Explanations that are Intrinsic to Recommendations

Arpit Rana
Insight Centre for Data Analytics
University College Cork, Ireland
arpit.rana@insight-centre.org

Derek Bridge
Insight Centre for Data Analytics
University College Cork, Ireland
derek.bridge@insight-centre.org

ABSTRACT

Explanations can give credibility to recommendations and help users to make better choices. In current recommender systems, explanation is a step that comes after recommendation. In this paper, we describe an approach that turns recommender systems on their head. In our approach, which we call *Recommendation-by-Explanation* (*r-by-e*), the system constructs a reason, or explanation, for recommending each candidate item; then it recommends those candidate items that have the best explanations. By unifying recommendation and explanation, *r-by-e* finds relevant recommendations with explanations that have a high degree of fidelity.

We present the results of an offline experiment using a movie recommendation dataset. We show that *r-by-e* achieves higher precision than a comparable recommender, while both produce recommendations with roughly equal levels of diversity and serendipity.

We also present the results of deploying a web-based system through which we have conducted two user trials. In one trial, we evaluate recommendation quality. Participants in this trial found *r-by-e*'s recommendations to be more diverse, serendipitous and relevant than those of the competitor system. In another trial, we evaluate explanation quality. We used a re-rating task: users rated recommendations initially in the case where they were given only the explanation and not the identity of the movie, and then re-rated in the case where they were given information about the recommended movie. We found a stronger correlation between the pairs of ratings in the case of *r-by-e*. This suggests that *r-by-e*'s explanations allow users to make more accurate judgments about the quality of recommended items.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Human-centered computing → User studies**;

KEYWORDS

Explanation; Recommendation; User Trial

ACM Reference Format:
Arpit Rana and Derek Bridge. 2018. Explanations that are Intrinsic to Recommendations. In *UMAP '18: 26th Conference on User Modeling, Adaptation and Personalization, July 8–11, 2018, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3209219.3209230

UMAP '18, July 8–11, 2018, Singapore, Singapore
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5589-6/18/07...$15.00
https://doi.org/10.1145/3209219.3209230

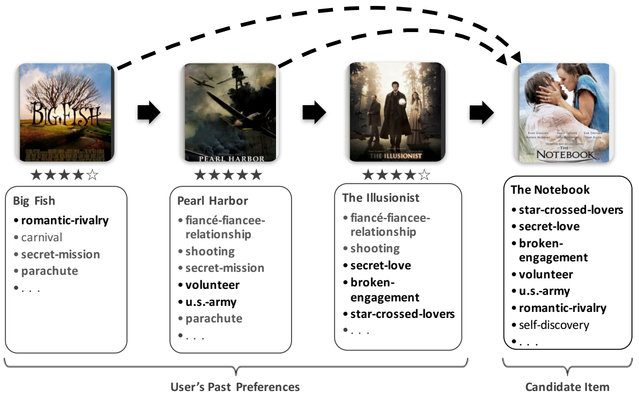


Figure 1: An Explanation Chain.

1 INTRODUCTION

Recommender systems provide explanations to help the user understand the rationale for a recommendation and to help her make a decision [1]. Conventionally, computing recommendations and generating corresponding explanations are two separate, sequential processes. This separation is one cause of low fidelity between the explanations and the operation of the recommender.

In this paper, we present Recommendation-by-Explanation *r-by-e*, in which explanation is intrinsic to recommendation. In *r-by-e*, the system constructs a reason, or explanation, for recommending each candidate item; then it recommends those candidates that have the best explanations. *r-by-e*'s explanations take the form of what we call *Explanation Chains*. Figure 1 shows an example of an Explanation Chain in the movie domain. The rightmost item (in this case, *The Notebook*) is the candidate for recommendation to the user, and will typically not already be in the user's profile. The other items (*Big Fish*, *Pearl Harbour* and *The Illusionist*) form the chain. They are drawn from positively-rated items in the user's profile and are intended to support recommendation of the candidate item. Pairs of successive items in a chain satisfy a local constraint in the form of a similarity threshold; additionally, each item in the chain satisfies a global constraint in the form of a threshold on the level of coverage it contributes towards features of the candidate item. For example, *Big Fish* has the keywords: `secret-mission` and `parachute` in common with *Pearl Harbour*, as well as the keyword `romantic-rivalry` in common with *The Notebook*.

We believe that *r-by-e* has the following characteristics:

- *Unified approach:* It is a unified approach that combines the processes of computing recommendations and generating corresponding explanations.

- *Fidelity:* By unifying recommendation and explanation, there is a guaranteed level of fidelity between explanations and the operation of the recommender.
- *Diversity and serendipity:* The approach uses hyperparameters whose values can be adjusted to loosen or tighten constraints between items in the chain and thus increase or decrease the diversity and serendipity of the recommendations.

We introduced *r-by-e* and the concept of Explanation Chains in [16]. In this paper, our contributions are: we present the algorithms in detail; we present more comprehensive results for an offline experiment; and we present previously-unpublished user trials.

2 RELATED WORK

Explanations of recommendations vary in many ways. They may vary in their goals: they may be intended to help the user make a better decision (effectiveness), change the user's behaviour (persuasion), make a system more correctable (scrutability), and so on [20]. In our work, we are interested in effectiveness, which is why one of our user trials is a re-ranking task.

Explanations of recommendations often relate the recommended item to the user through *intermediary entities*, which may be other users, other items, or features [1, 21]:

- user-based explanations say that an item is being recommended because users who are similar to the active user liked it, e.g. in [8], the explanation is a histogram of the active user's neighbours' ratings of the item;
- item-based explanations say that the item is being recommended because the user liked similar items, e.g. [3, 11];
- feature-based explanations say that the recommended item has features that the user likes, where the features might be, e.g., attribute-value pairs [19], linked data [14], item content [2], user-generated tags [7, 21], or features and opinions mined from user reviews [4, 5].

However, in the case of item-based explanations, we often want to show why the items in the explanation are similar to the recommended item, and this is typically done by showing the features that they have in common. Since these item-based explanations combine items and features, they have also been designated as 'hybrid' explanations [15]. Explanation Chains are of this kind: they are item-based but they expose item relationships through features.

More generally, the explanations of systems in Artificial Intelligence are categorized as white-box explanations and black-box (or model-agnostic) explanations. White-box explanations are built from traces of the system's reasoning. For example, if we have a recommender that makes recommendations by finding items liked by the active user's nearest neighbours, then a histogram of the neighbours' ratings [8] is a white-box explanation. By contrast, black-box explanations have no knowledge of how the system produced its decision. The explanations are post-hoc rationalizations. For example, the LIME system explains classification decisions by interrogating the classifier to obtain a dataset from which LIME builds a distinct explanation model [17]. Some black-box explanations even resort to using data that was not used by the decision-making system. In [18], for example, recommendations are made by matrix factorization on a ratings matrix but the recommendations are explained using topic models that are mined from textual data associated with the items but not used by the recommender. By contrast, Explanation Chains are white-box explanations.

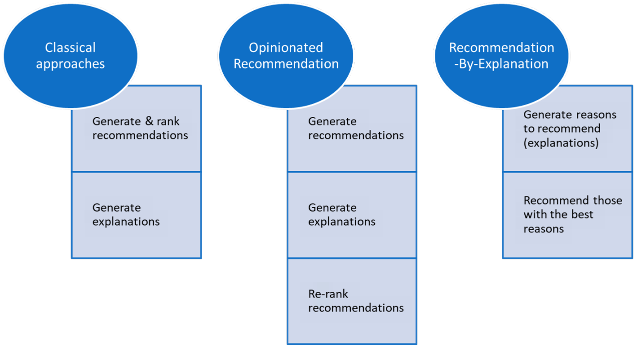

Classical approaches
Generate & rank recommendations
Generate explanations
Opinionated Recommendation
Generate recommendations
Generate explanations
Re-rank recommendations
Recommendation -By-Explanation
Generate reasons to recommend (explanations)
Recommend those with the best reasons

Figure 2: Ways of computing recommendation explanations.

This raises the issue of fidelity [10] (also called objective transparency [7]): the extent to which the explanation reveals the logic of the underlying recommender. In an experiment with a music recommender, Kulesza et al. found that the more that explanations were both sound and complete with respect to the recommender, the greater the users' trust in the recommender and the better their understanding [10]. Arguably, black-box systems cannot achieve fidelity. (The LIME system[17], which is model-agnostic, claims to achieve 'local fidelity', but this is not the same concept.) Recommendation-by-Explanation seeks to achieve quite high fidelity since, in *r-by-e*, explanation is intrinsic to recommendation.

Finally, it seems obvious that a recommender should first produce its recommendations and then seek to build explanations for them. This is the classic approach depicted leftmost in Figure 2. All of the systems that we have cited so far work in this way. A new approach, Opinionated Recommendation, shown in the middle of Figure 2, modifies this a little [12, 13]: the system finds some recommendations, it generates explanations for the recommendations, it scores the explanations, and it re-ranks the recommendations based on their explanation scores before showing them to the user. Recommendation-by-Explanation is shown rightmost in Figure 2: it finds explanations for the candidates and recommends the candidates that have the best explanations.

3 RECOMMENDATION-BY-EXPLANATION

Recommendation by Explanation (*r-by-e*) is a novel approach that unifies recommendation and explanation: it computes recommendations by generating and ranking corresponding personalized explanations in the form of Explanation Chains. Here we explain in detail how *r-by-e* constructs the chains for candidate items and select the n that it will recommend.

3.1 *r-by-e* top-n recommendation

Let $\mathbb{I}$ be the set of all items. *r-by-e* works in a scenario of implicit ratings, where the user's profile $P \subseteq \mathbb{I}$ is the set of items that she likes. *r-by-e* will recommend up to n items from a set of candidate items, $I \subseteq \mathbb{I}$. Candidates I can be defined in whatever way is suited

Algorithm 1 *r-by-e* top-n recommendation.

```
Input: $n$, number of recommendations
       $I$, set of candidate items
       $P$, user's profile
       $\theta$, similarity threshold
       $\epsilon$, marginal gain threshold
Output: $L^*$, ranked list of top-$n$ Explanation Chains.
function Recommend($n, I, P, \theta, \epsilon$)
    $L \leftarrow [\,]$
    for each $i \in I$ do
        $C \leftarrow$ GenerateChain($i, P, \theta, \epsilon$)
        if $|C| > 0$ then
            append $\langle C, i \rangle$ onto $L$
    return SelectChains($L, n$)
```

to the task in hand. Typically, for example, they will be items not already in P. But they could be further constrained by contextual factors such as time or location, e.g. recently-released movies, TV shows to be broadcast in the next few hours, or restaurants in the vicinity of the user. Another way to obtain candidates is to take the top-n' recommendations of another recommender system ($n' >> n$); in this case, *r-by-e* will filter and re-rank the other system's recommendations. In our experiments later in this paper, we define I to be items that are not in the user's profile but which do have at least a certain degree of similarity to the user's profile, $I = \left\{ i \in I \setminus P \mid \text{sim}(f_i, f_p) > \theta, \exists p \in P \right\}$. Here f_i and f_p denote the features of items i and p, and we define sim as Jaccard similarity.

For each candidate item, *r-by-e* generates an Explanation Chain and then it selects the top n of those chains to recommend to the user; see Algorithm 1.

3.2 Explanation chain generation

Given a candidate item, *r-by-e* works backwards to construct a chain: starting with the candidate item, it finds predecessors, greedily selects one, finds its predecessors, selects one; and so on; see Algorithm 2. The predecessors of an item are all its neighbours in the item-item similarity graph that satisfy four conditions: (a) they are members of the user's profile P; (b) they are not already in this chain; (c) their similarity to the subsequent item in the chain exceeds a similarity threshold θ; and (d) their reward (see below) exceeds a marginal gain threshold ϵ. When there are no further predecessors, the chain is complete.

At each step, the predecessor that gets selected is the one with the highest *reward*. The reward $\text{rwd}(p, i, C)$ of adding predecessor p to partial chain C that explains candidate item i is given by:

$$rwd(p, i, C) = \frac{\left|(f_p \setminus \text{covered}(i, C)) \cap f_i\right|}{|f_i|} + \frac{\left|(f_p \setminus \text{covered}(i, C)) \cap f_i\right|}{\left|f_p\right|} \tag{1}$$

Here again f_i and f_p denote the features of items i and j. $\text{covered}(i, C)$ is the set of features of candidate i that are already covered by members of the chain C, i.e. $\text{covered}(i, C) = \bigcup_{j \in C} f_j \cap f_i$. Then the first term in the definition of $\text{rwd}(p, i, C)$ measures p's coverage of those features of i that are not yet covered by the chain. The second term in the definition measures the same but with respect to the size of f_j rather than the size of f_i and therefore assures j's fitness to explain the candidate by penalizing items that have high coverage simply by virtue of having more features.

Algorithm 2 Explanation Chain generation.

```
Input: $i$, a candidate item
       $P$, user's-profile
       $\theta$, similarity threshold
       $\epsilon$, marginal gain threshold
Output: $C$, an Explanation Chain $C$ for candidate $i$.
function GenerateChain($i, P, \theta, \epsilon$)
    $C \leftarrow [\,]$
    $sum\_rwds = 0$
    $j \leftarrow i$
    while True do
        $J \leftarrow \{p \in P \setminus C \mid \text{sim}(f_j, f_p) > \theta \wedge \text{rwd}(p, i, C) > \epsilon\}$
        if $|J| = 0$ then
            return $C$
        $j = \arg\max_{p \in J} \text{rwd}(p, i, C)$
        append $j$ onto $C$
        $sum\_rwds = sum\_rwds + \text{rwd}(j, i, C)$
```

Algorithm 3 Chain selection.

```
Input: $L$, list of Explanation Chains for different candidate items
       $n$, number of recommendations
Output: $L^*$, ranked list of top-$n$ Explanation Chains.
function SelectChains($L, n$)
    if $|L| \leq n$ then
        sort $L$ using $score$
        return $L$
    $L^* \leftarrow [\,]$
    while $|L^*| < n$ do
        $\langle C, i \rangle^* = \arg\max_{\langle C, i \rangle \in L} \text{score}(\langle C, i \rangle, L^*)$
        append $\langle C, i \rangle^*$ onto $L^*$
        remove $\langle C, i \rangle^*$ from $L$
    return $L^*$
```

3.3 Chain selection

After constructing a chain C for each candidate item i, we must select the top-n chains so that we can recommend n items to the user, along with their explanations. This is done iteratively based on a chain's total coverage of the candidate item's features and the chain's dissimilarity to other chains already included in the top-n; see Algorithm 3.

Specifically, we score $\langle C, i \rangle$ relative to a list of all the items that appear in already-selected chains L^* using the following:

$$\text{score}(\langle C, i \rangle, L^*) = \frac{sum\_rwds}{|C| + 1} + \frac{\left|C \setminus \bigcup_{j \in L^*} j\right|}{|C| + 1} \tag{2}$$

Here, the first term is the sum of the rewards of the items in the chain divided by its length plus 1 (so as to include candidate item i). It gives higher scores to chains that do a better job of covering the features of their candidate item. The second term gives higher

Table 1: Results of the Offline Experiment. All of the *r-by-e* results are statistically significant with respect to *CB*-$|C|$ (Wilcoxon signed rank test with $p < 0.05$) except the one shown in italics.

Recommender	θ & ϵ optimized for	Precision	$Div_{keywords}$	Div_{genres}	Surprise	Novelty	Coverage	% of explanations of size 2–4
r-by-e	Precision	0.1089	0.9352	0.7631	0.7834	0.3771	0.1358	0.2868
CB-$|C|$		0.0701	0.9091	0.7272	0.8135	0.4179	0.1509	0.2719
r-by-e	$Div_{keywords}$	0.0370	0.9760	0.8177	0.8886	*0.5404*	0.6014	0.3156
CB-$|C|$		0.0087	0.9736	0.7834	0.9370	0.5365	0.8700	0.6534
r-by-e	% of explanations of size 2–4	0.0677	0.9626	0.8007	0.8635	0.4756	0.3541	0.7598
CB-$|C|$		0.0097	0.9711	0.7820	0.9336	0.5119	0.8976	0.7506

scores to a chain if its members are not also members of already-selected chains and hence encourages the final recommendation list to cover as many items in the user's profile as possible. (Note that the second term is about coverage of *items* that appear in already-selected chains, not their features.) We have found this term to have the effect of diversifying the recommendation list.

4 OFFLINE EXPERIMENT

We ran an offline experiment to evaluate *r-by-e*'s performance. We compare it with a content-based recommender, which works as follows. Given candidate item i, it finds the items in P whose similarity to i exceeds θ; it takes the k of these neighbours with highest similarity; it scores the candidate by taking a similarity-weighted average of their ratings. It recommends the n candidates with highest scores.

The main difference between the two recommenders is that the content-based recommender relies on similarity relationships between members of P and the candidate item, whereas *r-by-e*, by requiring consecutive members of chains to be similar to each other, additionally takes into account similarity relationships between members of P themselves. We wanted this experiment to reveal the effect of this difference. So we otherwise tried to ensure that the two systems were as similar as possible. They both use the same item features (keywords, see below), and they both use the same similarity measure (Jaccard). For the content-based system, we chose to set k in a dynamic fashion, as follows. If, for candidate item i, *r-by-e* generates a chain of length $|C|$, then the content-based system uses $k = |C|$ when it scores that candidate item. It follows that k is set dynamically: different candidates may have different values for k. We designate this system *CB*-$|C|$, using a name that emphasizes that dynamically $k = |C|$.

Although in this paper we only report results that compare *r-by-e* with *CB*-$|C|$, in [16], we have also compared with a more conventional content-based system with a fixed value for k (set at 7 by a hyperparameter optimization process). The results in that paper show that *CB*-$|C|$ and *CB-7* have quite similar performance.

4.1 Experiment settings

We used the hetrec2011-movielens-2k dataset[1] augmented by keywords from IMDb[2]. The dataset comprises 2113 users, 5992 movies, 80639 keywords, and over half a million ratings. On average, each movie has 107 keywords (ranging from 2 to 626) and has non-zero similarity with 77% of the other movies.

In *r-by-e*, user profiles simply contain items the user likes. We treated ratings of 4 and 5 as 'likes', so user u's profile is given by $\{i \mid r_{u,i} \geq 4\}$. We split each user's ratings into training, validation and test sets in the ratio $60 : 20 : 20$, repeated five times.

On the test sets, for $n = 10$, we measured precision, diversity, surprise, novelty and coverage using definitions of these measures given in Section 7 of [9]. In the case of diversity, we use the measure that [9] denotes by $obj_{diversity}$, which is the average all-pairs distance between items in the recommendation list. For distance, we use the complement of Jaccard similarity, and we measure it both using all keywords (designated $Div_{keywords}$) and using just genres (designated Div_{genres}). In the case of surprise, we use the measure that [9] denotes by $obj_{cont}^{surprise}$, which is based on minimum distances between recommended items and the user's profile.

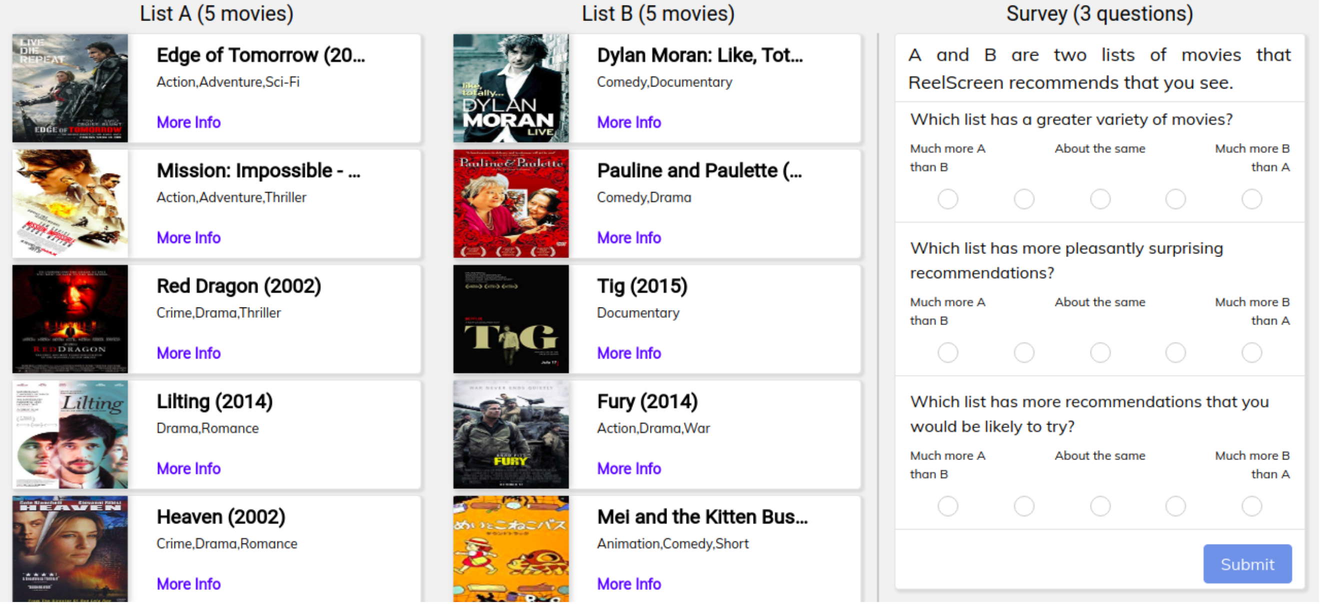
We also suspect that users will find an explanation to be easily intelligible only if it is fairly small (chains or sets of neighbours of size 2–4 items), so we recorded the percentage of explanations that were of this size.

We experimented with five different values of each of the similarity threshold (θ) and the marginal gain threshold (ϵ): $[0.03, 0.06, 0.09, 0.12, 0.15]$, giving 25 configurations of *r-by-e*. When choosing the best configuration, there is an issue about what to optimize. It makes sense, for example, to choose the configuration that optimizes precision on the validation sets. But it could be interesting to choose configurations that optimize other criteria. Therefore, we also show results in the case where we choose the configuration that optimizes for diversity on the validation set, and in the case where we choose the configuration that maximizes the percentage of explanations of size 2–4.

4.2 Experiment results

Table 1 summarizes the results. For the most part differences in the results for *r-by-e* and *CB*-$|C|$ are small but, since standard deviations are low, in all but one case they are statistically significant. In two cases, differences are larger: *r-by-e* has better precision and *CB*-$|C|$ has better catalogue coverage. It is noteworthy that *r-by-e* can produce more accurate recommendations without sacrificing diversity and surprise.

[1] https://grouplens.org/datasets/hetrec-2011/
[2] http://www.imdb.com



List A (5 movies)

Edge of Tomorrow (20...
Action,Adventure,Sci-Fi
More Info

Mission: Impossible - ...
Action,Adventure,Thriller
More Info

Red Dragon (2002)
Crime,Drama,Thriller
More Info

Lilting (2014)
Drama,Romance
More Info

Heaven (2002)
Crime,Drama,Romance
More Info

List B (5 movies)

Dylan Moran: Like, Tot...
Comedy,Documentary
More Info

Pauline and Paulette (...
Comedy,Drama
More Info

Tig (2015)
Documentary
More Info

Fury (2014)
Action,Drama,War
More Info

Mei and the Kitten Bus...
Animation,Comedy,Short
More Info

Survey (3 questions)

A and B are two lists of movies that ReelScreen recommends that you see.

Which list has a greater variety of movies?
Much more A than B — About the same — Much more B than A
○ ○ ○ ○ ○

Which list has more pleasantly surprising recommendations?
Much more A than B — About the same — Much more B than A
○ ○ ○ ○ ○

Which list has more recommendations that you would be likely to try?
Much more A than B — About the same — Much more B than A
○ ○ ○ ○ ○

Submit

Figure 3: A screenshot showing top-5 recommendations from the two recommenders and survey questions.

5 USER TRIALS

We also built a web-based system in order to conduct user trials, again comparing *r-by-e* with *CB*-|*C*| using the hyperparameter values (θ and ϵ) that optimized the percentage of explanations of size 2–4. *r-by-e* is, above all, a recommender and so we designed one trial to measure recommendation quality as well as a trial to measure explanation quality. In total, 190 people attempted the trials. The majority of them were undergraduate and postgraduate students recruited online from universities in India and Ireland. To increase the chances of user familiarity with the movies, the web-based recommmenders use only movies released between the years 2000 and 2016 inclusive: 3668 of the 5992 movies in the dataset used in the offline experiment.

Each participant begins by creating a user profile containing at least 30 movies. The instructions were that the movies should be the ones the user likes. The user interface offers both a scrollable grid of movies and a search box to enable her to find these movies.

We assigned half the participants to the recommendation trial and the other half to the explanation trial. Of the 190, only 115 completed all parts of the trial to which they were assigned.

5.1 Recommendation trial

5.1.1 Experiment settings. The recommendation trial is a within-subjects trial: users see two lists of recommendations, one list from *r-by-e* and the other from *CB*-|*C*| and they answer questions that compare the quality of the two lists [6]; see Figure 3. Lists have length 5 and are sorted in decreasing order of recommender scores.

Before displaying the recommendations, we ensured that the two lists contained different movies. Each movie that was common to both lists was removed and the next best recommendations from the top-10 were added to the end of the lists. (If it was not possible to create two different lists of length 5 from the top-10 recommendations, the user's responses to the survey were discarded. We did this to avoid skewing responses about the diversity of recommendations: shorter lists are less likely to be diverse.)

Table 2: Results of the Recommendation Trial.

User's opinion	Diversity	Serendipity	Satisfaction
Much more *r-by-e*	14	15	28
More *r-by-e*	14	11	13
About the same	14	23	8
More *CB*-|*C*|	11	10	7
Much more *CB*-|*C*|	13	7	10

For half the users, the list on the left ('List A') came from *r-by-e* and the list on the right ('List B') from *CB*-|*C*|; for the other half of the users, List A was from *CB*-|*C*| and List B from *r-by-e.*

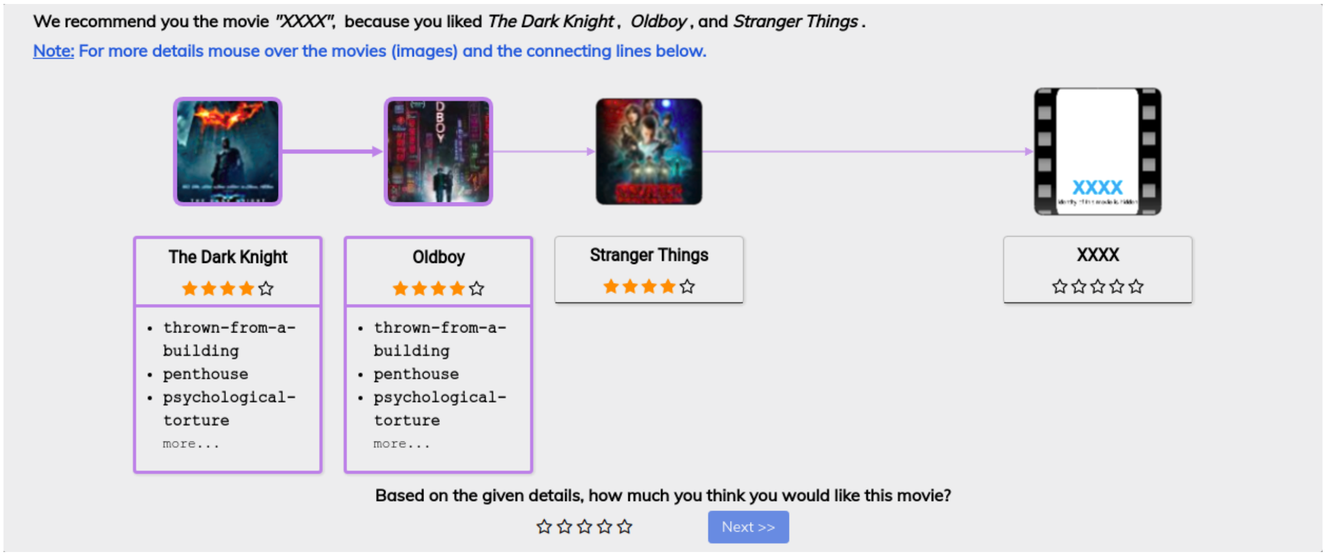
Participants were required to answer three questions:

- Diversity: Which list has a greater variety of movies?
- Serendipity: Which list has more pleasantly surprising recommendations?
- Satisfaction: Which list has more recommendations that you would be likely to try?

Their answers were on a 5-point: Much more List A than List B; More List A than List B; About the Same; More List B than List A; and Much more List B than List A.

5.1.2 Experiment results. Sixty-six participants completed this trial. Table 2 summarizes their responses.

- *Diversity question:* 42.4% of participants found *r-by-e* recommendations to be much more or more diverse than *CB*-|*C*| recommendations, 21.2% found the recommendation lists to



We recommend you the movie *"XXXX"*, because you liked *The Dark Knight*, *Oldboy*, and *Stranger Things*.

Note: For more details mouse over the movies (images) and the connecting lines below.

The Dark Knight ★★★★☆

- thrown-from-a-building
- penthouse
- psychological-torture

more...

Oldboy ★★★★☆

- thrown-from-a-building
- penthouse
- psychological-torture

more...

Stranger Things ★★★★☆

XXXX ☆☆☆☆☆

Based on the given details, how much you think you would like this movie?

☆☆☆☆☆

Figure 4: A screenshot of an Explanation Chain. The user has moused over the arrow that connects the first two movies, which causes the system to bring up boxes of keywords that these two movies have in common.

be equally diverse, leaving 36.4% finding *CB*-|*C*| to be much more or more diverse.

- *Serendipity question:* 39.4% of participants found *r-by-e* recommendations to be much more or more pleasantly surprising, 34.8% found the recommendation lists to be equally surprising, leaving 25.8% finding *CB*-|*C*| to be much more or more surprising.
- *Satisfaction question:* 62.1% of participants found *r-by-e* recommendations to be ones they would be much more or more likely to try, 12.1% found the recommendations to be equally worthy of trying, leaving 25.8% finding *CB*-|*C*| to be much more or more worth trying.

On all criteria *r-by-e* produced the better recommendation lists. However, only in the case of the satisfaction question was this statistically significant. (We used two-tailed proportion tests with significance level $p_0 = 0.05$. The null hypothesis was that those preferring *r-by-e* was equal to those preferring *CB*-|*C*|, i.e., ignoring those who thought the two lists were about the same.)

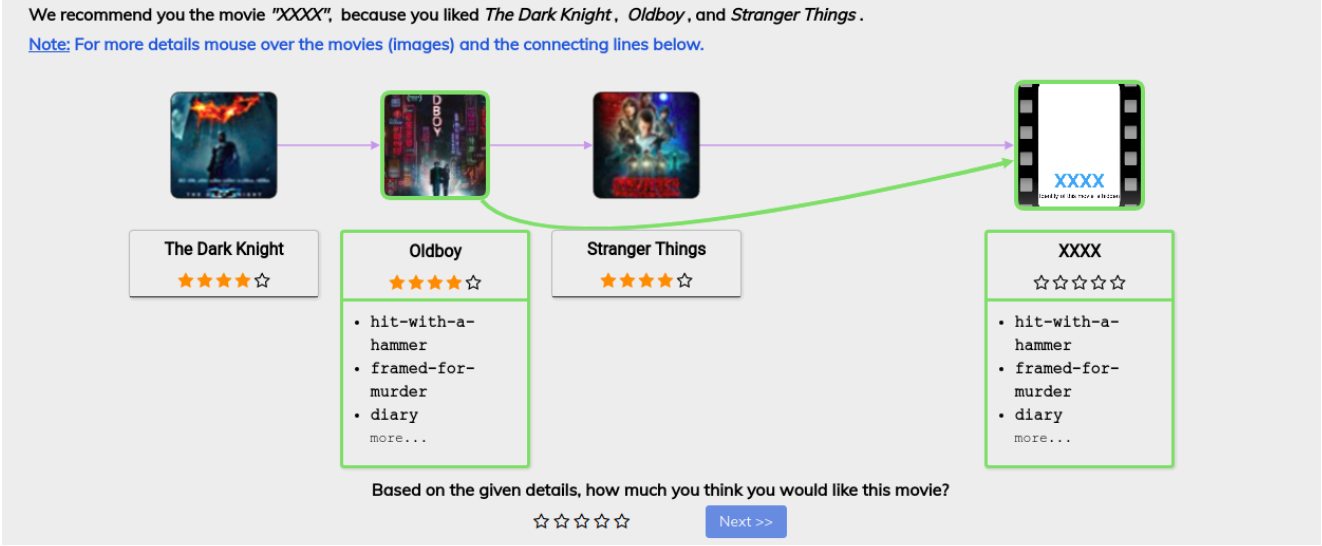
5.2 Explanation Trial

Users who were directed to this trial participated in a re-rating task. Re-rating tasks are an established method of evaluating explanation quality when the goal of the explanation is effectiveness: helping users make better decisions [1, 7]. A user is initially asked to rate a recommendation in the case where she is given only the explanation and not the identity of the movie. This is called the *explanation-rating*. The user is asked later to re-rate the recommended item in the case where she is given information about the item, including its identity. This is called the *actual-rating*. An effective explanation is one where the explanation-rating is close to the actual-rating: the explanation allowed the user to predict how much she would like the item. Effective explanations will be ones for which (a) μ_d (the mean difference between explanation-ratings and corresponding actual-ratings) is close to zero; (b) σ_d (their standard deviation) is small; and (c) r (their Pearson correlation) is highest.

5.2.1 Experiment settings. In our Explanation Trial, we used *r-by-e* to generate the top-n recommendations for the user, $n = 5$. Each of these, of course, came with an explanation in the form of Explanation Chain, C. For the *same* movies, we then generated the explanations that the *CB*-|*C*| system would have produced had it made these recommendations: the $k = |C|$ most similar movies in the user's profile. If the set of movies in *r-by-e*'s chain and *CB*-|*C*|'s neighbours were identical, we replaced the recommendation by the next best recommendation from *r-by-e*'s top-10 recommendations. (In contrast to the Recommendation Trail, in this trial, where we were not able to make 5 recommendations from the top-10, we did not discard the user's responses: we are comparing the effectiveness of pairs of corresponding explanations not, for example, the diversity of lists of recommendations.)

For n recommendations, we have $2n$ explanations to show to the user: two of each kind. We show them to the user in a random order and with the identity of the recommended movie redacted (shown as "XXXX"). Explanation Chains were displayed in the fashion shown in Figures 4 and 5: arrows connect a movie to its successor in the chain. *CB*-|*C*|'s explanations (sets of neighbours, rather than chains) were displayed in the fashion shown in Figure 6: arrows connect each movie to the recommended movie. In both cases, the user can mouse over parts of the explanation, which causes the system to display keywords that movies have in common (see the captions of the Figures). A maximum of three keywords is displayed in any box, and they are selected by their *tf-idf* scores.

As can be seen at the foot of Figures 4, 5 and 6, we asked the user to supply an explanation-rating (1-5 stars): how much they thought they might like the movie based only on the explanation.



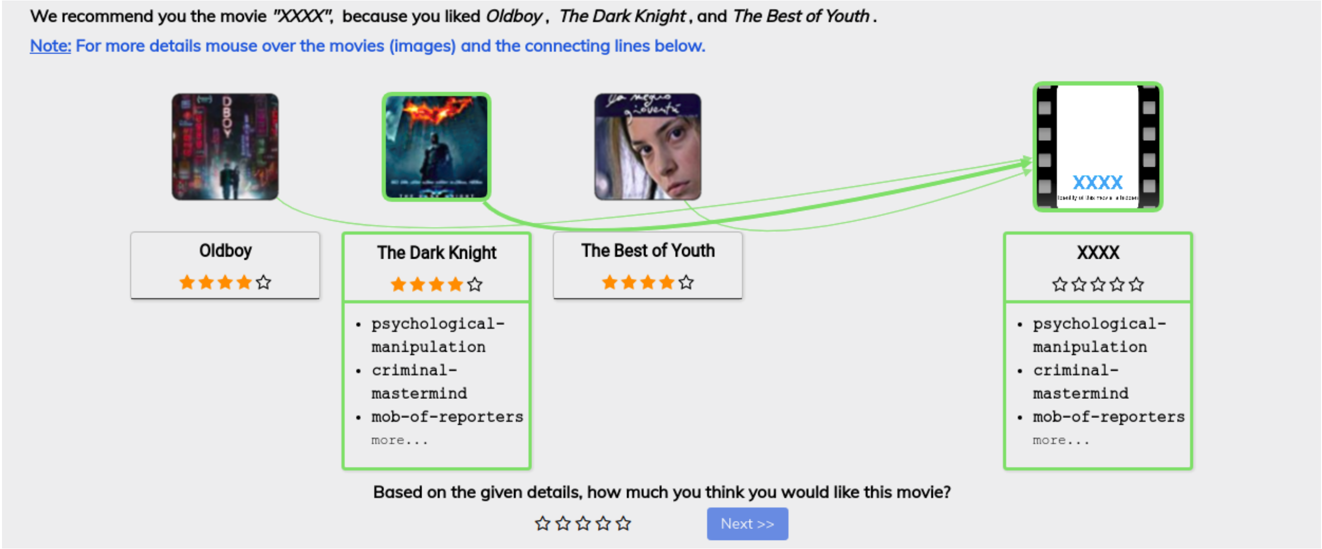
We recommend you the movie *"XXXX"*, because you liked *The Dark Knight*, *Oldboy*, and *Stranger Things*.

Note: For more details mouse over the movies (images) and the connecting lines below.

Based on the given details, how much you think you would like this movie?

☆☆☆☆☆ Next >>

Figure 5: A screenshot of an Explanation Chain. The user has moused over the icon for the second movie, which causes the system to display an arrow between that movie and the recommended movie and to bring up boxes of keywords that these two movies have in common.



We recommend you the movie *"XXXX"*, because you liked *Oldboy*, *The Dark Knight*, and *The Best of Youth*.

Note: For more details mouse over the movies (images) and the connecting lines below.

Based on the given details, how much you think you would like this movie?

☆☆☆☆☆ Next >>

Figure 6: A screenshot of a *CB*-$|C|$ explanation. The user has moused over the icon for the second movie, which causes the system to increase the width of the arrow between that movie and the recommended movie and to bring up boxes of keywords that these two movies have in common.

After the user has given these $2n$ ratings, the system then shows her in a random order each of the n recommended movies again. This time, the identity of the movie is not redacted but no explanation is shown. Instead, we show genre, plot synopsis, main cast members, directors, writers, duration, and release date. Again we ask the user for a rating (the so-called actual-rating) to indicate how much she thinks she will like the movie.

Note that, although the user has rated the same movie three times, nothing in the on-screen instructions makes this apparent.

5.2.2 Experiment results. Forty-nine participants completed this trial: it is quite onerous and more participants abandoned it part-way through than did for the other trial. In total, we obtained 597 ratings, this being three ratings for 199 recommended movies. (As

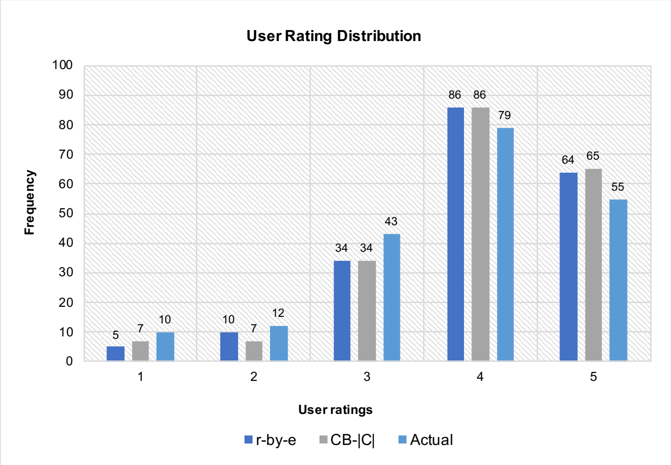


Figure 7: Ratings from the Explanation Trial.

Table 3: Ratings from the Explanation Trial.

Rating type	μ	σ	r
Actual	3.7889	1.0711	–
r-by-e	3.9749	0.9610	0.4855
CB-|C|	3.9799	0.9794	0.2367

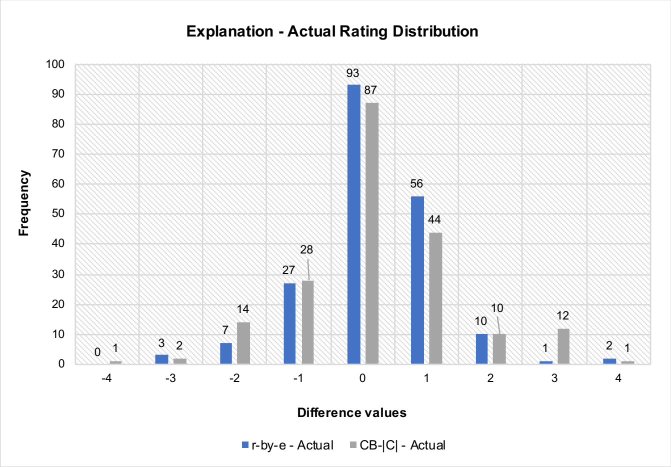


Figure 8: Differences in ratings from the Explanation Trial.

we mentioned above, we did not always have 5 recommendations per user, e.g. where explanations contained identical movies).

Figure 7 shows the distribution of the users' ratings; Table 3 gives summary statistics. We can see that users mostly think they will like the movies that the system recommends, both when they see explanations only and when they see movie identity. For the differences between explanation-ratings and actual-ratings, Figure 8 shows the distribution of values and Table 4 gives summary statistics. The mean difference between *r-by-e* ratings and actual ratings is 0.1859; for *CB*-|*C*|, it is 0.1910. Hence, both kinds of explanations cause users to overestimate their actual-ratings. Using a two-tailed paired t-test ($p_0 = 0.05$), we observed that in this study, i) the difference between *r-by-e*-ratings and actual-ratings are statistically different; ii) the differences between *CB*-|*C*|ratings and actual-ratings are also statistically significant; and iii) *r-by-e*-ratings and *CB*-|*C*|-ratings are not statistically different. In terms of μ_d and σ_d, then, neither kind of explanation is better than the other. But there is still the question of correlation with the actual-ratings.

Table 4: Differences in ratings from the Explanation Trial.

Explanation type	μ_d	σ_d	95% Conf. Int.
r-by-e	0.1859	1.0350	(0.0412, 0.3306)
CB-|C|	0.1910	1.2688	(0.0136, 0.3683)

Table 3 shows r, the Pearson correlation between explanation-ratings and actual-ratings. We see that *r-by-e*-ratings are better correlated with actual-ratings. We calculated the probability of getting this correlation due to chance to be 0 in both cases.

6 CONCLUSIONS

Recommendation-by-Explanation (*r-by-e*) unifies recommendation and explanation. It computes explanations first and then recommends the items with the best explanations. Its explanations take the form of Explanation Chains, which are sequences of items from the user's profile. There are local relationships between consecutive items in the chain: they have some similarity to each other. There are also global relationships between items in the chain and the recommended item: the items are chosen in an effort to cover the features of the recommended item. The items to be recommended are selected based both on this feature coverage relationship and the degree of coverage of item's in the user's profile.

This paper presents experiments to evaluate *r-by-e*. An offline trial shows the approach to have better precision than a closely-related content-based recommender, while remaining competitive on measures of diversity and serendipity.

We use a web-based system to conduct user trials. The Recommendation Trial shows that *r-by-e* produces recommendations that are more diverse and serendipitous than those of a content-based recommender (although not statistically significantly so) and with statistically significantly higher levels of satisfaction. The Explanation Trial is a re-rating task for measuring explanation effectiveness. Users rate an item given only an explanation (with its identity hidden) and later re-rate when given the identity without the explanation. The correlation between these pairs of ratings is much greater in the case of *r-by-e* explanations.

ACKNOWLEDGMENTS

We thank Kapil Bhagchandani & Arpit Jain for help with web design. This paper emanates from research supported by a grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 which is co-funded under the European Regional Development Fund.

REFERENCES

[1] Mustafa Bilgic and Raymond J Mooney. 2005. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, Vol. 5. 153.

[2] Roi Blanco, Diego Ceccarelli, Claudio Lucchese, Raffaele Perego, and Fabrizio Silvestri. 2012. You should read this! let me explain you why: explaining news recommendations to users. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 1995–1999.

[3] Derek Bridge and Kevin Dunleavy. 2014. If you liked Herlocker et al.âĂŹs explanations paper, then you might like this paper too. In *Joint Workshop on Interfaces and Human Decision Making in Recommender Systems*. 22.

[4] Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. 2016. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 175–182.

[5] Li Chen and Feng Wang. 2017. Explaining Recommendations Based on Feature Sentiments in Product Reviews. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. 17–28.

[6] Michael D Ekstrand, F Maxwell Harper, Martijn C Willemsen, and Joseph A Konstan. 2014. User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 161–168.

[7] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies* 72, 4 (2014), 367–382.

[8] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 241–250.

[9] Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Transactions on Interactive Intelligent Systems* 7, 1 (December 2016), 2:1–2:42.

[10] Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. 2013. Too much, too little, or just right? Ways explanations impact end users' mental models. In *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*. IEEE, 3–10.

[11] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[12] Khalil Muhammad, Aonghus Lawlor, Rachael Rafter, and Barry Smyth. 2015. Great explanations: Opinionated explanations for recommendations. In *International Conference on Case-Based Reasoning*. Springer, 244–258.

[13] Khalil Muhammad, Aonghus Lawlor, and Barry Smyth. 2016. On the Use of Opinionated Explanations to Rank and Justify Recommendations. In *Proceedings of the FLAIRS Conference*. 554–559.

[14] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2016. ExpLOD: A Framework for Explaining Recommendations Based on the Linked Open Data Cloud. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 151–154.

[15] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. 2012. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery* 24, 3 (2012), 555–583.

[16] Arpit Rana and Derek Bridge. 2017. Explanation Chains: Recommendations by Explanation. In *Procs. of the Poster Track of the 11th ACM Conference on Recommender Systems*, Domonkos Tikk and Pearl Pu (Eds.). CEUR Workshop Proceedings, vol-1905.

[17] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *CoRR* abs/1602.04938 (2016).

[18] Marco Rossetti, Fabio Stella, and Markus Zanker. 2013. Towards explaining latent factors with topic models in collaborative recommender systems. In *Database and Expert Systems Applications (DEXA), 2013 24th International Workshop on*. IEEE, 162–167.

[19] Christian Scheel, Angel Castellanos, Thebin Lee, and Ernesto William De Luca. 2012. The reason why: A survey of explanations for recommender systems. In *International Workshop on Adaptive Multimedia Retrieval*. Springer, 67–84.

[20] Nava Tintarev and Judith Masthoff. 2007. A survey of explanations in recommender systems. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*. IEEE, 801–810.

[21] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 47–56.