recommendation: Fusing Acoustic and Cultural Cues. *Transactions of the International Society for Music Information Retrieval*, 3(1):1–16.
- Zhang, F., Yuan, N. J., Lian, D., Xie, X., and Ma, W.-Y. (2016). Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.