Explanation-Driven Carousels for Two-Dimensional Recommendation Interfaces

André Levi Zanon[1,*], Derek Bridge[1]

[1]Insight Centre for Data Analytics, School of Computer Science and Information Technology,University College Cork, Ireland

Abstract

Many deployed Recommender Systems present recommendations in a two-dimensional grid of carousels, where each carousel has an explanation in the form of a descriptive title. For instance, movie recommendations may be grouped into carousels entitled "Films based on Books" or "Award-winning science fiction", which are explanations shared by all items in a carousel. In this paper, we propose a method for organizing a set of recommendations into a two-dimensional interface that is driven by the generation of shared explanations. Our approach starts from a ranked list of recommendations produced by a RS. For each recommended item, we use a Knowledge Graph to generate a post-hoc explanation in the form of a set of metadata entities, which we encode as a vector. We then cluster these vectors to group recommendations with similar explanations where each resulting cluster is displayed as a carousel in the 2D interface. Our results shows that the method can organize carousels with explanations.

Keywords

Recommender Systems, Recommendation explanation, Recommendation evaluation, Carousel

1. Introduction

Currently, many Recommender Systems (RSs) display recommendations in a user interface that comprises multiple rows, called carousels, that form a two-dimensional grid structure [1]. Each carousel contains related recommendations — some visible, others requiring swiping — and comes with a descriptive title (such as "Popular Shows") explaining what the items have in common [2]. These descriptive titles act as shared explanations for the items within the carousels, providing a common rationale that applies to all recommendations in the row. Descriptive titles must be common to all items in a carousel, to avoid misinformation (e.g.: a comedy movie on a drama carousel). In systems such as Netflix, they highlight that titles should be diverse to account for uncertainty in the recommender's understanding of user preferences and familiar to help users quickly locate relevant items [3].

Despite the importance of carousel-based interfaces in deployed RSs, the algorithmic organization of recommendations into carousels, and the generation of meaningful descriptive titles, remain underexplored [4]. Therefore, in this paper, our main contribution is *an explanation-driven method for organizing a top-k list of recommendations into carousels, where explanations are in the form of descriptive titles associated to all items in a carousel* . To achieve this objective, for each item recommended by a RS, we use a Knowledge Graph (KG) to generate an explanation in the form of a set of metadata entity nodes. We then cluster recommended items based on explanation similarity. Each cluster becomes a carousel, and metadata entities shared across explanations are used to generate the carousel's descriptive title.

The order of the carousels (from top-to-bottom in the grid) and the placement of items within the carousels (from left-to-right across the grid) is also important. Eye-tracking studies show that in two-dimensional (2D) interfaces with multiple carousels, users do not browse recommendations sequentially from left-to-right and top-to-bottom, as shown in Figure 1a. Instead, attention often follows a "golden triangle" pattern, as illustrated in Figure 1b [5]. All of this has consequences for the experimental evaluation of recommendations in carousel-based interfaces. In this paper, we report an experiment

Late-breaking work, Demos and Doctoral Consortium, colocated with the 4th World Conference on eXplainable Artificial Intelligence: July 01–03, 2026, Fortaleza, Brazil

*Corresponding author.

andre.zanon@insight-centre.org (A. L. Zanon); d.bridge@cs.ucc.ie (D. Bridge)

0000-0003-0526-2678 (A. L. Zanon); 000-0002-8720-3876 (D. Bridge)

(a) Sequential

(b) Golden triangle

Figure 1: User assessment of top-20 recommendations on a carousel interface. Figure inspired by [5].

which evaluates (i) the quality of the carousels; (ii) the precision of the recommendations but taking into account their placement in the grid; and (iii) the quality of the descriptive titles (shared explanations). The results show show that our method produces near-optimal layouts as well as meaningful and interpretable carousel titles. Our code is available [1] and includes the datasets as well as all generated outputs.

2. Related Work

A 2D interface for a RS consists of multiple lists of items. Each list, known as a carousel, forms a row on the interface and is associated with a descriptive title that acts as a shared explanation for all the items within the carousel. The limited amount of research on these interfaces falls into two groups. There is some work on creating carousels from recommended items; and there is some work on RS evaluation where the recommendations have been placed into a 2D interface.

Some studies focus on the creation of multi-lists of items assuming that the explanation for each row is predefined [6, 7, 8]. In [6], Ding et al. generated a "Trending Now" carousel by integrating a RS architecture with time series forecasting to recommend items predicted to become popular. Instead, Ma et al. created an iterative hierarchical clustering algorithm to produce carousels for events[7]. For a carousel of "New Discoveries", Briand et al. applied contextual bandits to recommended cold-start items [8].

To generate a 2D interface with multiple carousels, Dacrema et al. used quantum computing to select the optimal set of rows, where each of them was generated by a different recommendation algorithm [9]. However, the use of multiple collaborative filtering algorithms can bias rows to display only popular items [3]. To circumvent this problem, works such as [10, 11] display rows whose descriptive titles are based on different criteria can diversify the recommendations. Edziel at al. for instance, proposed a framework that integrates user embeddings from a matrix factorization RS with graph embeddings to generate item representations to organize a top-k list of recommendations into carousels.

Consequently, in the literature few are the methods that generate a set a 2D interface. Instead, most works consider generating a single carousel [8] or analyze user behavior [1, 12]. The most similar work to ours is [11]. However, they only consider genres to form descriptive titles, while we extract a KG which expand the expressiveness.

3. Explanation-Driven 2D Reordering

Our method takes a recommendation list generated by a RS and uses a KG to create post-hoc explanations for each recommendation in the form of sets of metadata entities. These explanations are transformed into vector representations and fed into a clustering algorithm. In the 2D interface, each carousel corresponds to a cluster, with shared explanation entities providing the descriptive titles.

A KG is a set of triples $KG = \{(h, r, t) : h, t \in E, r \in R\}$, where E and R denote, respectively, the sets of entities (nodes) and edge types. Figure 2 depicts a fragment of a movie KG. An *explanation* consists of metadata entities on paths between the recommended item and those the user interacted

[1] https://github.com/andlzanon/list-aware-explanations

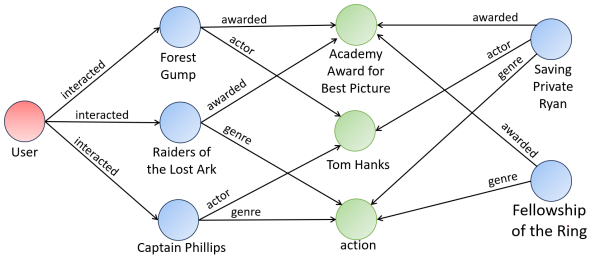

Figure 2: KG connecting user (in red), items (in blue) and metadata (in green).

with. For instance, recommendation "Fellowship of the Ring" is explained by the user's interactions with Academy Award-winning and action movies by the edge types of "awarded" and "genre", respectively.

We prepare explanations for clustering by encoding them as vectors. Each position in the vector corresponds to one of the metadata entities, and the simplest encoding uses 1 to indicate that a metadata entity is part of an explanation and 0 otherwise. For example, taking the metadata entities "Academy Award for Best Picture", "Tom Hanks" and "action" in that order, the explanation for "Saving Private Ryan" is encoded as $[1, 1, 1]$; for "Fellowship of the Ring", the encoding is $[1, 0, 1]$.

In the following, let I_u be the set of items that the user has interacted with (her user profile). Let R_u be the set of items recommended to user u. For a given recommended item $i \in R_u$, let $E_{u,i}$ be the set of metadata entities that comprise the explanation of why i is being recommended to u; in other words, metadata entities that connect i to items in I_u. We experiment with four different encodings:

1. *Binary* encoding, where 1 in the vector represents that a metadata entity e is part of an explanation $E_{u,i}$, and 0 otherwise.
2. *Count* encoding, defined as $count(e) = (\frac{n_{e,I_u}}{|I_u|})$ where each position gives the number of edges n_{e,I_u} between metadata node e and items the user interacted with I_u relative to the size of I_u.
3. *Term Frequency–Inverse Document Frequency (TF-IDF)* encoding, which is inspired by TF-IDF scores for terms in documents. We use the $count$ function, and discount these by an equivalent to inverse document frequency. Consequently this encoding is defined as $TFIDF(e) = count(e) \times IDF(e)$. In our scenario, $IDF(e) = \log(\frac{N}{df_e})$, where N will be the number of items and df_e will be the number of items linked to metadata entity e.
4. *ExpLOD* encoding (Equation 1), where each position records the metadata's ExpLOD score [13]. ExpLOD is a KG post-hoc explanation algorithm that scores metadata nodes in explanations.

$$explod(e, I_u) = (\alpha \frac{n_{e,I_u}}{|I_u|}) + (\beta \frac{n_{e,R_u}}{|R_u|}) \times IDF(e) \quad (1)$$

n_{e,I_u} and n_{e,R_u} denote the number of links between metadata entity e and the sets of interacted items (I_u), and the number of links between e and recommended items (R_u), respectively. These are weighted by α and β, that were set to 0.5 following [13].

The vector encodings for the explanations of the recommended items are input to a clustering algorithm. We experimented with the following algorithms: (1) *Agglomerative clustering*; (2) *HDBSCAN*; (3) *Spectral clustering*; (4) *Bissecting KMeans (BKMeans)*. In all cases, the distance metric used was the (complement of) cosine similarity. We configure each clustering algorithm to produce clusters equal to the desired number of rows in the 2D interface (see Section 4.2).

Each cluster of items is represented as a carousel, which is then placed into a distinct row of the 2D interface. These carousels are vertically ordered (top-to-bottom) based on the relevance of their contents, as determined by the rankings produced by the RS. Specifically, the cluster containing the top-ranked item overall is placed in the first row, followed by the cluster with the next highest-ranked item, and so on. Within each carousel, items are arranged horizontally from left to right according to their individual ranks assigned by the RS within that cluster. For each carousel, and to align to our baseline [13], described in Section 4.1, up to two of the metadata entities with the highest scores

in the encoding vectors are selected as the descriptive title, therefore constraining explanations to 2-edge paths. Titles must be shared by all items in a carousel. As a result, the common metadata nodes corresponding to the highest values in these vectors are chosen as the descriptive title. In cases where multiple metadata entities share the highest value, ties are broken randomly.

4. Experiments

4.1. Datasets and Models

We evaluate our approach using the MovieLens-Latest-Small (MovieLens) [14] and Hetrec2011-LastFM-2k (LastFM) [15] datasets. KGs were extracted from Wikidata[2] for each. We used this version of the MovieLens dataset because it includes a mapping of items to their *IMDB* identifiers, which was used to query Wikidata. For LastFM, data was retrieved by matching artist names to Wikidata entities. We obtained metadata for 98% of MovieLens items and 66% of LastFM items. On MovieLens the KG contains 9,535 items, 23 different edge types, and 69,487 metadata notes. On the LastFM, we found 11,646 items on Wikidata, 33 different edge types and 23,001 metadata nodes.

We discarded from the two datasets items for which we were unable to collect Wikidata. The resulting MovieLens dataset comprised 100,521 ratings for 610 users and 9,517 items. This was 99% of the original ratings. The resulting LastFM contained 83,038 interactions (89% of the original) for 1,884 users and 11,646 items. We split each dataset, using 80% of each user's ratings or interactions for training and 20% for testing. We generate carousels and evaluate explanations with offline metrics for all test users.

The RS we use is Bayesian Personalized Ranking with Matrix Factorization (BPR-MF) from the Cornac library [16] with the following hyperparameter: embedding size of 200, a learning rate of 0.01, regularization factor of 0.001, and 200 training epochs. We evaluated seven versions of our proposed approach: four of them use Agglomerative Clustering but vary the encoding technique (Binary, Count, TF-IDF, and ExpLOD) and three of them vary the clustering algorithm (BKMeans, Spectral, and HDBSCAN) using the ExpLOD encoding technique.

Baseline: We also have a Baseline model. It uses the state of the art ExpLOD [13] for each carousel. This placement of items is based on their predicted relevance score from the RS. It fills the first carousel with the items that have the highest relevance scores; the next most highly-ranked items fill the second carousel; and so on. ExpLOD can give explanations in the form of metadata entities to *groups* of items. Here, when scoring the metadata entities, it uses Equation 1; however, R_u becomes the set of recommended items on a single carousel rather than the set of all items recommended to user u. For comparability with our method, the descriptive title will be up to two metadata entities from the carousel explanation shared by the items in the carousel.

4.2. 2D User Interface Configuration

We generate a 2D interface of the top-k recommendations for every user in both datasets. Consequently, the following information is required: the value of k, the number of carousels on the 2D interface, how many carousels are visible without swiping, and how many items in each carousel are visible without swiping. We decided to use two configurations: one suitable for small-screen mobile devices and another for personal computers.

For MovieLens experiments, we were guided by the number of carousels and items visible in the Netflix mobile app and personal computer web user interfaces. On mobile devices, we chose $k = 350$, placed in 35 carousels, with the initial 3 carousels and 3 items of each carousel visible to the user. On personal computers, we chose $k = 500$ recommendations, placed in 40 carousels, with 3 carousels and 6 items of each carousel visible. For LastFM, we were guided by Spotify app interfaces to set the same numbers. On mobile devices, we chose $k = 300$, placed in 30 carousels, with 2 carousels and 2 items of each carousel visible. On personal computer devices, we chose $k = 250$ recommendations, placed in 25 carousels, with 2 carousels and 5 items of each carousel visible.

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

4.3. Evaluation Metrics

We evaluate using five metrics that measure three different aspects: two that measure the quality of the clustering; one that measures the placement of items on the 2D interface; and three to evaluate descriptive titles.

Quality of clustering: We assess cluster quality using the *Silhouette* and *Entropy* metrics. The Silhouette score measures the quality of a clustering by evaluating how well each element is assigned to its cluster compared to other clusters. In addition, we use Entropy to measure how well-distributed recommended items are across the clusters. As a result, the ideal value is for the entropy to be high, meaning that the clusters have similar size, and, consequently, carousels will have a similar length.

Placement of items: We use Normalized Discounted Cumulative Gain (NDCG)-2D [5], which extends NDCG to consider how users assess recommendations in 2D interfaces (Figure 1b). The main objective of the metric is to provide high values for a recommendation interface that requires the minimum amount of effort from the user in scrolling and swiping to find relevant items. The discount function is: $2d\_discount(i,j) = \log_2(\alpha i + \beta j + \gamma swipes(i, init_v, step_v)) + \lambda swipes(j, init_h, step_h)$.

The function $swipes$ is calculated as $swipes(p, init, step) = max(0, \lceil \frac{p-init}{step} \rceil)$ and it calculates when a swipe from the user is required for an item to appear on the screen. This happens when the position (horizontal or vertical) p is higher than the number of items $init$ that are appearing horizontally or vertically. Parameters α, β, γ, and λ were set to 1, as in [5]. Variables $init_h$ and $init_v$ have their respective values assigned depending on the domain (movies or musical artists) and device (mobile or web) as described in Section 4.2 where the number of carousels is $init_v$ and number of visible items in each carousel is $init_h$. For both datasets and for both the mobile and personal computer configurations, we set $step_h$ and $step_v$ to 1.

Descriptive titles: Carousel titles should be familiar, to aid intelligibility, yet diverse across carousels to offer choice [3]. To assess familiarity, we apply the *Shared Entity Popularity (SEP)* metric [17] to each metadata entity in the title, and take the mean. SEP is calculated as $SEP(e^i, v^i) = (1-\beta) \times SEP(e^{i-1}, v^{i-1}) + \beta \times v^i$. Here, e is the metadata entity, v is its linkage (popularity) count, and i is its index after sorting by v. Since it is a recursive function, when $i = 0$, it returns the number of links of the least 'popular' entity. SEP scales v^i by a parameter β, which we set to 0.3 as per [17]. Finally, the values are normalized to $[0, 1]$ before taking their mean. Higher values indicate that the descriptive title contains 'popular' metadata entities.

We based our measure of diversity of the descriptive titles on the *Explanation Type Diversity (ETD)* metric in [17]. In our version, ETD is calculated as the number of unique metadata entities in all of a user's carousel titles divided by the total number of metadata entities in those titles. Values range from 0 to 1, with higher values indicating that there is no repetition of titles in across all carousels. It can happen that the explanations of the items on a carousel have nothing in common and so a descriptive title cannot be formed. For this we compute *ClusterMiss*, defined as the number of a user's carousels where there is no metadata entity shared across all items in the carousel divided by the total quantity of carousels generated for the user. Values range from the ideal 0, where all have titles, to 1, when no carousels have titles.

What are good carousels? An ideal algorithm should not excel in only one metric, but instead produce a good balance between all the metrics. In [3], it is described that a platform like Netflix seeks descriptive titles that balance the popularity and diversity of descriptive titles. Therefore, it is required that SEP and ETD are in equilibrium. In addition, we argue that each carousel should be made by clearly separated and well-distributed clusters, place items in accordance with what the RS considers relevant. Consequently, it is also important to improve the quality of clustering metrics and the placement of items metrics.

5. Results

Results are shown in Table 1 and Table 2, where each column is an algorithm and the best values in each column are highlighted in *bold*. Within each cell, the upper value is for the mobile device interface, while below is for the personal computer interface, as described in Section 4.2 and Section 4.3. Overall, results were also mostly consistent across the two datasets.

	Baseline	Binary	Count	TF-IDF	ExpLOD	BKMeans	Spectral	HDBSCAN
NDCG-2D	*0.4305*	0.4074	0.4045	0.4080	0.4088	0.4061	0.4106	0.3755
	0.4610	0.4293	0.4277	0.4324	0.4327	0.4277	0.4340	0.3947
SEP	0.5176	0.5611	*0.7435*	0.5894	0.3406	0.4560	0.5685	0.5454
	0.4100	0.5544	*0.6363*	0.3529	0.4153	0.4708	0.4644	0.5585
ETD	0.0799	*0.5758*	0.2525	0.5721	0.5190	0.3070	0.5484	0.4212
	0.0933	0.5701	0.2618	*0.5747*	0.5061	0.2839	0.5568	0.3399
Cluster Miss	0.4823	0.0874	*0.0*	0.0240	0.0177	0.0753	0.0844	0.1135
	0.5899	0.1153	*0.0*	0.0295	0.0211	0.0811	0.0817	0.0772
Silhouette	-0.0777	0.0604	*0.3040*	0.1973	0.1742	0.0919	0.0814	-0.0506
	-0.0791	0.0533	*0.3193*	0.1970	0.1708	0.0955	0.0663	-0.0790
Entropy	*3.5553*	3.0198	2.5400	2.9129	2.9290	3.4019	3.1571	1.9733
	3.6826	3.1145	2.4501	2.9947	3.0278	3.5236	3.2842	2.1719

Table 1
MovieLens results. In each cell, the upper value is for a mobile interface and beneath is for a computer interface.

	Baseline	Binary	Count	TF-IDF	ExpLOD	BKMeans	Spectral	HDBSCAN
NDCG-2D	*0.3381*	0.3264	0.3123	0.3204	0.3208	0.3273	0.3233	0.3116
	0.3274	0.3149	0.3030	0.3102	0.3116	0.3158	0.3135	0.3038
SEP	0.0017	0.4571	0.5704	0.4792	0.5745	0.5852	0.4341	*0.6227*
	0.0016	0.4501	0.5761	0.4753	0.5628	0.5785	0.4532	*0.6260*
ETD	1.0	0.7676	0.7042	*0.8299*	0.7134	0.3830	0.6818	0.3760
	1.0	0.8028	0.7353	*0.8514*	0.7479	0.4031	0.7196	0.3963
Cluster Miss	0.9981	0.1195	*0.0027*	0.0392	0.0224	0.1525	0.1190	0.0883
	0.9983	0.1287	*0.0056*	0.0537	0.0263	0.1599	0.1286	0.1050
Silhouette	-0.1258	0.1088	*0.4857*	0.3222	0.2592	0.1373	0.1217	0.0255
	-0.1171	0.1113	*0.4879*	0.3185	0.2596	0.1443	0.1445	0.0416
Entropy	*3.4011*	2.6640	2.1062	2.5026	2.4996	3.1528	2.2974	2.1413
	3.2188	2.4770	2.0140	2.3637	2.3750	2.9912	2.2494	2.0788

Table 2
LastFM results. In each cell, the upper value is for a mobile interface and beneath is for a computer interface.

The Baseline, have a good entropy and placement of items (NDCG-2D), since it assigns items to carousels, filling one before filling the next. However this it has also the highest values for Cluster Miss: only half of its MovieLens carousels have titles; and for LastFM, titles were nearly absent. In the case of LastFM, this makes SEP and ETD values meaningless since there were almost no explanations shared across all items in a carousel. In this latter case, it affected the diversity metric ETD, which reached its highest value (1.0 in Table 2). Due to this phenomenon, we excluded this result when highlighting best values in bold. *Hence, carousels of items not clustered by their explanation metadata mostly did not share metadata, preventing them from having a descriptive title.*.

Let us now look at the results for the four encoding methods used with agglomerative clustering. The Binary encoding balances SEP and ETD, especially in the MovieLens dataset, but at the cost of a high Cluster Miss. On the MovieLens dataset, Count looks to perform well: zero Cluster Miss, the highest SEP and the highest Silhouette scores. But on this dataset, the diversity of its titles (ETD) is the lowest of the approaches: there is a relatively high overlap between the descriptive titles that it shows to a user.

For this dataset, TF-IDF and ExpLOD encodings seem to give the best balance: low Cluster Miss and a reasonable trade-off between SEP and ETD, while having good cluster quality (Silhouette and Entropy). Across the two datasets, we conclude that *the 'richer' encodings (TF-IDF and ExpLOD) give better-balanced results than the simpler encodings (Binary and Count).*

Lastly, we look at the three other clustering algorithms: BKMeans, Spectral and HDBSCAN. For these we used ExpLOD encoding. They fail to find a descriptive title in roughly 7.5–16% of cases, making them worse than Count, TF-IDF and ExpLOD encodings with agglomerative clustering. Their

Silhouette scores were also worse than those for agglomerative clustering using Count, TF-IDF and ExpLOD encodings but their SEP and ETD scores are competitive.

Across all seven approaches and both datasets, *the algorithm that achieves the best balance across all metrics is the ExpLOD encoding using agglomerative clustering*. On the MovieLens dataset, for instance, it has the second lowest Cluster Miss, meaning that it only rarely fails to find descriptive titles and it has a competitive NDCG-2D. Its Entropy values are also close to the algorithms with the best values. Finally, it can also balance the popularity and diversity of metadata entities, measured by the SEP and ETD metrics. A screenshot of carousels generated by this algorithm for the MovieLens user with id 1 are shown in our source code repository[3].

6. Conclusions

In this work, we introduced a method for organizing the output of a RS into a 2D interface of carousels, each accompanied by an automatically generated descriptive title. Our approach clusters recommended items based on the similarity of their explanations, which are derived from metadata entities extracted from a KG. Each resulting cluster corresponds to a carousel, while the metadata entities shared across the explanations of the items in a cluster form the basis of its descriptive title. We evaluated the quality of the clusters, the placement of the items and carousels on the screen, and the quality of the descriptive titles that we generated. Our results are promising but there are limitations to be addressed such as the impact of the KG and RSs on the results, scalability, and using other encoding techniques such as embeddings. In addition conducting user studies to compare different carousel construction and explanation strategies, constitute directions for future work.

Acknowledgments

This publication has emanated from research conducted with the financial support of Research Ireland under Grant number 12-RC-2289-P2, which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] A. Starke, E. Asotic, C. Trattner, "serving each user": Supporting different eating goals through a multi-list recommender interface, in: Proceedings of the 15th ACM Conference on Recommender Systems, RecSys '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 124–132. doi:10.1145/3460231.3474232.

[2] B. Loepp, J. Ziegler, How users ride the carousel: Exploring the design of multi-list recommender interfaces from a user perspective, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1090–1095. doi:10.1145/3604915.3610638.

[3] C. Alvino, J. Basilico, Learning a personalized homepage, Netflix Tech Blog, 2015. April 9, 2015. Retrieved from https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a.

[3] https://github.com/andlzanon/list-aware-explanations/blob/main/pics/screenshot.PNG

[4] J. Kislinger, Full-page recommender: A modular framework for multi-carousel recommendations, in: Proceedings of the Nineteenth ACM Conference on Recommender Systems, RecSys '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 1479–1484. URL: https://doi.org/10.1145/3705328.3748753. doi:10.1145/3705328.3748753.

[5] N. Felicioni, M. Ferrari Dacrema, F. B. Perez Maurera, P. Cremonesi, et al., Measuring the ranking quality of recommendations in a two-dimensional carousel setting, in: Italian Information Retrieval Workshop, volume 2947, CEUR-WS, 2021, pp. 1–14.

[6] H. Ding, B. Kveton, Y. Ma, Y. Park, V. Kini, Y. Gu, R. Divvela, F. Wang, A. Deoras, H. Wang, Trending now: Modeling trend recommendations, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 294–305. doi:10.1145/3604915.3608810.

[7] L. Ma, N. Sinha, P. Vajge, J. H. D. Cho, S. Kumar, K. Achan, Event-based product carousel recommendation with query-click graph, in: 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 4119–4125. doi:10.1109/BigData52589.2021.9671649.

[8] L. Briand, T. Bontempelli, W. Bendada, M. Morlon, F. Rigaud, B. Chapus, T. Bouabça, G. Salha-Galvan, Let's get it started: Fostering the discoverability of new releases on deezer, in: N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, I. Ounis (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2024, pp. 286–291.

[9] M. Ferrari Dacrema, N. Felicioni, P. Cremonesi, Optimizing the selection of recommendation carousels with quantum computing, in: Proceedings of the 15th ACM Conference on Recommender Systems, RecSys '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 691–696. doi:10.1145/3460231.3478853.

[10] S. Vashishtha, A. Kumar, L. Morishetti, K. Nag, K. Achan, Leveraging user-generated reviews for recommender systems with dynamic headers, in: ECAI 2024, IOS Press, 2024, pp. 4734–4740. doi:10.3233/faia241071.

[11] B. Edizel, S. H. Koppuravuri, M. Gannaway, P. Das, K. Ahmadi, Harnessing the power of graph neural networks for personalized rail recommendations, in: 2024 IEEE International Conference on Big Data (BigData), 2024, pp. 2307–2313. doi:10.1109/BigData62323.2024.10825079.

[12] B. Rahdari, P. Brusilovsky, Under the hood of carousels: Investigating user engagement and navigation effort in multi-list recommender systems, in: Proceedings of the 30th International Conference on Intelligent User Interfaces, IUI '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 1485–1498. doi:10.1145/3708359.3712130.

[13] C. Musto, F. Narducci, P. Lops, M. De Gemmis, G. Semeraro, Explod: A framework for explaining recommendations based on the linked open data cloud, in: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 151–154. doi:10.1145/2959100.2959173.

[14] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (2015). doi:10.1145/2827872.

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

[16] Q.-T. Truong, A. Salah, T.-B. Tran, J. Guo, H. W. Lauw, Exploring cross-modality utilization in recommender systems, IEEE Internet Computing 25 (2021) 50–57. doi:10.1109/MIC.2021.3059027.

[17] G. Balloccu, L. Boratto, G. Fenu, M. Marras, Post processing recommender systems with knowledge graphs for recency, popularity, and diversity of explanations, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22, Association for Computing Machinery, 2022, p. 646–656. doi:10.1145/3477495.3532041.