Towards High Quality Recommendations: A Goal-Oriented and Ontology-Based Interactive Approach

Ronaldo Gonçalves Junior, Robert Ahn, Tom Hill, Lawrence Chung
Department of Computer Science
The University of Texas at Dallas
Richardson, TX, USA
{ronaldo.goncalves, robert.sungsoo.ahn, chung}@utdallas.edu,tom.hill.fellow@gmail.com

***Abstract*—Recommender systems are aimed to offer good recommendations when the search space is too big or even uncertain. For example, when deciding on a movie to watch or a restaurant to go. However, when users are not satisfied with such results, they might spend a considerable amount of time interacting with the system while being unsatisfied. Most commercial recommender systems seem to lack methods to understand user needs while guiding the user with an intelligent human-like interactive process towards a high-quality recommendation. In this paper, we present a goal-oriented and ontology-based interactive framework for high quality recommendations. Through a goal-oriented approach, the recommender system address and define user needs by using quality attributes, the so-called Non-Functional Requirements (NFRs), such as responsiveness, efficiency, accuracy of the recommendation, and so on. A list of potentially competing recommendations, pertaining to a user interest, are evaluated using machine learning (ML) algorithms, and, by following an interactive approach, system and user exchange information derived from the ontology in order to avoid problems related to uncertain, vague, or even incorrect, user inputs. Results from the assessment of a real online database (TMDb) combined with an experimental study shows that the proposed approach is able to guide the user towards high quality recommendations.**

Keywords- Recommender Systems, Interactive Systems, Non-Functional Requirements, Ontology, Goal-Orientated, Machine Learning

I. INTRODUCTION

Recommender systems are aimed to offer good recommendations to users (e.g., a good restaurant to go) and can be especially useful in scenarios where it is difficult for the user to evaluate options individually [1]. For instance, more than 500 hours of video are uploaded to YouTube every minute[1]. In the absence of a systematic approach, a vast amount of content that is potentially interesting to the user is likely to remain unexplored. Many commercial systems, including streaming platforms such as Amazon[2] and Netflix[3], have recommender systems in place to help users navigate through a seemingly boundless set of contents. Even though recommender systems are more common and popular, many challenges still require attention [2]. Some techniques have been proposed to better understand users' needs by making use of the communication between a user and the system [3][4], but most recommender systems seem to lack methods to understand user needs while guiding the user with an intelligent, human-like, interactive process towards high quality recommendations.

In this paper, we present a goal-oriented and ontology-based interactive framework for high quality recommendations. System and user interact in order to increase the prospect of satisfactory recommendations. Through a goal-oriented approach, we consider important quality attributes, or the so-called Non-Functional Requirements (NFRs), such as responsiveness, efficiency, accuracy of the recommendation, and so on. In this approach, a list of potentially competing recommendations pertaining to a user interest are evaluated by using machine learning (ML) algorithms, with the necessary features coming from the ontology. Results from the assessment of a real online database (TMDb) combined with an experimental study shows that the proposed approach is able to guide the user towards high quality recommendations.

II. BACKGROUND

Recommendation algorithms are typically divided into three different approaches [5][6]: Collaborative Filtering (CF), Content-Based (CB), and Hybrid systems. CF is one of the most common types of recommender systems [7], where the similarity between users is utilized to build recommendations [8]. Traditional CB recommender systems utilizes the similarity between items to build recommendations and are less wide-spread [9], but have been proven to contribute to the research field [10]. Finally, Hybrid systems can provide other benefits that the previous types do not offer [11][12]. For instance, Hybrid recommender systems have been used to incorporate justification in recommendations to achieve customer acceptance and trust [13].

Some of the related work closest to our approach perform item selection as a step in a conversational process and the system inquiries about item attributions while waiting for user response [14]. Other interactive approaches have similar concepts for recommendations with different contributions [4][15], such as knowledge based systems that use facts and rules to improve recommendations [16][17]. However, to the best of our knowledge, there is no consideration of user needs in

DOI reference number: 10.18293/SEKE2020-140
[1] https://www.youtube.com/about/press/
[2] https://www.amazon.com/gp/video/getstarted
[3] https://www.netflix.com/

a Goal-Oriented approach, where satisfactory recommendations are modeled using NFRs and further analyzed using measurable observations. Another important aspect is that there is limited consideration of domain-dependent ontologies within the interactive steps of the recommendation process.

III. THE PROPOSED APPROACH

The process of the proposed approach consists of four steps: 1) *Model NFR Softgoals*, 2) *Build recommendations*, 3) *Explore ontology-based concepts*, and 4) *Interact with the user*. The following sections describe each of these steps in further detail.

A. Step 1: Model NFR Softgoals

The proposed approach uses quality attributes, or the so-called Non-Functional Requirements (NFRs), to define high quality recommendations. In the context of recommendations, high quality can be further decomposed into different NFRs in order to better identify user needs. More specifically, a high quality recommendation may be AND-decomposed into three different NFR Softgoals: an efficient recommendation, a responsive recommendation and an accurate recommendation. Each of these Softgoals can be further decomposed and connected to other NFRs. For instance, an efficient recommendation may help a responsive recommendation, but an accurate recommendation has some negative contribution to responsiveness. An example of a complete model for high quality recommendations is shown in Fig. 1. It is worth mentioning that it is possible to make refinements to the graph to include other quality attributes such as security, user-friendliness, cost, and so on.

Note that it is important to define how these requirements will be satisficed. Operationalization Softgoals are related to external entities, such as ML algorithm results and user interaction. Each Softgoal may be marked by the labels satisficed, weakly satisfied or denied based on these external observations. These labels will, subsequently, be propagated to other goals bottom-up [18] until high quality recommendation can be validated or invalidated.

B. Step 2: Build recommendations

Using a Machine Learning-Based approach, it is possible to create prediction models that will help identify which items the user might be interested in. Consequently, instead of showing a list with all available items, we can show a reduced list with items that exclusively pertain the interest of the user. The first step in building recommendations is the acquisition of user data. This may be done by using private databases, publicly available datasets, etc. If user data is not available, it is possible to perform a customer survey, purchase data from providers, and so on. After acquiring user data, it is important to verify that undesired information is removed, such as null values, unavailable entries, and so on. Last but not least, the right format must be ensured. The main input feature for the CF recommender system algorithm utilized in this paper needs to be in the form of {<*userID, itemID, rating*>}. This implementation of the CF algorithm includes the computation of the distance between different users, which can be done by using a Pearson Correlation score, Cosine function, among others [5]. In other words, these scores measure how similar two users are. The scores for all users are stored in order to avoid repeated computation. The execution of the recommendation algorithm may take long periods of time, especially when dealing with considerably large datasets. However, by storing these values there is no need to re-execute the algorithm for the same data.

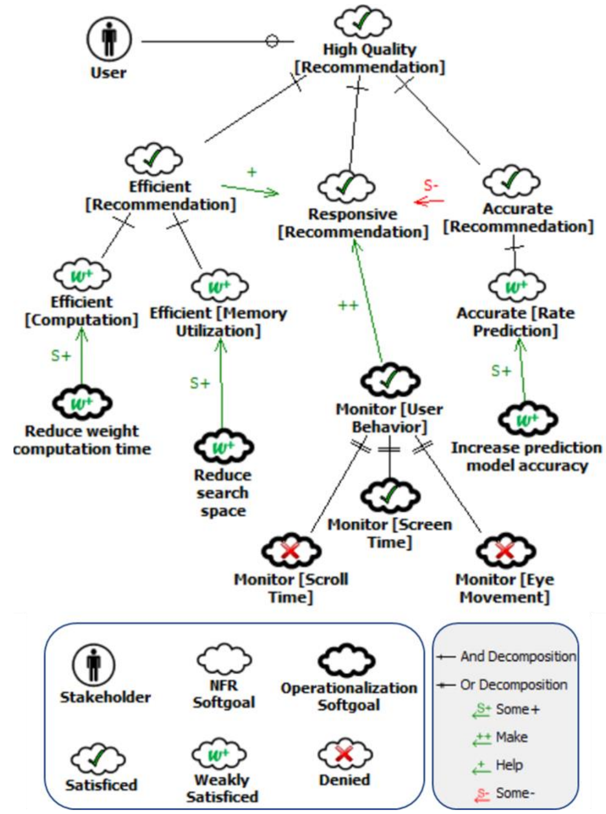

Fig. 1. A model for high quality recommendations.

C. Step 3: Explore ontology-based concepts

It is important to explicitly represented concepts such as User, Softgoals, and Recommendation Model in a domain-independent approach to avoid omissions while mapping Goal-Orientation (Step 1) and machine learning (Step 2). In addition, some concepts may be derived from a domain-dependent ontology. A complete set of concepts and their relationships can be found in Fig. 2. This ontology is domain-independent and can be used for various domains. In this step, we want to identify domain level concepts to guide the user for better recommendations. The movie domain is the most common domain for research in recommender systems [2]. For this reason, Fig. 3 shows portion of a domain-dependent diagram for the movie domain. The proposed approach uses each ontology concept as options to interact with the user. Each ontology concept is directly or indirectly connected to *Movie*, e.g., *Genre* and *Entertainment* respectively. Let *CN(a,b)* be the proposition that *a* is a conceptual neighbor of *b*, such that:

$$CN(a, b) \leftrightarrow \exists c[(a \in c) \wedge (b \in c) \wedge (c \neq M)] \qquad (1)$$

The propositional statement (1) states that a is a conceptual neighbor of b *iff* there exists an arbitrary element c in the movie domain, such that both a and b are members of c, and c is not

Movie (*M*). For example, both Information and Entertainment and members of Genre, and Genre is not Movie (M). Thus, *CN(Information, Entertainment)* evaluates to *True*. The next section shows the usage of this definition during the interaction with the user.

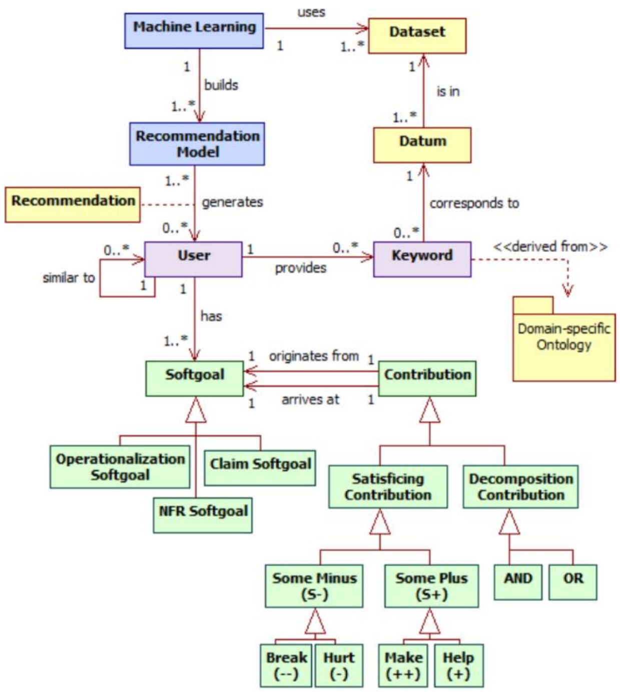

Fig. 2. Domain-independent ontology diagram for the proposed approach.

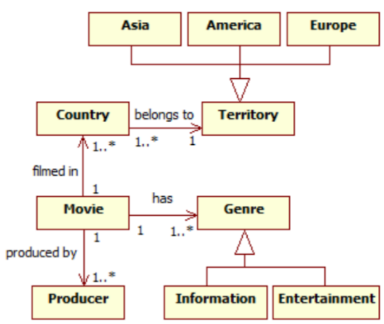

Fig. 3. An example of ontology concepts for the movie domain.

D. Step 4: Interact with the user

By interacting with a user, a recommender system may be able to better understand the user needs. This step of the proposed approach generates options derived from the ontology and present them to the user towards more satisfactory recommendations. To illustrate this step for the movie domain, let us consider that a user is looking for movie recommendations which are similar to the movie *The Mission (1986)*. First, it is necessary to monitor the user behavior to identify when help is needed. Consider that the user defines a time threshold of 60 seconds for screen time, i.e., the time spent on a screen without performing any actions, such as clicking, scrolling, etc. When this threshold is reached, the system will attempt to gather additional information from the user and provides options for selection. For the movie domain, we present the following set of options: {*Actor, Award, Country, Genre, ..., Producer*}. Note that these are ontology-based options for the movie domain. Different options would be available for a different domain. To illustrate this step, let us assume that the user does not remember the title of the movie, but selects *Brazil* as *Country*. Note, however, that *The Mission (1986)* takes place in Argentina.

The system will filter recommendations from other countries, except Brazil, and goes back to monitoring. The user may provide information multiple times. However, if the user is not able to provide additional information and help is still needed, the system will suggest alternatives. At this point, the system will evaluate each given option and modify the search space to comprehend conceptual neighbors of each option, one at a time. By using the conceptual relationships computed in Step 3, the system adjusts the user input from *Brazil* to the following set:{*Argentina, Bolivia, Brazil, Chile, ..., Venezuela*}, which is the set of conceptual neighbors of *Brazil* with respect to *South America*. Now, instead of movies that took place in Brazil, we also include movie recommendations that took place in many other countries, including the correct country, Argentina. This step can be executed multiple times, and through each iteration, we go back to Step 1 of the proposed approach to rebuild the NFR model and check if we achieved the goal of high-quality recommendations.

IV. An Experiment for the movie domain

In this experiment, we monitor the user screen time and, by following the Step 1 of the proposed approach, we mark the corresponding Softgoal in Fig. 1 as satisficed and mark the others as denied. By following the label propagation process [18], since at least one of the sub goals for *Monitor user behavior* is satisficed, *Responsive recommendation* is marked as satisficed. It is important to mention that, at this point, we cannot verify if we achieved *Efficient recommendation* or *Accurate Recommendation*. In this case, we leave these goals unlabeled and proceed to the next step of the proposed approach.

The data used in Step 2 for this experiment is from a popular dataset (MovieLens [19]) that was collected from The Movie Database [4] (TMDb). This dataset contains more than 45 thousands of movies, 26 million ratings and 270,000 users. Note that, due to space limitations, we show here the results for the first iteration of the proposed approach by using the CF algorithm, implemented in Java. With the similarity between users stored, it is now possible to recommend movies to users. By following the Step 3 of the proposed approach, all conceptual neighbors for the movie domain are built and stored as interaction options. Finally, by following the Step 4 of the proposed approach, it is possible to see a significant reduction in the search space. For instance, if a user selects *Italian* as *Language*, the search space is reduced from 45,466 movies to 1,529. Multiple aspects are worth mentioning here. First, different users will have different results based on the interaction. Second, the recommender system does not recommend the entire dataset, only movies predicted to be satisfactory. However, the original input for the ML algorithm will be reduced, which narrows the recommendations

[4] https://www.themoviedb.org/

significantly and lowers the amount of training examples being used, decreasing the weight computation time.

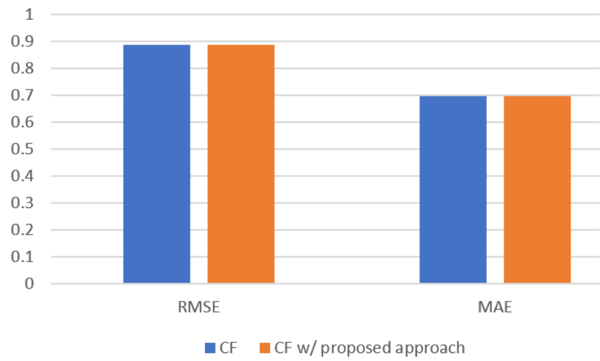

Fig. 4. CF algorithm results by accuracy (RMSE and MAE).

Once step 4 is completed, we go back to Step 1 to check if we achieved high quality recommendations in the NFR model. Since the search space and weight computation time were both reduced by a certain degree, we mark the corresponding Softgoals as weakly satisfied. Finally, for an arbitrary selection of users that rated movies, the first step of the proposed approach increased the accuracy of the results. Note that both Root Mean Squared Error (RMSE) and Mean Average Error (MAE) metrics were reduced by a considerably small margin, as shown in Fig. 4. Hence, *Increase prediction model accuracy* is marked as weakly satisficed. A complete iteration over the process of the proposed approach will result in a model fully labeled as shown in Fig 1. For this experiment, note that *High Quality Recommendation* is marked as satisficed after the label propagation process [18]. In other words, we can proceed to Step 2 once again to build more satisfactory recommendations. Note that this process can be repeated indefinitely, by check if *High Quality Recommendation* is satisficed or denied.

V. Discussion and Future Work

This work presented an approach that considers Non-Functional Requirements (NFRs) as key drivers for helping the user find good recommendations through an interactive system. Results from the assessment of an online database (TMDb) combined with an experimental study show that users are guided towards high quality recommendations by using the proposed approach. The work presented in this paper, to the best of our knowledge, is one of the first to propose the usage of a Goal-Oriented approach where satisfactory recommendations are modeled using NFRs and further analyzed by measurable observations. The outcomes of the proposed approach include: 1) a typical set of high quality goals and relationships between them, as well as ways to achieve them; 2) a domain-independent ontology for recommendations; and 3) an approach for the interaction between user and system to guiding the user towards more satisfactory recommendations.

Regarding our future work, we plan to to investigate ways to improve accuracy in a more impactful manner, as well as experimenting for different domains other than movies. Moreover, we intend to implement a tool to support the approach during all steps of the process for recommendations.

References

[1] P. Resnick and H. R. Varian. Recommender systems. Commun. ACM, 40(3):56– 58, March 1997.

[2] D. Jannach, M. Zanker, M. Ge, and M. Gröning. Recommender systems in computer science and information systems – a landscape of research. In Christian Huemer and Pasquale Lops, editors, E-Commerce and Web Technologies, pages 76–87, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[3] T. Mahmood and F. Ricci. Improving recommender systems with adaptive conversational strategies. In Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, HT '09, pages 73–82, New York, NY, USA, 2009. ACM.

[4] L. McGinty and B. Smyth. On the role of diversity in conversational recommender systems. In Proceedings of the 5th International Conference on Case-based Reasoning: Research and Development, ICCBR'03, pages 276–290, Berlin, Heidelberg, 2003. Springer-Verlag.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[6] C. He, D. Parra, and K. Verbert. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. Expert Systems with Applications, 56:9 – 27, 2016.

[7] M. Zanker, M. Jessenitschnig, D. Jannach, and S. Gordea. Comparing recommendation strategies in a commercial context. IEEE Intelligent Systems, 22(3):69–73, May 2007.

[8] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. R. Grouplens: An open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM.

[9] S. B. Ticha, A. Roussanaly, A. Boyer, and K. Bsaïes. User semantic preferences for collaborative recommendations. In EC-Web, 2012.

[10] M. Gemmis, P. Lops, G. Semeraro, and P. Basile. Integrating tags in a semantic content-based recommender. In Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08, pages 163–170, New York, NY, USA, 2008. ACM.

[11] R. Burke. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4):331–370, Nov 2002.

[12] I. Cantador, A. Bellogín, and P. Castells. A multilayer ontology-based hybrid recommendation model. AI Commun., 21(2-3):203–210, April 2008.

[13] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Providing justifications in recommender systems. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 38(6):1262–1272, Nov 2008.

[14] C. A. Thompson, M. H. Göker, and P. Langley. A personalized system for conversational recommendations. J. Artif. Int. Res., 21(1):393–428, March 2004.

[15] L. Ardissono, G. Petrone, and M. Segnan. A conversational approach to the interaction with web services. Computational Intelligence, 20(4):693–709, 2004.

[16] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. An integrated environment for the development of knowledge-based recommender applications. Int. J. Electron. Commerce, 11(2):11–34, December 2006.

[17] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In Recommender systems handbook, pages 1–35. Springer, 2011.

[18] L. Chung, B. A Nixon, E. Yu, and J. Mylopoulos. Non-functional requirements in software engineering, volume 5. Springer Science & Business Media, New York, NY, USA, 2012.

[19] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst., 5(4):19:1–19:19, December 2015.