The Reason Why: A Survey of Explanations for Recommender Systems

Christian Scheel[1], Angel Castellanos[2],
Thebin Lee[3], and Ernesto William De Luca[3(✉)]

[1] DAI-Labor, Technische Universität Berlin, Berlin, Germany
scheel@dai-lab.de
[2] UNED, Madrid, Spain
acastellanos@lsi.uned.es
[3] Fachhochschule Potsdam, Potsdam, Germany
{lee,deluca}@fh-potsdam.de

Abstract. Recommender Systems refer to those applications that offer contents or items to the users, based on their previous activity. These systems are broadly used in several fields and applications, being common that an user interact with several recommender systems during his daily activities. However, most of these systems are black boxes which users really don't understand how to work. This lack of transparency often causes the distrust of the users. A suitable solution is to offer explanations to the user about why the system is offering such recommendations. This work deals with the problem of retrieving and evaluating explanations based on hybrid recommenders. These explanations are meant to improve the perceived recommendation quality from the user's perspective. Along with recommended items, explanations are presented to the user to underline the quality of the recommendation. Hybrid recommenders should express relevance by providing reasons speaking for a recommended item. In this work we present an attribute explanation retrieval approach to provide these reasons and show how to evaluate such approaches. Therefore, we set up an online user study where users were asked to provide movie feedback. For each rated movie we additionally retrieved feedback about the reasons this movie was liked or disliked. With this data, explanation retrieval can be studied in general, but it can also be used to evaluate such explanations.

Keywords: Hybrid recommender systems · Explanations · Evaluation · Persuasion · Satisfaction · Decision support

1 Introduction

In recent years, recommender systems have seen a steady increase in predictive accuracy in terms of quantifiable measures such as precision, recall, or the root-mean-square error [1]. Generally, the assumption by recommender systems researchers and developers is that the goal of a recommender system is to reach

A. Nürnberger et al. (Eds.): AMR 2012, LNCS 8382, pp. 67–84, 2014.
DOI: 10.1007/978-3-319-12093-5_3

higher levels of accuracy or lower levels of predictive errors. Given this assumption, the recent development indicates progress. We should however ask whether we might be missing a bigger point? Predictive accuracy is only a measure of the algorithmic quality of a system, it does not affect the perception of the user. Keeping this in mind, it is possible that even a "perfect" recommender might generate recommendations which are poorly perceived by the user [2], if the recommendations are presented in an inappropriate way - or if the user fails to see why the recommendation should be good [3,4]. If a system can motivate why a recommendation is a good one, there is a higher chance of a higher perceived quality of the recommendation and the system in general [5].

Recommender systems do not only provide recommendations, they help users in decision making processes, they try to persuade users, filter out irrelevant data, etc. Thus, a recommendation should be more than just the presentation of an item. The context of the recommendation, i.e. why the item was recommended is just as important in order to satisfy the different needs of users [3]. For this purpose, this work presents a dataset collected through a user study where the task users where asked to perform was to find and rate movies, and in addition to this provide information on why a certain movie was liked or disliked.

Because explanations in the field of hybrid recommender systems are an important factor for the perception of recommended items, we provide a benchmark dataset from a user study for evaluating these explanations. In this user study users were asked to rate movies and provide reasons for and against watching them.

We analyze the dataset's characteristics, provide baseline explanation retrieval approaches and show how to evaluate its performances.

2 Related Work

Searching for "interesting" news, "exciting" videos, or items to purchase are common activities conducted everyday by millions of users on the web. Nevertheless, since its birth in the early 1990s, Internet has grown exponentially, even more with the rise of Social Networks and user-generated contents. The huge amount of data available on the web, far from being and advantage, is one of the main problems in the information searching process. This is because the ability of the users to discover new relevant information is seriously affected. In this context the first Recommender Systems (RS) appeared to automatically explore big data repositories and offer interesting contents to the users [6].

Since these first seed works, one aspect was pointed out by the RS community: how to convince the users to use recommended items. Even though, some RS could achieve a high theoretical performance, it will fail on their operation if the users don't interact with its recommendations. It is for this that some research lines in the RS field are focused on user confidence in recommendations, like for example Explanations in Recommendations. Through these explanations, RS are able to explain its decisions, hoping that in this way the users trust and use these recommendations [7].

2.1 Recommender Systems

The operation of a RS can be basically defined as the estimation of the interest that a given user could have in a given content. To carry out this estimation, RS have to take into account some considerations and challenges. Some of them are set out in [8]: (*1*) scalability, to be able to manage very large amounts of data; (*2*) pro-activity, to automatically offer recommendations without the users have to ask about [9], but without interfere with the user activity [10]; and (*3*) user privacy, since in spite of RS have to collect as much information as it is possible, users must be able to set which information want to share and which don't [11].

According to the technique followed to offer recommendations, in the state of the art three RS types have been traditionally proposed [12]:

- **Collaborative Filtering RS (CFRS)** [13,14]**.** These systems based their operation on the interactions conducted by the users (readings, consults, purchases). Through these interactions the system is able to bring together users with similar tastes; for instance, two users who have purchased the same products in the past. User interactions can be collected "explicitly", the system ask to the user for them (e.g. please, rate this movie), or "implicitly", the system automatically collect the information (e.g. the user clicks on some link about some movie). According to the way in which this interactions are managed, there are three types of CFRS:
  - **User based** [15]**:** Users who have interacted in the same way with the same item set are grouped together in the same *user neighbourhood*. It is expected that if in the past users shared the same tastes, in the future they will continue sharing tastes. Then, if an user like/purchase/interact with a new item, it will be recommended to the rest of the user in his neighbourhood. Some aspects that should be addressed in this kind of systems are: estimation of the neighbourhood size [16], how to compute the similarity between users or rating normalization [12].
  - **Item based** [17]**:** Item based CFRS use the user interactions but instead of grouping together users, they group together, in the same *item neighbourhood*, items that has been liked/purchased/consulted for the same users. Recommendation will be conducted by, given an user who likes a set of items, offering him new contents in the same item neighbourhood that the already liked items.
  - **Model based** [18]**:** Both previous methods (item and user based) has the same problem: its complexity is very high (quadratic) to be applied in a real scenario. If U is the number of total users, I is the number of total items and K is the neighbourhood size, the complexity of item based systems will be $O(I^2 * U * K)$, while the complexity of the user based will be $O(U^2 * I * K)$. Due to this problem, model based CFRS was proposed. Basically these systems process in somehow the information to generate offline data models that simplifies the recommendation process. Simplification techniques proposed are, among others [19,20]: clustering, Latent Semantic Analysis, Latent Dirichlet Allocation or Support Vector Machines. Through

these techniques complexity of the systems is reduced to $O(I * U * K * L)$, being L the number of iterations needed to generate the data models.

- **Content-Based RS (CBRS)** [21,22]**.** CBRS are based on the content of the items to be recommended. These systems try to improve the recommendation process by taking advantage of the information contained on the items, and not only of the interactions with them. Instead of the user USER0001 likes the item ID0001 (as CFRS do), in these systems the interaction will be: USER0001 likes the item ID0001, which is a *movie* with *title*, *content*, *genres*, *actors*, *director*, etc.

  CBRS have to cope with two aspects: how to model item contents and how to use these models to recommend new items. Regarding to the former, the work of Pazzani and Billsus in [23] discusses about different ways to represent and model item contents. Regarding to the later, also in [23] it is exposed that CBRS can be seen as Information Retrieval (IR) Systems, where the items to be recommended compose the IR Index and the user profiles are understood as the IR queries. Other method that has been proposed to carry out recommendations is the application of classification methods. It has been posed, for instance, in the work in [24] where items are classified in relevant/irrelevant by using as training the previous items liked by the users.
- **Hybrid RS (HRS)** [25,26]**.** CBRS and CFRS address the recommendation problem from different points of view. HRS try to exploit the benefits of both approaches by combining CBRS and CFRS in a single system.

  The most basic approach to hybridize systems is to separately execute both systems and then merge its results. This approach is followed by the systems presented in [27,28], or [29]. Although these first works achieve a performance increment of the basis systems, because of their simplicity they don't really explode the potential of hybridizing systems. In this sense other more elaborated approaches has been recently proposed. Jack and Duclaye in [30] present a system that use the data generated for a CFRS to enrich the item description (with information of similar items and users). The item content plus the information added by CFRS are used to generate recommendations by applying a CBRS. Berkovsky et al. present other hybridization technique [31]: if there is enough information about some item content, a CBRS is applied; however, if there is no enough information a CFRS is applied.

Independently of the type, there is some general considerations about RS, which are addressed in this work: Semantic Based Recommendation, Context-Aware Recommendation and Trust-Based Recommendation.

Semantic Recommendation. Semantic RS refer to all those systems that use some knowledge base, containing semantic information, for their operation. Rationale behind these systems is the exploitation of semantic information (present in such knowledge bases) to mitigate the lacks in the item contents representation [32]. Several works have been developed in this field [33–35]. These works are mostly based on the use of the information and the relationships in

knowledge databases (for example, FreeBase or DBPedia) to infer some knowledge with which enrich both user profiles and item descriptions.

Context-Aware Recommendation. Context-Aware Recommendation is related with the problem of *situated action*, posed by Mobasher [36]: the relevance of an item for an user is dependant on the user context. This problem has been addressed by means of different approaches, but one of the most common is to generate sub-profiles of the user profile according to the context information about the user. An example of this approach is the work of Said et al. [37], where it is proposed a movie recommender system in which different sub-profiles are generated according to *where* and *when* an user has watched a movie. These sub-profiles are subsequently used to personalized the recommendation step according to the current context of the user. A broad study about context-awareness can be consulted in [38], where the authors motivates the problem and recompiles some of the more novel approach in this field. The issues related with context-awareness are also exposed in the survey presented by Bettini et al. [39]. In this survey, they start presenting the most primitive techniques based on keywords or object-roles. They explain the utilization of spatial models and finally the application of ontology based models.

Trust-based Recommendation. Trust-based Recommendation (TBR) pursues to increment the user confidence in recommendations offered by the system, as it also intended in the work presented here. This kind of systems try to reproduce the natural process in which an user get recommendations in the real world. With this, it is intended to obtain more accurate and reliable recommendations. As it is posed before, user satisfaction is not only dependent on system performance; it is also related to psychological aspects, as trust. In this sense, an interesting user behaviour is explained by Shina and Swearingen in [40]. They show that users prefers recommendation made by their friends and acquaintances, even though these recommendations tend to be less novel and accurate. Another points of view in TBR are: the one proposed by Barman and Dabber in [15]. The authors base the recommendation in the "popularity" of the recommended items; or the social-based approaches that pretends to take advantage of the information in such platforms [41].

2.2 Explanations

Explanations of recommendations is not a new research topic, e.g. [42,43], the body of conducted work in this topic is by no means small.

Explanations in Recommender Systems. Tintarev and Masthoff present in [44] an extensive survey about this field. In this survey the authors present seven aspects that represent the facilities that explanation can offer to recommender systems operation and how the existing systems cope with these aspects.

One of the most important conclusion of their study is that the system has to be able to offer the explanations to the users "in their own terms". In this sense, the approach presented in this paper cover this aspect by explaining recommendations based on how the users have explained this preferences. The dataset especially developed for this work also offers the possibility to evaluate system explanations in the same way.

A similar approach to the one presented here is proposed by Symeonidis et al. in [45]. The authors propose the use of item attributes of the rated items to construct user profiles and to offer and explain recommendations. Unlike to the work presented here, the work in [45] doesn't offer the possibility that users individually rate the item attribute/s. It only allows the rating of the whole item, setting this rating for all of its attributes, doing that the system can infer wrong user preferences (i.e. I like The Godfather only because I like Marlon Brando, but I don't like Robert Duvall). The same limitation is shared by the explanation system presented in [3]. This system uses the tags with which an item (movies in MovieLens collection) is annotated to explain recommendations. But, it doesn't allow rating these tags one by one. The own authors expose this limitation in their conclusions.

Herlocker et al. present a model for *how* and *why* recommendation explanations should be implemented based on the conceptual model of the user's recommendation process and support this by empirical evidence [4]. They argue that explanations help detect, or estimate the likelihood of faulty recommendations (so-called recommendation errors). In a conceptually similar work, Bilgic and Mooney measure explanations in terms of *satisfaction* and *promotion* [46], where promotion refers to the explanation that most successfully convinces the user to pick an item and satisfaction refers to the to the explanation that best allows the user to assess the quality of an item.

In one of the earlier works on explanations in recommender systems-related work, Johnson and Johnson [42] attempt to identify what an explanation is and present both strengths and weaknesses in explanations in information systems. They list three limitations in explanation-related work: (*1*) the lack of a unifying theory of explanation, (*2*) the inability to identify and develop criteria for evaluating explanations, and (*3*) the lack of empirical studies in the field. It should be noted that the as the first and second limitations still apply to some extent, during the two decades since their work, a significant amount of empirical studies in the field of explanations has been undertaken [2,46].

Explanations in Persuasive Systems. Persuasive systems aim at heightening the user's experience of a system by trying to persuade the user to change her attitude or behavior [47]. Fogg lists several persuasion techniques and how they apply to different contexts and situations [48]. Physical persuasion, for instance, is the way a person acts while trying to persuade someone else. Mostly related to the type of persuasion explanations create is the concept of language-based persuasion, which according to Fogg's examples is the "interactive language use", either by a person or a system.

In their overview of the last few years of state-of-the-art in persuasive systems, Torning and Oinas-Kukkonen [49] note that the majority of the work focuses on behavioral changes created by persuasive systems, rather than persuading changes in users' attitudes. They also categorize recent papers into one of three types of persuasion context; *the intent*, *the event*, and *the strategy*.

Explanations in Decision Support. Decision support-related explanations are common in E-Commerce systems where the explanation is intended to help the customer select a specific product (and in the end be happy with the choice). Al-Qaed and Sutcliffe [50] tested user reactions to supporting tools and system advice and found that explanations and suggestions need to be personalized and contextualized in order to influence the user most.

Häubl and Trifts investigate how *interactive decision aids* (i.e. among other things explanations) affect consumer decision making in online shopping environments [51]. They find that there are strongly favorable effects on quality, efficiency, and satisfaction with purchase decisions that are supported by explanations and similar decision aids. Jedetski et al. make similar satisfaction-related observations in tasks performed on websites with decision support systems [52].

3 Attribute Explanations

The core of RS is the items to be recommended; these items can be: person, animal, book, movie, etc. They all have in common that they represent entities which can be described by attributes, e.g. writer of a book or director of a movie. In most cases there is a semantic description available where these attributes are predicate-object relations, e.g. *Coppola is director* of *The Godfather*. The idea is that these attributes serve as explanation for an item when being recommended, like "I think you'll like this movie because of *actor x* and *director y*, even though you don't like *actor z*."

Any attribute explanation for a recommended item should consist of a set of attributes of the recommended item which supports the recommendation and a set of attributes speaking against this item. The information about those attributes should be visualized or highlighted in the user interface, so that the user perceives the pros and cons of the recommended item immediately.

3.1 Items

Semantic knowledge bases like Freebase[1] and DBpedia[2] store structured information about entities. Hence these knowledge sources can be used to retrieve candidate attributes for recommender explanations. To find proper predicates (like for movies *is starring in* or *is director of*) that can serve for recommender

[1] http://www.freebase.com
[2] http://dbpedia.org

explanation we suggest to do a survey, where users have to provide this information about item of the same type.

In the sense of semantic representation an item i is a collection of triples, where all subject-predicate-object relations refer to the same subject identifier. All items of the same type share the same semantic predicates and often even the same predicate-object relations.

3.2 Attribute Explanation Model

Let i be a recommended item and A_i a set of attributes of this item. Each $a \in A$ is a predicate-object relation. An attribute explanation for item i consists of two sets of attributes A_i^+ and A_i^-, where A_i^+ includes those a which support the recommendation of i and A_i^- contains those a which speak against the recommendation. No attribute a is part of both A_i^+ and A_i^-, but there might be "neutral" a which are not part of A_i^+ or A_i^-.

3.3 Attribute Ratings as Attribute Explanation Retrieval Approach

As baseline attribute explanation approach we suggest to pass item ratings to the belonging attributes and average this value. The attribute rating r_a is computed with

$$r_a = \frac{\sum_{i \in I_a} \sum_{r \in R_i} r}{\sum_{i \in I_a} |R_i|} \tag{1}$$

where I_a is the set of items containing attribute a and R_i is a set of ratings for item i.

The retrieved attribute ratings can be seen as a measure of quality which helps to select those attributes which should be used to explain recommendations.

Let m be a model which generates attribute explanations for any item i. The model m needs to retrieve A_i to add selected attributes to the attribute sets A_i^+ and A_i^-. A_i^+ contains all $a \in A_i$ where r_a is known and $r_a > t_1$, where t_1 is a manually selected threshold for retrieving "good" attributes. A_i^- contains all $a \in A_i$ where r_a is known and $r_a < t_2$, where t_2 is the threshold for retrieving "bad" attributes. For instance on a rating scale with 5 options ($\{1, 2, 3, 4, 5\}$) the thresholds can take the values $t_1 = t_2 = 3$.

4 Dataset Retrieval

To create and evaluate attribute explanation retrieval approaches there is a need of collecting ratings for items and belonging attributes. In spite of all of the work conducted in this area, there are no available datasets (to our knowledge) providing the same kind of information as the one presented in this work. We collected user feedback for movies through an online user study, where users were asked to select a movie, rate it and then provide feedback about the role of the attributes which have influenced the rating.

4.1 User Study

The online user study was created to continuously collect data to improve the resulting dataset[3].

To participate, users have to login to the survey, but to respect privacy, users in the final dataset are anonymized.

On the main view, there are three ways of selecting a movie. First, there is the possibility to search for movies. Second, movies can be selected from the members of a Freebase list of rated movies. Third, movies can be selected from a list of the last selected movies of all users. When a movie is selected, the user is directed to the movie page.

The movie page provides additional information about the movie and the possibility to rate it (see Fig. 1). After rating, users are asked to provide the reasons speaking for and against the movie. The help text points out that even a bad movie might have reasons to watch and a good movie might have reasons against watching it. When the user (in Fig. 1) scrolls down, movie attributes from genres, country of origin, music by, story by, produced by, executive produced by, production companies, edited by, rated, adapted from and awards won can be selected as reasons for or against this movie.

For each of these attribute types (e.g. "Performances") the known attributes (e.g. actors) are presented and the user can decide if one specific director, actor, etc. has influenced the given rating. To do so, the user can click on a green or red flag to annotate reasons for or against watching this movie. If an attribute did not influence the rating the user is asked to provide no attribute rating.

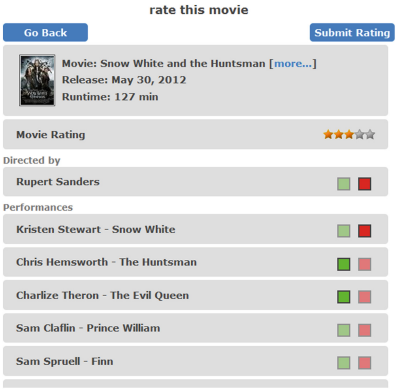


Fig. 1. Example movie page with given rating and provided reasons for and against watching it.

[3] http://www.dai-labor.de/~scheel/dataset/

4.2 Data Source and Data Structure

The online user study is completely based on Freebase data. Hence all identifiers refer to Freebase entities.

While the user study will go on, there will always be tagged versions which should be used and referenced, so that on the one hand research is done on current data, but on the other hand can be compared to others.

4.3 Data Analysis

We refer to the dataset as RER_{movie} (movie recommender explanation retrieval). The dataset was cleaned. By removing all movie ratings with no rating or reason given. Also all users which provided less than two ratings were removed.

At the time of writing, the dataset contains feedback from 53 users. These users provided 650 movie ratings (consisting of a rating and at least one reason) for 299 different movies. The mean number of movie ratings per user is 12.26 and the mean number of reasons per movie rating is 10.15. The inter-annotator agreement of movie ratings is the mean distance of the ratings of equal movies. It is equal to 1.01 for a maximum rating distance of 4 (the rating interval is between 1 and 5).

The inter-annotator agreement of the reasons can be calculated by looking at the attributes of each movie which have been chosen as reasons by more than one user. The agreement is then calculated by counting how often the reason type (pro or con) matches. Figure 2 shows the agreement per reason type, the overall agreement is 0.85. However, this value hides the fact, that the probability that there are equal reasons is only 0.54. In other words, there is a 54 % chance that a reason which was selected by one user is also a reason for another user, but if both select the same reason, they agree to 85 % that this reason is a reason for or against watching it. Figure 2 shows that the reasons with the most disagreement are rather uncommon movie features like age rating and editor.

In total users provided 6,597 reasons for movies. The most prominent movie attributes speaking for a movie are the genres (35 % of all movie supporting reasons) and the actors (31 %) followed by the director (6 %). The main reasons against movies are actors (26 % of all reasons for not watching movies) and genres (18 %). Hence, in the domain of movie recommendation genres and actors are the main sources for selecting attribute explanations.

The overall percentage of the reasons against movies is low. The reasons can be found in the majority of positively rated movies in the dataset (one star: 27, two stars: 48, three stars: 103, four stars: 217, five stars: 255). Figure 3(a) supports this fact by showing the mean number of reasons per item among the numerical ratings. For a bad movie (one star rating) there are in average 6.9 reasons against and still 3.5 reasons for a movie. For top rated movies (five star rating) there are in average 9.6 reasons for and 1.3 reasons against watching them. It can be seen that positively rated movies have a higher percentage of reason for a movie than against a movie, but in average always at least one reason against a movie can be found. For the worst rating, the number of reasons against

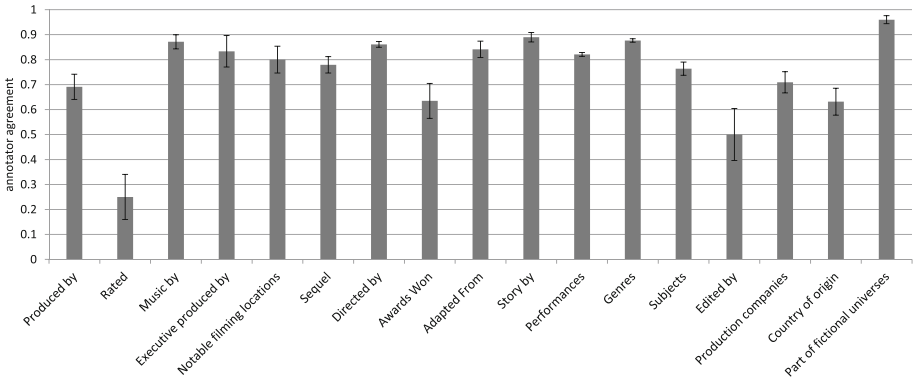
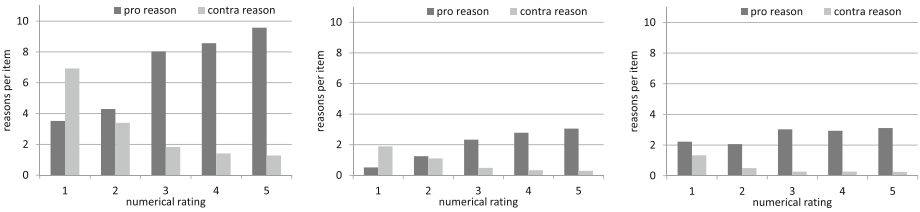


Fig. 2. Annotator agreement on reason types (pro or con) for equal attributes. Annotators agree if they selected the same attribute with the same reason type.



(a) All (b) Performances (c) Genres

Fig. 3. Mean number of reasons among the numeric ratings. In Fig. 3(a), each reason was taken, while in Fig. 3(b, c) the reasons are filtered to actors and genres.

movies is higher than the number of reasons for movies. This statement is still valid when only looking at the actors which were chosen as reason in Fig. 3(b), but invalid when looking at the genres in Fig. 3(c). Even if a movie is bad, the user might like the genre.

5 Attribute Explanation Retrieval Performance

Besides the possibility of studying explanation retrieval in general (see Sect. 4.3), the dataset can be used to evaluate approaches for attribute explanation retrieval. We assume that there is a model m returning attribute explanations as described in Sect. 3.2. We first describe how such a model can be evaluated and perform an evaluation on the approach described in Sect. 3.3.

5.1 Evaluation of the Explanation Retrieval

The evaluation on RER_{movie} dataset is done iteratively. For each movie i m provides a set of explanations speaking for the movie and one set for reasons

speaking against watching the movie. These sets will be compared with the user given set of reasons for i. Even when there are different sets of reasons available for i, they are evaluated independently. After evaluating all proposed explanation sets, the performance values are averaged to a final performance value.

To compare the user given feedback for each i with the attribute explanations coming from m, the precision and the recall has to be computed first.

$$precision = \frac{TP}{|proposed\ attributes\ from\ m|} \tag{2}$$

$$recall = \frac{TP}{|attributes\ from\ user\ feedback|} \tag{3}$$

TP is equal to proposed attributes which are part of the user feedback and share the same type of reason (pro or con), the precision on the RER_{movie} dataset is defined as the fraction of the attributes in the proposed explanation set that are correctly classified reasons. This means, that a proposed attribute only is counted, when it was selected by the user and when the user and the explanation agree in the fact that it speaks for or against the movie. In general the explanation retrieval approach should return all user given reasons. The recall is needed to punish approaches which only return some attributes to raise the precision value.

A suitable measure for evaluating the performance of explanation retrieval approaches is the F-measure, because it combines precision and recall. For this evaluation, F_1 measure, the harmonic mean of precision and recall, has been applied.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4}$$

5.2 Baseline Explanation Retrieval

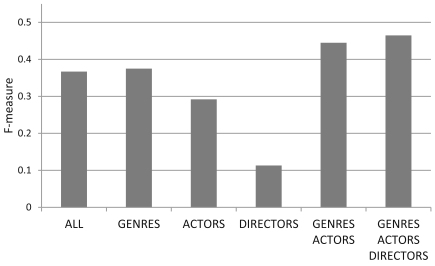
Given that there is no available dataset including attributes explanation, we couldn't compare our approach to already existing baselines. Then, we applied the baseline attribute explanation retrieval approach from Sect. 3.3 to our RER_{movie} dataset. Additionally, two primitive approaches will be evaluated, which do not judge if attributes speak for or against a movie, but return all attributes as reasons for or against it. These primitive approaches help to reflect the baseline's performance values.

Setting. Section 4.3 results in an observation that there are two types of reasons which have been chosen most: "performances" (actors) and "genres", followed by "directed by". To appreciate this observation, the evaluation is done also on subsets of movie attributes. Besides taking all attributes (ALL) also only genre (GENRES), actor (ACTORS) and director (DIRECTORS) attributes and combinations of these subsets are taken.

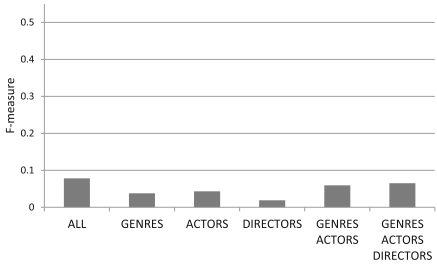
The evaluation was done by conducting a five-fold cross validation, where each fold contained a training set consisting of four subsets and a test set consisting of one subset. The baseline recommender candidate used the training set to apply the attribute rating approach, which passed the movie ratings to belonging attributes. If this average value was higher than 3 an attribute was declared as reason for a movie, otherwise as reason against a movie. Note that for instance ALL means all attributes with known attribute rating.

Evaluation. In Fig. 4 six sets of attributes are taken to evaluate selected approaches. While set ALL contains all attributes of a movie, the set GENRES only consists of genre attributes, etc.

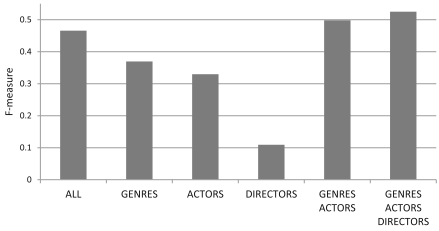
Figure 4(a) shows evaluation results for the approach returning all selected attributes as reason for the movies. In contrast to this approach, the approach in Fig. 4(b) tags all selected attributes as reason against a movie. Figure 4(c) shows evaluation results of the approach described in Sect. 3.3.



(a) Recommend attributes as reasons for the movies.



(b) Recommend attributes as reasons against the movies.



(c) Recommend attributes as reasons for or against the movies according to attribute ratings.

Fig. 4. Proposed baseline explanation retrieval approaches. The task was to generate explanations for each movie in the RER_{movie} dataset. In Fig. 4(a) the returned reasons have been declared as reason for a movie and in Fig. 4(b) against a movie. In Fig. 4(c) attribute ratings (which were passed from movie ratings of the training sets) determine if an attribute speaks for or against a movie.

While the recall of returning all attributes (in average 37.9) in Fig. 4(a) is very high, there is a high number of false positives, leading to a low precision and hence to a low F-measure. The best primitive attribute recommender would be to return all genres of a movie (in average 6.7), claiming they are reasons for watching a movie. This recommender can be improved by adding all actor attributes (in average 16.4) and further improved by adding all director attributes (in average 17.5). According to the F-measure in Fig. 4(b) there is no primitive attribute recommender for recommending attributes speaking against a movie, which is caused by the high percentage of reasons for movies in the dataset.

The best evaluated results can be found in Fig. 4(c) which should be taken as baseline when developing new approaches. When taking all attributes (in average 25.2), this approach reaches the performance of the best primitive approach which only returned all genre, actor and director attributes. When applied to this set of attributes (in average 12.7) the rated attribute approach in Fig. 4(c) performs reaches a F-measure value of 0.53.

The approach in Fig. 4(c) is designed to return reasons for and against movies. The mean percentage of reasons for the movie on the set ALL was 86.2 % and on the set containing all genre, actor and director attributes 89.8 %.

6 Conclusion and Future Work

This work presents a survey of explanations in recommender systems, particularly focused on Hybrid Recommender Systems. On top of the conclusions extracted from this review, we research on improving user's personal perception of recommended items. For a better perception, item attributes have to be presented in such way that the users may recommend or even speak against the item, based on them. These attributes will be presented along with other explanation types, if available.

This work includes a user study in the domain of movies, where users were asked to provide information about self-selected movies. This information includes a numerical rating and a set of attributes speaking for or against the rated movie. Unlike the rest of the works existing in the literature, which infer the attribute rating based on rating given for the items, the one presented here allows that the users directly rate the item attributes. The obtained dataset will allow recommending new items based on the rating given for their attributes. We refer to the resulting dataset as RER_{movie}. It can be downloaded at http://www.dai-labor.de/~scheel/dataset/download. This work includes a user study in the domain of movies, where users were asked to provide information about self-selected movies. This information includes a numerical rating and a set of attributes speaking for or against the rated movie. Unlike the rest of the works existing in the literature, which infer the attribute rating based on rating given for the items, the one presented here allows that the users directly rate the item attributes. The obtained dataset will allow recommending new items based on the rating given for their attributes. We refer to the resulting dataset as RER_{movie}. It can be downloaded at http://www.dai-labor.de/~scheel/dataset/download.

The collected data shows that genres and actors of movies are the most chosen features for reasons speaking for and against watching a movie. The ratio of reasons for and against movies depends on the numerical rating. For a bad movie there are in average 6.9 reasons against and still 3.5 reasons for a movie. For top rated movies there are in average 9.6 reasons for and 1.3 reasons against watching them.

The objective of collecting the data was to create a benchmark dataset for evaluating attribute explanation retrieval. For evaluation, the provided reasons for each movie in the dataset are compared with the proposed explanations coming from the evaluated approaches. This approach, which returns attribute explanations for and against movies, can also be applied to other items and its attributes.

The dataset can be used to create explanation retrieval models. We have shown how to evaluate such models and as an example evaluated an approach which passed ratings for movies to belonging attributes to create a ranked list of attributes which could be used to decide if attributes speak or against watching a movie.

For future work, we will apply learning to rank approaches on the data to receive a better list of ranked attributes or even enable personalized attribute explanation retrieval models. Another direction could be to receive quality values for attributes from external data like the Netflix data.

Although this dataset can be used already, we are still collecting data to be able to create and evaluate personalized explanation retrieval approaches.

References

1. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer, New York (2011)
2. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI '06 Extended Abstracts on Human factors in Computing Systems, CHI EA '06, pp. 1097–1101. ACM, New York (2006)
3. Vig, J., Sen, S., Riedl, J.: Tagsplanations: explaining recommendations using tags. In: Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI '09, pp. 47–56. ACM, New York (2009)
4. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, CSCW '00, pp. 241–250. ACM, New York (2000)
5. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**, 5–53 (2004)
6. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, pp. 175–186. ACM, New York (1994)
7. McSherry, D.: Explanation in recommender systems. Artif. Intell. Rev. **24**(2), 179–197 (2005)

8. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 1–35. Springer, New York (2011)
9. Sae-Ueng, S., Pinyapong, S., Ogino, A., Kato, T.: Personalized shopping assistance service at ubiquitous shop space. In: International Conference on Advanced Information Networking and Applications Workshops, pp. 838–843 (2008)
10. Puerta Melguizo, M.C., Boves, L., Deshpande, A., Ramos, O.M.: A proactive recommendation system for writing: helping without disrupting. In: Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore!, ECCE '07, pp. 89–95. ACM, New York (2007)
11. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 627–636. ACM, New York (2009)
12. Candillier, L., Jack, K., Fessant, F., Meyer, F.: State-of-the-art recommender systems. In: Chevalier, M., Julien, C., Soule-Dupuy, C. (eds.) Collaborative and Social Information Retrieval and Access-Techniques for Improved User Modeling, pp. 1–22. IGI Global, Hershey (2009)
13. Boim, R., Milo, T., Novgorodov, S.: Diversification and refinement in collaborative filtering recommender. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 739–744. ACM, New York (2011)
14. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 145–186. Springer, New York (2011)
15. Barman, K., Dabeer, O.: Local popularity based collaborative filters. In: 2010 IEEE International Symposium on Information Theory Proceedings (ISIT), pp. 1668–1672. IEEE (2010)
16. Bellogín, A., Cantador, I., Castells, P.: A study of heterogeneity in recommendations for a social music service. In: Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 1–8. ACM (2010)
17. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. (TOIS) **22**(1), 143–177 (2004)
18. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Morgan Kaufmann Publishers Inc. (1998)
19. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press, Cambridge (2010)
20. Kelleher, J., Bridge, D.: An accurate and scalable collaborative recommender. Artif. Intell. Rev. **21**(3–4), 193–213 (2004)
21. Castellanos, A., Cigarrán, J., García-Serrano, A.: Content-based Recommendation: Experimentation and Evaluation in a Case Study. Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2013) (2013)
22. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 73–105. Springer, New York (2011)
23. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)

24. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
25. Kim, H.-N., Ha, I., Lee, K.-S., Jo, G.-S., El-Saddik, A.: Collaborative user modeling for enhanced content filtering in recommender systems. Decis. Support Syst. **51**(4), 772–781 (2011)
26. Lucas, J.P., Luz, N., Moreno, M.N., Anacleto, R., Figueiredo, A.A., Martins, C.: A hybrid recommendation approach for a tourism system. Expert Syst. Appl. **40**(9), 3532–3550 (2012)
27. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997)
28. Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. Artif. Intell. Rev. **13**(5–6), 393–408 (1999)
29. Vozalis, M., Margaritis, K.G.: Enhancing collaborative filtering with demographic data: The case of item-based filtering. In: 4th International Conference on Intelligent Systems Design and Applications, pp. 361–366 (2004)
30. Jack, K., Duclayee, F.: Improving explicit preference entry by visualising data similarities. In: Intelligent User Interfaces, International Workshop on Recommendation and Collaboration (ReColl), Spain (2008)
31. Berkovsky, S., Kuflik, T., Ricci, F.: Cross-representation mediation of user models. User Model. User-Adapt. Inter. **19**(1–2), 35–63 (2009)
32. Peis, E., del Castillo, J.M., Delgado-López, J.: Semantic recommender systems. Analysis of the state of the topic. Hipertext.net **6**, 1–5 (2008)
33. Ghani, R., Fano, A.: Building recommender systems using a knowledge base of product semantics. In: Proceedings of the Workshop on Recommendation and Personalization in ECommerce at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web based Systems, pp. 27–29 (2002)
34. Cantador, I., Castells, P.: Multilayered semantic social network modeling by ontology-based user profiles clustering: Application to collaborative filtering. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 334–349. Springer, Heidelberg (2006)
35. Wang, R.-Q., Kong, F.-S.: Semantic-enhanced personalized recommender system. In: 2007 International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4069–4074. IEEE (2007)
36. Mobasher, B.: Contextual user modeling for recommendation. In: Keynote at the 2nd Workshop on Context-Aware Recommender Systems (2010)
37. Said, A., De Luca, E.W., Albayrak, S.: Inferring contextual user profiles-improving recommender performance. In: Proceedings of the 3rd RecSys Workshop on Context-Aware Recommender Systems (2011)
38. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 217–253. Springer, New York (2011)
39. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive Mob. Comput. **6**(2), 161–180 (2010)
40. Sinha, R.R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries'01, pp. -1–1 (2001)
41. Walter, F., Battiston, S., Schweitzer, F.: A model of a trust-based recommendation system on a social network. Auton. Agent. Multi-Agent Syst. **16**(1), 57–74 (2008)

42. Johnson, H., Johnson, P.: Explanation facilities and interactive systems. In: Proceedings of the 1st International Conference on Intelligent User Interfaces, IUI '93, pp. 159–166. ACM, New York (1993)
43. Johnson, H., Johnson, P.: Different explanatory dialogue styles and their effects on knowledge acquisition by novices. In: Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, 1992, vol. 3, pp. 47–57 (1992)
44. Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop, ICDEW '07, pp. 801–810. IEEE Computer Society, Washington, DC (2007)
45. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Moviexplain: a recommender system with explanations. In: Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09, pp. 317–320. ACM, New York (2009)
46. Bilgic, M., Mooney, R.J.: Explaining recommendations: Satisfaction vs. promotion. In: Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces, San Diego, CA, January 2005
47. Berkovsky, S., Freyne, J., Oinas-Kukkonen, H.: Influencing individually: Fusing personalization and persuasion. ACM Trans. Interact. Intell. Syst. **2**, 9:1–9:8 (2012)
48. Fogg, B.J.: Persuasive technology: using computers to change what we think and do. Ubiquity **2002** (2002)
49. Torning, K., Oinas-Kukkonen, H.: Persuasive system design: state of the art and future directions. In: Proceedings of the 4th International Conference on Persuasive Technology, Persuasive '09, pp. 30:1–30:8. ACM, New York (2009)
50. Al-Qaed, F., Sutcliffe, A.: Adaptive decision support system (adss) for b2c e-commerce. In: Proceedings of the 8th International Conference on Electronic Commerce: The New e-commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet, ICEC '06, pp. 492–503. ACM, New York (2006)
51. Häubl, G., Trifts, V.: Consumer decision making in online shopping environments: The effects of interactive decision aids. Mark. Sci. **19**, 4–21 (2000)
52. Jedetski, J., Adelman, L., Yeo, C.: How web site decision technology affects consumers. IEEE Internet Comput. **6**, 72–79 (2002)