Modeling a Dialogue Strategy for Personalized Movie Recommendations

Pontus Wärnestål
Department of Computer Science
Linköping University
ponjo@ida.liu.se

ABSTRACT

This paper addresses conversational interaction in user-adaptive recommender systems. By collecting and analyzing a movie recommendation dialogue corpus, two initiative types that need to be accommodated in a conversational recommender dialogue system are identified. The initiative types are modeled in a dialogue strategy suitable for implementation. The approach is exemplified by the MADFILM movie recommender dialogue system.

INTRODUCTION

A crucial issue for recommender system performance is the way the system acquires preferences from the user in order to generate a correct and sufficient preference model [3]. In general, this issue can be addressed by two approaches: The first is to improve or combine recommendation techniques and algorithms in order to predict items to a specific user. This approach is focused on the internal workings of the recommender system, and has resulted in significant advances [3]. The second approach is focused on exploring interaction models and techniques for conveying preferences to the system, and is thus more interaction-oriented. Of the two, the former approach has received the most attention, while the latter has been neglected to a large extent [5].

Advances in natural language processing have provided techniques for building speech interfaces and *spoken dialogue systems* [11]. The most common type of such systems is information-providing dialogue systems, and this research body has provided a number of frameworks and methods for building such systems. Natural language interaction and dialogue may have a positive impact on *recommender system* interaction and performance, since it allows a wide range of users to express their preferences in new ways, e.g. attribute tweaking and the candidate/critique model [4, 10]; and in various environments, e.g. using speech for personalized recommendations while driving [13].

Consider the example dialogue between a user and a spoken dialogue recommender system in Figure 1. This dialogue

U1: Hi! I would like a movie recommendation please
S1: Okay. What kind of movie do you feel like?
U2: I feel like watching a thriller
S2: Can you tell me one of your favorite thriller movies?
U3: Uhm, I'm not sure
S3: Okay. Can you tell me one of your favorite actors or actresses?
U4: I think *Bruce Willis* is good
S4: I see. Have you seen *Pulp Fiction*?
U5: No
S5: *Pulp Fiction* is a thriller starring *Bruce Willis*

Figure 1: Example recommendation dialogue about movies between the MADFILM system (S) and a user (U). Movie titles and actor names are in *italics*.

gives an idea of how recommendation dialogues in MADFILM are carried out. The system guides the user by asking for relevant preferences, but remains flexible depending on the user's responses (as in utterances U3 and S3 in Figure 1). Recommendations are presented and explained in a conversational manner (S5), based on the gathered user preferences (U2 and U4).

This paper aims to make a contribution on natural language dialogue in user-adaptive recommender systems. Specifically, two issues are addressed: (1) how recommendation dialogues between humans are manifested, and (2) how recommendation dialogue strategies can be modeled in a spoken recommender dialogue system. The contribution put forward is a dialogue control strategy suitable for implementation. The approach has been implemented in the MADFILM system, which is described in detail in [8].

HUMAN-HUMAN RECOMMENDATION DIALOGUES

In order to get empirical data covering aspects of recommendation dialogues, a corpus of human-human dialogues was collected. This section describes the study and its implications on dialogue strategy design.

Participants

Forty-eight participants (24 female and 24 male) were recruited. The participants did not know each other's movie

Workshop: Beyond Personalization 2005
IUI'05, January 9, 2005, San Diego, California, USA
http://www.cs.umn.edu/Research/GroupLens/beyond2005

preferences. Each dialogue session required two subjects, one acting in the role of a movie *recommender*, and the other in the role of a *client* wanting movie recommendations. In order to avoid repetition of recommendation strategies in the dialogues, each session had a new recommender. All sessions were kept as varied as possible in terms of gender and roles, including male-female, male-male and female-female dialogues, in both recommender and client roles in order to vary the dialogues as much as possible.

Apparatus

The study was set in a home environment designed for usability studies. Apparatus for the study included a laptop connected to an extensive movie information database [1] for the recommender to use, and a movie recommendation protocol for keeping track of movie recommendations. The dialogue sessions were recorded on Mini Disc.

Procedure

The recommender got a 15-minute tutorial on how to use the movie information database prior to the session. She also received a scenario, which provided her with the task of recommending in total ten movies. Five of these should have been seen by the client and verified as "good" recommendations (in order to indicate that recommendations were fit for the current client), and five should be previously unseen. When the recommendation protocol was completed, the session was terminated. The client was also presented with a scenario, which explained the client role. A total of 24 dialogues were recorded with two participants in each session, resulting in 7.5 hours of recorded material. Transcription of the dialogues resulted in 2684 utterances with a mean of 112 utterances per dialogue.

Results

When analyzing the material a number of dialogue phenomena were identified and quantified. The results include a categorization of critical issues to consider when modeling recommendation dialogues. In particular, two initiative types were identified and is the topic of this section. The material has been analyzed from other aspects as well, e.g. multimodal interaction, object manipulation, and global focus shifts. These aspects of the study are covered in [7].

Information Requests

The first of the two initiative types—*information requests*—is concerned with information about movies found in the database. Information requests are client-driven and occur when clients ask about properties (e.g. director or actor information) for a specific movie. Requests may be posed as a stand-alone initiative introduced by the client, or as a subdialogue within a recommendation initiative. Figure 2 shows an example of this.

[1] The Internet Movie Database (http://us.imdb.com).

R1: have you seen *The Bone Collector*?
C1: who is acting in it?
R2: these guys [displays a list of actors]
C2: yeah / I liked that one

Figure 2: Dialog excerpt where the client issues an information request about a movie (C1) in order to respond to the recommender's question (R1).

Preference Requests and Recommendations

The second initiative type—*preference requests*—is concerned with information about clients' movie preferences. They are recommender-driven, where the client's responses aid the recommender to assess a client "preference profile" in order to make qualified movie suggestions.

At an early stage in the dialogues a "recommendation base" is established. This is typically one of the first explicit attributes that the client puts forward, and on which a series of recommendations are based. It is common to return to the recommendation base throughout the dialogue, such as when wrapping up a sub-series of recommendations. On top of the original recommendation base several modifying attributes can be put for a few turns. The recommendation base can thus be modified; but it can also be changed completely. Changing the recommendation base requires some sort of explicit utterance from either of the dialogue partners. Four principal types of recommendation base changes or verifications are found in the corpus:

1. Changing the recommendation base when the client is done with the current one.
2. Changing or relaxing the recommendation base when there are no more options (i.e. all movies matching the current recommendation base have been considered).
3. Providing a new recommendation base suggestion when the client is uncertain or out of ideas.
4. Ensuring or verifying the attributes and importance of attributes in the current recommendation base depending on client feedback on recommendations (i.e. if the agreed recommendation base seems to generate several "bad" recommendations in the client's eyes).

An important implication for a recommender dialogue system design is thus to (a) allow users to change the recommendation base explicitly, as well as (b) let the system suggest recommendation base when the user does not suggest one herself, or when the current recommendation base has been exhausted.

Integration of Information and Preference Requests

As exemplified in Figure 2, information requests are often posed by clients in order to respond to recommenders' preference requests. Another important coupling between the two request types is that information request responses can

drive the recommendation dialogue forward, since the presented information triggers the user to provide new preference data or issue new information queries in the dialogue. Figure 3 shows an example of this. R1 is a preference re-

R1: I see / please name another good movie
C1: uhm / who's starring in *Ransom*
R2: here are all the actors in *Ransom* [shows the actor list of *Ransom*]
C2: so what other movies has *Mel Gibson* done?
R3: all of these [points at Gibson's filmography list]
C3: right / oh yeah / *Braveheart* is one of my absolute favorites
R4: oh then I think you'd like *Gladiator*

Figure 3: Dialogue excerpt showing how client-initiated information requests move the dialogue forward.

quest issued by the recommender. The sub-dialogue initiated by the client in C1 results in another list of movies from which the client can pick favorites. C3 can thus be viewed as a response to the overall goal introduced in R1. However, without the possibility to interleave information requests, the client would not have arrived at the list in R3, and the preference given in C3. The dialogue continuing from R4 is a direct result of the content of C3, which in turn is a result of the information requests in C1 and C2[2]. Handling information and preference requests and the integration of the two in this manner is thus an important issue for modeling the dialogue in a system.

HUMAN-MACHINE RECOMMENDATION DIALOGUES

In order to operationalize the dialogue in an implementation, we need to analyze the collected corpus in greater detail. The basis for this analysis is through the process of *dialogue distilling*, which is a method for analyzing dialogue corpora with a particular aim for dialogue system development [9]. When distilling, we aim at systematically re-writing the original human-human dialogue corpus into a plausible human-machine dialogue by applying a set of guidelines. In the movie recommendation corpus, we appoint the recommender participant to function as "the system", and the client as "the user". In the following, this is how they are referred to.

Recommendation Dialogue Control

With the completion of the distillation, we have an empirical ground to start developing the language resources (e.g. lexicons and grammars) necessary for a system implementation. We also consider the two initiative types from the previous section for the dialogue management components. The simplest form of dialogue management is finite-state machines, which guide the user through a series of pre-defined steps. This form of dialogue is completely system-driven in terms of initiative [1]. One way to visualize a distilled dialogue is to draw a finite-state network consisting of system utterances represented as nodes, and user utterances represented as arches. Each separate network has a unique linear sequence, since there is exactly one user utterance following a given system utterance in each of the dialogues.

As outlined above, recommendation dialogues in the corpus consists of a combination of (a) system-driven preference requests, and (b) user-driven information requests. The mixed-initiative character of the dialogue can roughly be said to correspond to a seamless integration of these initiative types. Based on this assumption, we turn to the network graphs of the distilled corpus and try to merge the system-driven initiative into one general recommendation dialogue network graph modeled as a finite-state machine covering the distilled corpus. This is done by comparing system utterances with focus on *preference requests*, and then incrementally adding arches corresponding to the various user utterances that occur as responses to the nodes. The resulting network is—not surprisingly—larger and more complex than the individual dialogue networks. The view of the complete dialogue graph in Figure 4 corresponds to system-driven preference requests.

Dialogue Node Functionality

This section describes the nodes in Figure 4 and their function in the dialogue.

Initiating the Recommendation Dialogue

The START node simply generates a welcome message. A user response indicating that she wants a movie recommendation leads to the RECBASE node. In Figure 4, there is only one arrow arching out from the START node, leading to the RECBASE node. This is a simplification, since users have the possibility to directly set the recommendation base from this node, thus skipping the RECBASE node. In RECBASE, we establish a "recommendation base", which is the principal attribute set that future recommendations will be based on. There are several possible responses to the RECBASE depending on what attribute the user prefers. Most users want to base their recommendations on genre (e.g. a drama, comedy, or action movie), whereas some users aim for movies starring their favorite actor (e.g. "I would like a movie starring *Cary Grant* please").

Getting Attribute Values

GETVALGENRE is responsible for trying to assess what genre(s) the user is interested in. The GETVALACTOR node functions in a similar way, asking the user for names of their favorite actors or actresses. The information retrieved by these two GETVAL nodes is integrated in the recommendation base.

Acquiring Title Ratings

A central issue when utilizing recommender engines is to acquire title ratings from the user [12]. The more titles that are included in the user preference model, the better recom-

[2] *Mel Gibson*, who is mentioned in C2, occurs on the *Ransom* actor list in R2. *Braveheart* occurs in the filmography list of *Mel Gibson* in R3.

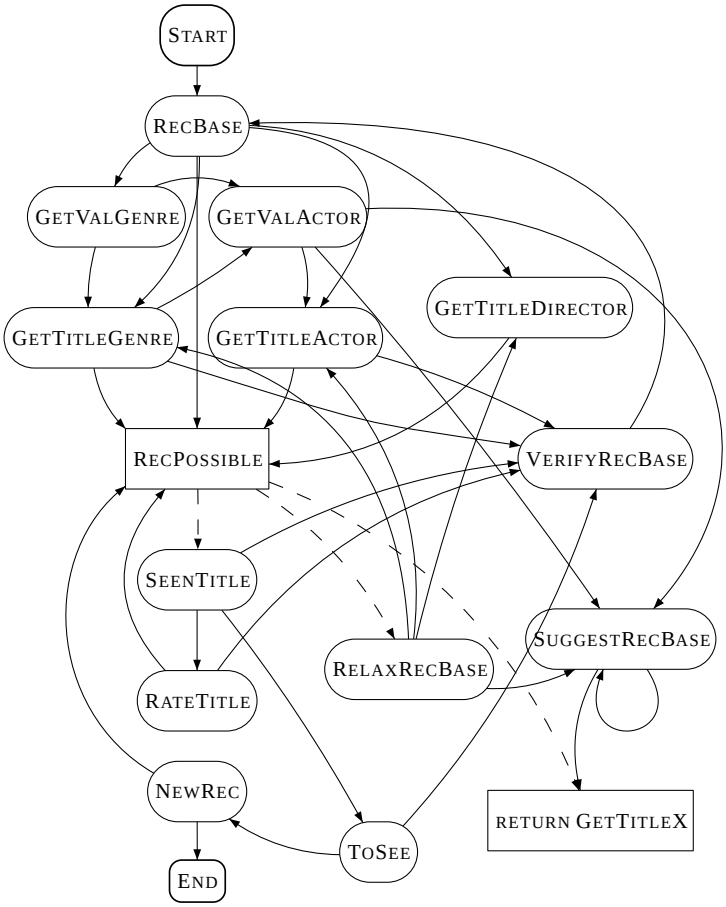

Figure 4: A recommendation dialogue network graph covering dialogue flows of the 24 distilled dialogues from the movie recommendation dialogue corpus. Some arches have been omitted for clarity. Rounded nodes correspond to a system utterance, whereas square nodes correspond to internal system actions not visible to the user. Solid arches correspond to interpreted user utterances.

mendations the engine can provide. Furthermore, the system needs some way of keeping track of which movies the user has seen, so the system does not recommend them again.

Thus, we have three GETTITLE nodes, each based on one of the attributes *genre*, *actor*, and *director*. The typical GETTITLE node usage is when the user has provided an attribute value (such as the name of an actor). The system then provides the user with a list of titles matching the given attribute values and asks her to identify movies that she likes. Note that this list is a non-personalized list and not a recommendation set. The GETTITLE nodes typically occur *before* any requests have been passed to the recommendation engine. Interleaved information requests can influence how the lists turn out (such as the excerpt in Figure 3). Thus, there is no hard connection between the GETTITLE node and the current recommendation base, since the titles in the list at any given moment do not need to reflect the recommendation base. This serves two purposes. First, we do not decrease the user's freedom of posing information requests, and indeed utilize these in the recommendation task. Second, it is good for the user preference profile to be as diverse as possible and not only include ratings for movies matching the current recommendation base.

RATETITLE comes into function *after* a recommendation has been proposed. Its function is to extract the rating of an already seen recommended movie, so that we constructively can utilize an otherwise "useless" recommendation, while maintaining a conversational tone in the interaction.

Performing Recommendations

SEENTITLE is one of the central nodes in the usage situation, since this is where the system presents a movie suggestion to the user. The corresponding system utterance for this node is "Have you seen this movie?" along with the title of the highest ranked recommendation. All nodes that have arches leading to SEENTITLE need to pass a check[3], since there are cases where it is not possible to traverse to SEENTITLE (i.e. perform a recommendation). This depends on the chosen recommendation engine. The SEENTITLE node is thus called only if the recommendation engine is able to deliver a suggestion. Otherwise, there is a need to continue to get ratings from the user (by returning to an appropriate GETTITLE node), or to change the current recommendation base.

Handling Changes

As pointed out above, the user may change the recommendation base. A change in the recommendation base can also arise from the system's part (e.g. to relax the constraints posed by the current recommendation base). The excerpt in Figure 5 shows an example of how the system suggests to change the recommendation base. In terms of network

S1: Have you seen *The Fifth Element*
U1: yeah / awesome
S2: It seems like we have covered all movies. Is there any other kind of movie you would like to watch?
U2: uhm / are there any movies directed by *Oliver Stone*?

Figure 5: Dialogue excerpt showing how MADFILM suggests a relaxation of the recommendation base when the matching titles have been exhausted.

traversing, S1 is an instantiation of the SEENTITLE node. The response in U1 is a positive rating of the recommended title, causing the system to return to the RECPOSSIBLE node to perform another suggestion based on the current recommendation base. Now, since all movies based on the current recommendation base have been considered, we traverse to the RELAXRECBASE node (S2). From this node there are several options, depending on the user's response. Since

[3]This check is represented as the square RECPOSSIBLE node in Figure 4.

the user provides a new recommendation base (recommendations should henceforth be based on the director in U2) the system moves to the GETTITLEDIRECTOR node according to Figure 4.

Managing Recommendation Dialogue

In case the suggested title in a SEENTITLE node is indeed unseen by the user, we have a potential recommendation. The system now needs to explain, or motivate, the recommendation objectively following the theory of building trust [2], and according to the findings in the dialogue corpus. This is done in the TOSEE node, which (a) generates an explanation by relating to the matching attributes in the current recommendation base, and (b) provides the user with the option of putting the recommended movie on the recommendation protocol. In case the user declines, the system needs to verify the current recommendation base, since this response is interpreted as negative feedback to the recommendation. On the other hand, if the user responds positively, we have a successful recommendation. The system can then add the recommended movie to the protocol and move on.

After a successful recommendation has been made the system asks if the user wants a new recommendation in the NEWREC node. A wide range of responses may follow this question. A simple "no" indicates that the session is terminated (moving the END node), whereas a simple "yes" is equally easy to handle, since we simply test if we can go to the SEENTITLE node to perform a new recommendation (after passing the RECPOSSIBLE check). However, the user may also change the recommendation base if she decides to continue the dialogue. It is easy to assume that this is because the users want variation in a set of recommendations in a session and desires e.g. one action movie, one drama comedy starring their favorite actor, and one animated movie. Examples responses to the question "Would you like a new recommendation?" include:

- "yes / something like *Gladiator* please"
- "a drama starring *Mel Gibson* would be nice"
- "do you have any animated movies?"
- "sure / give me movies directed by *Ridley Scott*"

In the case of a changed recommendation base, we traverse to the appropriate GETTITLE node (depending on which attribute(s) has been changed), in order to get a complete picture of any modifying attributes to the new recommendation base before moving on to a new SEENTITLE node.

Influencing Transitions

Several nodes in Figure 4 have multiple arches branching to different nodes. Three ways of influencing the network node transition are identified in the corpus: (a) user utterances, (b) the user preference model, and (c) database content and current recommendation base.

User Utterances

The default way to resolve transitions is to take the content of the user's response into account. This is done by having nodes check the interpreted utterance and decide which node to traverse to next. The content of the user utterance is thus the most important way to influence dialogue node transitions. However, while this is the default and most common transition influence, there are cases where the content of a user utterance may yield two (or more) equally valid system responses. We then need to consider other parameters.

User Preference Model

One alternative parameter is the user's movie preferences. This reflects that the recommender needs to know a number of preferences (ideally covering both positive and negative preferences about the bulk of all available attributes) before a qualified recommendation can be issued. It seems sound to assume that the recommender utilizes previously known preferences about movies, actors, and genres to dictate his or her next utterance.

In recommender system terms, this relates to the *density* and *size* of the user preference model [12]. Concretely, a standard collaborative filtering (CF) system is not able to calculate any prediction scores unless the user preference model has reached a certain density and size. Other types of recommendation engines have other constraints. Interfacing with the back-end system is the purpose of the RECPOSSIBLE node, and may be customized depending on the chosen engine. Thus, the size and content of the user preference model serves as an input to the dialogue nodes' transition decisions. In Figure 4, this is shown as the dashed arch from the RECPOSSIBLE node to the GETTITLE node.

Database Content and Exhausted Recommendation Base

The third transition influence is when the recommender realizes that the user's preferences takes the form of too demanding *constraints*. The recommender then asks the user to relax these constraints. This happens both when an information query from the user is too narrow, or when all movies matching the current recommendation base have been considered.

When there are no matching movies, the system must have ways to proceed if the user does not take initiative and starts introducing new preferences or search constraints. An exhausted recommendation base can thus be the reason for traversing to a RELAXRECBASE node instead of a new SEENTITLE node (see Figures 4 and 5).

IMPLEMENTATION

Hitherto we have focused on system-driven preference requests and recommendations. However, as noted above, a system implementation will also have to accommodate user-driven information requests. Fortunately, there is a fairly large body of research addressing exactly this issue. One such initiative is the phase-based PGP design pattern[4] that allows for information-providing dialogue system construction [6].

[4] PGP is hosted at the NLPFARM open source initiative (http://nlpfarm.sourceforge.net).

The dialogue strategy presented in this paper has been implemented in the MADFILM movie recommender system by adopting the PGP pattern and integrating the finite-state recommendation dialogue network with the information-providing capabilities [8]. Each node in the graph thus holds the same basic phase-based information-providing machinery, so that users can issue information requests at any time in the underlying system-driven dialogue, in line with the empirical findings in the corpus. Figures 1 and 5 exemplifies dialogue interaction in MADFILM.

MADFILM'S back-end part consists of a CF server[5] and a movie information database holding information on actors, genres, directors, and plot information. The database is used both to accommodate information requests, as well as providing attributes for the recommendation base. The recommendation engine is thus a hybrid engine [3], since it utilizes both the CF server as well as the domain-dependent database.

CONCLUSION

Acquiring user preferences in recommender systems is a non-trivial problem. Due to its flexible nature, natural language dialogue interaction is one promising approach. The work reported on here provides an empirically based model for implementing recommendation dialogue initiative. The recommendation initiative is modeled as a finite-state graph of nodes representing the system-driven preference dialogue. The arches are affected by three ways to influence transitions, namely (a) the meaning of the utterance itself, (b) the accumulated user preference knowledge, and (c) database content. User-driven information requests are modeled by re-using existing techniques from standard information-providing dialogue systems. Combining the two initiative types adds flexibility to the interaction and also functions a natural way to drive the dialogue forward, and should be utilized for unobtrusive preference acquisition. The proposed dialogue strategy has been implemented in the MADFILM recommender dialogue system. Future work includes evaluating MADFILM, as well as exploring the approach proposed here to a more general recommendation dialogue initiative strategy.

ACKNOWLEDGMENTS

This research has been supported by the Swedish National Graduate School of Language Technology (GSLT), Santa Anna IT Research, and VINNOVA.

REFERENCES

1. J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. S. Magazine. Towards conversational human-computer interaction. *AI Magazine*, 22(4):27–38, 2001.
2. A. Buczak, J. Zimmerman, and K. Kurapati. Personalization: Improving Ease-of-Use, Trust and Accuracy of a TV Show Recommender. In *Proceedings of the 2nd Workshop on Personalization in Future TV*, Malaga, Spain, 2002.
3. R. D. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370, 2002.
4. R. D. Burke, K. J. Hammond, and B. C. Young. The FindMe Approach to Assisted Browsing. *IEEE Expert*, 12(4):32–40, 1997.
5. G. Carenini, J. Smith, and D. Poole. Towards more conversational and collaborative recommender systems. In *Proceedings of the International Conference of Intelligent User Interfaces*, pages 12–18, Miami, Florida, USA, 2003.
6. L. Degerstedt and P. Johansson. Evolutionary Development of Phase-Based Dialogue Systems. In *Proceedings of the 8th Scandinavian Conference on Artificial Intelligence*, pages 59–67, Bergen, Norway, 2003.
7. P. Johansson. MadFilm: A Multimodal Approach to Handle Search and Organization in a Movie Recommender System. In *Proceedings of the 1st Nordic Symposium on Multimodal Communication*, pages 53–65, Copenhagen, Denmark, 2003.
8. P. Johansson. Design and development of recommender dialogue systems. Licentiate Thesis 1079, Linköping Studies in Science and Technology, Linköping University, April 2004.
9. A. Jönsson and N. Dahlbäck. Distilling dialogues - a method using natural dialogue corpora for dialogue system development. In *Proceedings of the 6th Applied Natural Language Processing Conference*, 2000.
10. G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: The Automated Travel Assistant. In *Proceedings of the 6th conference on User Modeling*, pages 67–78, 1997.
11. M. F. McTear. Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys*, 34(1):90–169, Mar. 2002.
12. A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: learning new user preferences in recommender systems. In Y. Gil and D. B. Leake, editors, *Proceedings of the 2002 International Conference on Intelligent User Interfaces (IUI-02)*, pages 127–134, New York, 2002. ACM Press.
13. C. Thompson, M. Göker, and P. Langley. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21:393–428, 2004.

[5]The user rating matrix is provided by the GroupLens research group (http://www.grouplens.org).