Published in: Journal of Intelligent Information Systems, Volume 58, n.1. Springer Nature, February 2022.
DOI: https://doi.org/10.1007/s10844-021-00683-4

Noname manuscript No.
(will be inserted by the editor)

An Empirical Evaluation of Active Learning Strategies for Profile Elicitation in a Conversational Recommender System

Andrea Iovine · Pasquale Lops · Fedelucio Narducci · Marco de Gemmis · Giovanni Semeraro

Received: date / Accepted: date

Abstract Conversational Recommender Systems have received widespread attention in both research and practice. They assist people in finding relevant and interesting items through a multi-turn conversation. The use of natural language interaction also allows users to express their preferences with more flexibility. However, these systems often have to work in a cold-start situation, and most of the conversation is dedicated to the profile elicitation step. In order to ensure good recommendations, this profile should be as rich as possible, which requires great user effort. In this paper, we investigate the application of Active Learning techniques for improving the profile elicitation step in a Conversational Recommender System. We compared five different state-of-the-art techniques, and carried out a user study with 219 users in order to assess their effectiveness both in terms of recommendation accuracy and user effort. Results show that assisting users by providing personalized suggestions during the profile elicitation step improves the quality of the recommendations in terms of Hit Rate and nDCG, compared to a strategy that requires users to come up with preferences on their own.

Keywords Conversational Recommender Systems · Active Learning · Profile Elicitation · Information Retrieval

A. Iovine
University of Bari Aldo Moro
E-mail: andrea.iovine@uniba.it

P. Lops
University of Bari Aldo Moro
E-mail: pasquale.lops@uniba.it

F. Narducci
Politecnico di Bari
E-mail: fedelucio.narducci@poliba.it

M.de Gemmis
University of Bari Aldo Moro
E-mail: marco.degemmis@uniba.it

G. Semeraro
University of Bari Aldo Moro
E-mail: giovanni.semeraro@uniba.it

1 Introduction

Conversational interfaces are revolutionizing the way users interact with machines. This phenomenon is strongly influenced by the great diffusion of Digital Assistants (DAs) to the general public, which is making conversational interaction more commonplace [13]. Amazon Alexa, Google Assistant, Apple Siri are some examples of these intelligent systems. The great advantage of DAs is that they support users in several tasks by interacting through natural language and voice. *"Alexa, play some music", "Hey Google, suggest me a book", "Hey Siri I'm looking for a restaurant for dinner"* are just some examples of user requests for particular services.

Overall, users appear satisfied from interacting with DAs [4]. Tasks that a user can complete with a DA can be classified as either *simple*, i.e. that can be completed in the course of a question-answer pair (e.g., *What's the weather like tomorrow?*), or *complex* (e.g., *Find a place for vacation*), which require multiple dialogue steps in order to be completed. In the latter case, the effort required for the task completion becomes a key component for user satisfaction [31].

It can be argued that the conversational recommendation scenario falls in the second category. More specifically, a Conversational Recommender System (CoRS) belongs to a particular class of recommender systems whose main characteristic is the ability of interacting with users through a multi-turn dialogue [27]. In contrast to a standard recommender, a CoRS often has to work in a cold-start situation, thus it needs to acquire user preferences before generating a recommendation. Accordingly, a CoRS performs a *complex* task, as defined above, since it is composed of at least three steps: profile elicitation, recommendation, and user feedback. Profile elicitation is especially crucial in the context of conversational recommendation, and for this reason several works investigated strategies for selecting the most informative items to propose to the user [6,37,43].

By employing natural language interaction, users can directly express their preferences and their needs. In this way, the CoRSs can suggest items using only information that was explicitly mentioned by the user during the conversation, and can focus on recommending items based on what the user is currently looking for, instead of previous purchases or other kinds of implicit feedback. Moreover, users are more aware of the information the CoRS will exploit for generating the recommendation. Natural language interaction has been previously shown to be effective for conversational recommendation [25]. For this reason, it is necessary to work on improving the quality of the preferences acquired from the user [11]. Trivially, one might think that the problem can be simply overcome by asking the user to provide more information before making a recommendation. However, increasing the profile size requires more interaction steps which, in turn, increases user effort and negatively impacts user satisfaction.

A possible solution for improving the preference elicitation step is to employ *Active Learning* approaches. These approaches are used in various Machine Learning tasks in order to dynamically select data that is most useful for training [49]. Active Learning has already been successfully applied in Collaborative Filtering Recommendation scenarios [11], but little research has been done on the integration with CoRSs.

In this paper, we investigate the introduction of Active Learning strategies into an existing conversational content-based recommender system that interacts with users via natural language. The system could previously only acquire the user

profile by asking users to provide ratings on their own. Our belief is that adding the ability to suggest items to rate may help achieve better recommendations using less interaction turns.

This paper extends the work described in [24], in which an evaluation of active learning strategies was performed, with a specific focus on non-personalized techniques. Results proved that non-personalized techniques have a positive effect on both the recommendations quality and interaction costs, even though preliminary experiments on personalized techniques support the hypothesis that the balance between popularity and diversity might have an effect on the user experience of a conversational recommender system.

Hence, we extended the work along the following directions: *(i)* we have introduced and tested a new personalized Active Learning strategy; *(ii)* we have repeated the user experiment with new participants; *(iii)* we have added an investigation on the diversity of the recommendations produced by each strategy by calculating the Aggregate Diversity and the Gini Index.

Therefore, our contributions can be summarized as follows:

- We propose an approach for integrating system-driven suggestions on a Conversational Recommender System based on natural language interaction, which use Active Learning to assist users during preference elicitation by proactively presenting informative or interesting items to rate;
- We investigate how the introduction of system-driven suggestions based on Active Learning affects the quality of the recommendations and the user experience of a CoRS;
- We identify the advantages and challenges of introducing system-driven suggestions during the profile elicitation step of a CoRS.

We performed a user study to evaluate each strategy in terms of interaction cost and recommendation quality. The user study was performed in a live setting, in which users interacted with a working version of the system. The rest of paper is organized as follows: Section 2 describes related work in the area of CoRSs and Active Learning. Section 3 describes the architecture of the CoRS used in this study. Section 4 introduces an approach for introducing system-driven suggestions in the CoRS, as well as the Active Learning approaches involved in the study. Section 5 describes the experimental protocol of the user study, while Section 6 presents and discusses the results. Finally, Section 8 draws the conclusions, and outlines future work.

2 Related Work

The work presented in this paper cuts across two main research topics: the elicitation of user preferences in a conversational recommender system, and the Active Learning strategies used by machine learning algorithms. In the following, we will analyze these two aspects with a particular focus on the interaction based on dialogue.

2.1 Preference elicitation strategies for CoRSs

A *Conversational Recommender System* (CoRS) is defined as a system that provides recommendations to users via a multi-turn dialogue [27]. CoRSs are characterized by the fact that they acquire the user profile in an iterative fashion. The system can ask the user to rate some items, and the user can influence the outcome of the recommendation by providing feedback on the suggested items. Traditional recommender systems, on the other hand, require that all user information is provided before generating a recommendation [34]. CoRSs can differ from each other in many ways. For example, they can be developed either using a traditional form-based interface, or using natural language (written or spoken). CoRSs can also be classified based on the interaction initiative (i.e., whether the user or the system takes initiative in the conversation), and the profile elicitation strategy. Users can build their profiles by either providing example items, or by providing constraints over several facets. In the case of system-driven interactions, one of the problems a CoRS has to address is how to select the questions to ask. In fact, many works investigate the next-question problem, for example by developing of strategies for choosing the most appropriate facet-based questions for CoRSs. Examples are Göker and Thompson [17], Jannach and Kreutler [26], Sun and Zhang [48], and Priyogi [41].

This kind of interaction brings to mind one of the earlier well-known interaction approaches for CoRSs, called *critiquing* [5, 21]. The principal goal of critiquing strategies is to reformulate the user query in order to best fit the items in the catalog. Generally, this feedback is used for refining the user preferences, thus it shares a goal similar to the next-question problem as described above. Our paper is especially focused on the development of *mixed-initative*, *preference-based* conversational recommenders, in which the user can provide the system with item ratings on her own accord, and the system can also prompt the user to rate items if needed. Thus, a critiquing strategy is not considered in our work.

Bertomeu Castelló [3] developed an item selection strategy for CoRSs based on a Markov Decision Process (MDP) model. The system selects the most appropriate action at any moment, i.e. asking a facet or proposing an item. However, it does not support natural language interaction, and no experiment was conducted. Greco et al. [20] developed a framework for conversational recommendation based on neural networks and reinforcement learning, in order to concurrently learn several recommendation-related tasks. In the framework, there is a meta-controller, with the role of receiving the dialogue state and predicting the goal for that state. There goal is twofold: chitchat and recommendation. A goal-specific representation module converts the dialogue state to a score vector, which is then refined by an attention module to focus on the most important parts. Eventually, a controller uses these refined scores and takes an action to satisfy the given goal. The main limitation of this work is that constraints in the dialog are not explicitly modeled, thus the dialog is completely driven by the learned model.

Christakopolou et al. [8] developed a preference elicitation framework for CoRSs, whose objective is to identify the most appropriate questions to ask the user. An experiment was conducted to compare several question-selection techniques based on active learning and bandit learning approaches. Each technique selects an item to be rated, to which the user can provide feedback. While the main goal of this study is very similar to ours, the two studies are not directly comparable, as the

approach, research questions, and the experimental protocol used are very different. First of all, the system presented in [8] is based on a Probabilistic Matrix Factorization (PMF) recommendation algorithm, while ours is based on a Content-Based algorithm. Second, one of the main points of [8] is the comparison between absolute and relative feedback. The study concluded that the absolute model was able to achieve better performance. Our work compares several question selection models based on absolute feedback against a model that is completely *user-driven*, in which users provide preferences on their own. Finally, the experiment in [8] is conducted by generating a bootstrapped ground truth from a user questionnaire. Instead, we performed a live experiment, in which users tested a working system, and directly evaluated the recommended items.

Bandit learning algorithms are also implemented by Parapar and Radlinski [40]. The authors investigate how to improve the preference elicitation step in a (generic) recommender system. Indeed, their approach is independent of any particular recommendation algorithm, and results in broader user profiles. They propose to diversify the preferences elicited using Multi-Armed Bandits. This leads to improved diversity and serendipity of recommendations. The goal of our investigation is quite different since we compare state-of-the-art solutions for active learning in the specific context of CoRSs. However, we also investigate active learning methods based on popularity and diversity, although the strategies are different from those proposed in [40].

FPAN (Feedback-guided Preference Adaptation Network) is proposed by Xu et al. [53] with the aim of improving the preference elicitation step of a CoRS. The authors define a model for adapting the original user embeddings according to online item and attribute-level feedback. Experiments demonstrated that FPAN outperformed state-of-the-art baselines in terms of user preference estimation. Although the main goal is similar to ours, this model is only applicable to end-to-end architectures.

2.2 Active Learning approaches

Active Learning [49] is defined as "the process of guiding the sampling process by querying for certain types of instances based on previously seen data". Its objective is to select the training data to feed a machine learning technique, in order to improve the efficiency of the data. The examples that are most useful for the prediction task are selected, while the most uninformative are discarded. *Active Learning* strategies have been extensively researched in the area of recommender systems as a potential way to improve the efficiency of the profile acquisition process. In fact, they can help select the most informative items for the recommendation task. Accordingly, Active Learning can play a strategic role for supporting the preference elicitation step. Active Learning is especially useful when training data is scarce. This is relevant for CoRSs, since they frequently work in cold-start condition.

In [11,12], Elahi et al. provided a survey on the state of the art regarding Active Learning techniques for recommender systems based on collaborative filtering. These techniques can be divided into personalized and non-personalized. Both categories can in turn be classified as *single-heuristic* or *combined-heuristic*. Non-personalized strategies do not take into account the active user's previous

ratings, and only aim to select the items that are most popular, diverse or controversial, such as Merialdo [35], Rashid et al. [43,44], and Golbandi et al. [18,19]. Personalized strategies select items based on what was previously rated by the current user. Some examples are described in Rubens et al. [46] and Lee [33].

Hernández-Rubio et al. [23] propose a novel Active Learning approach focused on opinions about item aspects extracted from user reviews. The proposed approach outperformed state-of-the-art strategies in terms of both rating prediction and ranking. Aspects are extracted through a vocabulary-based aspect extraction method. Items with the highest similarities with the user's previously rated objects are selected. While the proposed model was implemented in a traditional recommender system, Aspect-based Active Learning could represent an interesting and promising extension of our work.

Chuan et al. [9] propose a chatbot specialized for determining the eligibility criteria for clinical trials using Active Deep Learning. The proposed chatbot simplifies the process for allowing users to participate in clinical trials. The complex and domain-specific criteria required to assess eligibility are separated into questions that users can answer. The sampling is getting from the *uncertainty cluster*, thus the algorithm performs clustering on uncertain cases and selects the centroid of the cluster to query the human oracle for the class label. The Active Learning algorithm is used for improving the accuracy of a Convolutional Neural Network (CNN). This is a very interesting research direction since CoRSs often make use of deep learning architectures.

The work presented in this paper fits well in the current state of the art: while there is a large quantity of research regarding the topics of Conversational Recommender Systems and Active Learning, very few works actually investigate the idea of combining the two areas, by integrating Active Learning strategies during the conversational recommendation scenario. Our intention is to bridge this gap by performing a user study that involves the integration of several Active Learning strategies into a natural language-based Conversational Recommender System. Accordingly, the principal goal of our study is to compare state-of-the-art approaches for Active Learning (as reported in Section 4) for assessing how they work in the context of conversational recommendations.

3 Conversational Profile Elicitation with Active Learning

The experimental object of this user study is a Conversational Recommender System (CoRS) specialized in providing movie recommendations. The system uses an interface based on natural language: users can interact with the system by writing text messages, and will receive feedback in the form of text and images. In the original system, user profile elicitation was only performed by asking users to talk about movies that they like or dislike, without any intervention from the system. For the purpose of this study, we integrated an item suggestion functionality into the system, which will be further explained in Section 4.

The interaction process of the CoRS follows three steps: user profile elicitation, recommendation, and user evaluation of the recommendations [25]. These steps are repeated over time until the recommender system has enough data to generate a satisfactory recommendation.

During the profile elicitation phase, users can express preferences to both *items* and their *properties*, i.e. characteristics or features that describe the items. In the movie domain, properties can be concepts such as actors, directors, or the genre. This distinction is consistent with the existing literature [30]. Both movies and their properties were extracted from Wikidata.[1]

Users can give a positive preference to an item to signal the desire to receive recommendations that are similar to it, or can express a negative preference otherwise. Users can also talk about some properties of the items that they like or dislike, which helps the recommender system focus on the characteristics that are most important. For example, in the movie domain, the user can write something like *"I like Mel Gibson, but I don't like Braveheart"* (Figure 2). In this case, the user expresses two preferences: a positive one for the property *Mel Gibson*, and a negative one for the item *Braveheart*.

When the system has acquired enough ratings, the user can then proceed by requesting some recommendations, e.g. by writing *"What can I watch tonight?"*. During the recommendation phase, the system proposes a set of recommended items, each of which can be either accepted or rejected by the user. Figure 3 shows an example of recommendation provided by the system.

The system and the data used in the experiment have been made available publicly on a Github repository[2].

3.1 CoRS architecture and implementation

The architecture of the CoRS (shown in Figure 1) mirrors that seen in Goal-Oriented Conversational Agents described in Williams et al. [52]. The system is made up of several components, each with its specific functions and responsibilities. One advantage of this approach is that modules are mostly independent from one another, and can be interchanged easily. For example, the module that handles the recommendation algorithm can be changed without affecting the dialogue model. Figures 2, 3 and 4 show some examples of interaction with the CoRS. The following sections describe the components in detail.

3.1.1 Natural Language Understanding

This component has the responsibility of understanding the user's message, and extracting all information contained within. This information is composed of: 1) the *intent*, i.e. the action or request made by the user; 2) the *entities* mentioned in the text, and 3) the *rating* expressed to each entity.

Intent recognition (IR) is the first step to understanding the user's message. The intents supported by our CoRS are consistent with those described in [27], and are detailed in Table 1.

The *provide preferences* intent is recognized when the user expresses a preference to one or more items or properties during the profile elicitation acquisition step. An example of interaction with this intent is shown in Figure 2. The *request*

[1] `https://www.wikidata.org`

[2] `https://github.com/aiovine/conversational-recommender-jiis/tree/master`

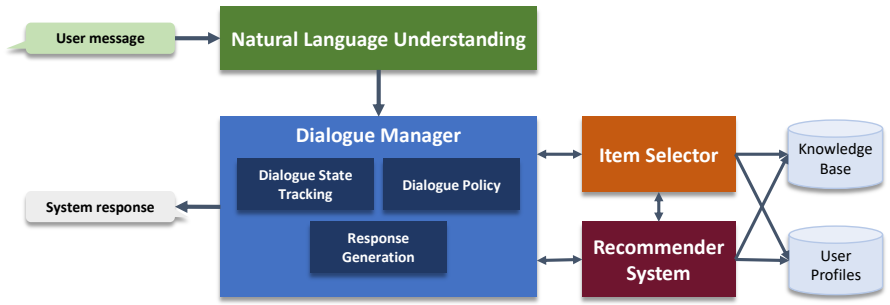

Fig. 1 Architecture of the CoRS

a recommendation intent is used when the user is ready to start the recommendation. Figure 3 shows an example of a user requesting a recommendation to the CoRS. The *feedback on recommendation* intent is activated when the user responds to an item recommended by the CoRS, e.g. by accepting or rejecting it.

For this experiment, we included a new intent called *feedback on suggestion*, which is recognized when the user provides feedback to an item that was suggested by the system. Figure 4 shows an example of system suggestion and user feedback. Section 4 will provide more details about how suggestions are generated and how feedback is handled. Intent recognition is treated as a *text classification* task, in which the entire message is classified against each of the aforementioned intents. The intent recognition functionalities of our CoRS are implemented using Dialogflow.[3]

When the *provide preferences* or *feedback on recommendation/suggestion* intents are recognized, the Natural Language Understanding (NLU) component is also tasked with understanding *what* the user is talking about by performing *Named Entity Recognition* (NER) on the text. The goal of NER is to extract *entity mentions* from text, and match them to entities in the knowledge base. We employed a custom-built NER solution that exploits knowledge graphs such as Wikidata. This component performs two steps: *spotting* and *linking*. In the spotting step, the algorithm analyzes the text in order to discover candidate entities.

[3] `dialogflow.com/`

Intent name	Description	Example
Provide preferences	The user has provided one or more preferences to the system	*I like Mel Gibson, but I don't like Braveheart*
Provide clarifications	The user is providing clarification to a previous preference (e.g. when multiple entities match the user input)	*I mean Ghostbusters (1984)*
Request a recommendation	The user is asking for movie recommendations, starting the recommendation phase	*What can I watch tonight?*
Feedback on recommendation	The user is responding to a system recommendation, by accepting or rejecting it	*I like this movie*
Feedback on suggestion	The user is rating a movie that the system proposed during the suggestion phase	*I like it, but I don't like the genre*
Show profile	The user is asking to review his/her profile	*Can I see my profile?*

Table 1 List of intents supported by the CoRS

In particular, the algorithm detects sequences of words (the *surface form*) matching a Wikidata alias, and then all the concepts that can be associated to the alias are retrieved. For the linking step, it uses holographic graph embeddings [39] to exploit the relations between concepts in the knowledge graph. In case multiple concepts can be assigned to a surface form, a *disambiguation* step is performed. Following the *one topic per discourse* hypothesis, it selects the concept that is more similar to the other concepts in the text. When the NLU component is not able to resolve the ambiguity in one or more entities, the system will ask the user to select from a list of candidate entities. The answer to this question will activate the *provide clarification* intent.

Finally, the *show profile* intent is activated whenever the user wants to review his/her profile.

Sentiment Analysis (SA) is then performed to extract the rating associated to each entity. To do this, we employed the Sentiment Tagger provided by the Stanford CoreNLP library.[4] The component assigns a positive or negative rating to each of the previously retrieved entities. To do this, *sentiment words* are identified from the text, and each sentiment word is then associated to the closest entity.

In the example shown in Figure 2, the user writes *"I like Mel Gibson, but I don't like Braveheart"*. In this case, the output of the IR is the *provide preferences* intent. The NER extracts the mentions to the property *Mel Gibson* and the movie *Braveheart*. Finally, the SA extracts the sentiment words *like* and *don't like*, and assigns a positive rating to the first entity, and a negative rating to the second.

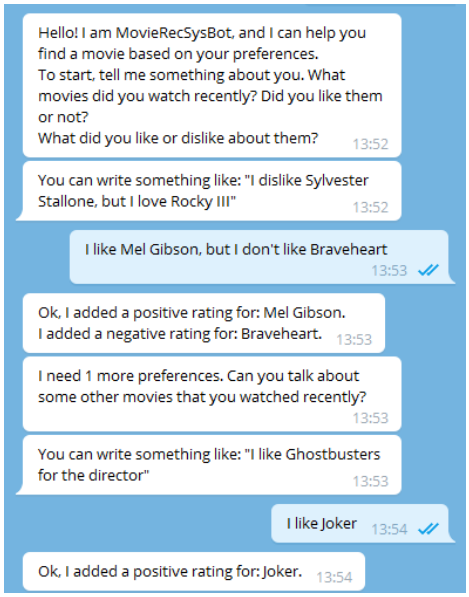

Fig. 2 Screenshot of the system during the profile elicitation phase

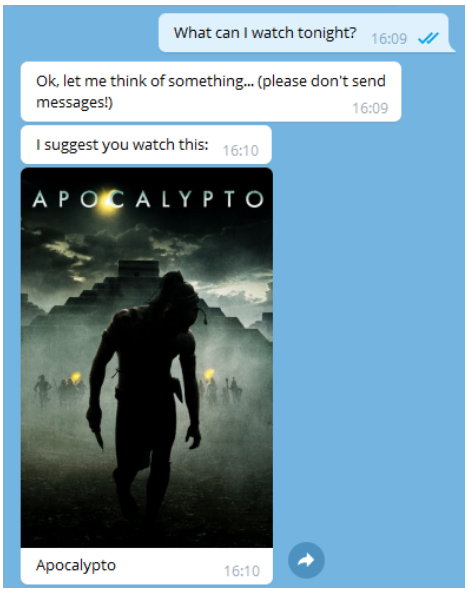

Fig. 3 Screenshot of the system during the recommendation phase

[4] https://stanfordnlp.github.io/CoreNLP/

3.1.2 Recommender System

This component is responsible for handling the recommendation process, as well as the user profiles. The algorithm used is the PageRank with Priors, also known as the Personalized PageRank (PPR) [22]. Recommendations are generated using a knowledge base of movies and movie properties extracted from Wikidata. The knowledge base is organized as a graph, in which each node is either a movie or a property (actors, directors, etc.). PPR adds a non-uniform personalization vector, that assigns different weights to different nodes in the graph. In our case we adopted the default distribution, *i.e.,* 80% of the total weight is evenly distributed among items liked by the user, and 20% is evenly distributed among the remaining nodes. As confirmed by recent research in the area [2, 7, 36, 50] PPR provides results in line with the most popular recommendation strategies. Moreover, one of the strengths of this algorithms lies in the fact that it can leverage preferences to both items and properties.

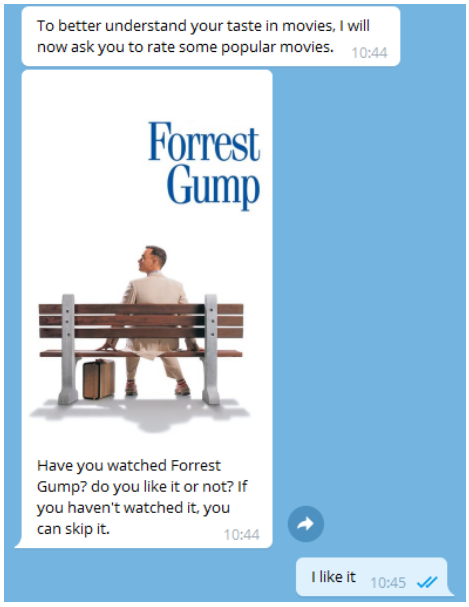

Fig. 4 Screenshot of the system suggesting an item during profile elicitation

3.1.3 Dialogue Manager

This component orchestrates the entire recommendation process. It handles three main tasks: *Dialogue state tracking*, i.e. keeping track of the current state of the conversation; *Dialogue policy*, i.e. selecting an appropriate action to perform given the current state; and *Response generation*, i.e. organizing the outputs of the other components in order to build the final response that will be sent to the user.

The Dialogue Manager was developed completely in-house. Dialogue state tracking and policy selection are performed using a rule-based approach: the ac-

tion performed by the system is chosen based on the current dialogue state and the intent recognized by the Natural Language Understanding component. Once the action has been performed, the dialogue state is updated accordingly, and a textual feedback is provided to the user.

For the Response generation task, we chose a simple template-based model: for each action that the system can perform, a template response is associated. When the action is performed, a text feedback is generated by filling in the template with contextual data, and then it is displayed to the user.

3.1.4 Item Selector

The Item Selector component was implemented for the purpose of this study. It is responsible for proposing movies during the suggestion phase of the CoRS, and supports several item selection strategies, which will be further detailed in Section 4. This component also cooperates with the Recommender System, as it uses the knowledge base and/or the user profiles to select the items to suggest (depending on the strategy used).

4 Mixed-initiative profile elicitation with Active Learning

The user profile elicitation strategy used in the original system is largely *user-driven*. The onus of providing preferences is mostly on the users, which can make the interaction slower and more frustrating as they have to come up with enough items and properties to rate before the recommendation can begin. We propose to overcome this limitation by making profile elicitation a *mixed-initiative* process, i.e. the flow of the conversation can be controlled by both the user and the system in different moments. Effectively, the two parties collaborate to reach the ultimate goal, i.e. finding a useful recommendation. We enforce mixed-initiative dialogue by dividing the profile acquisition into two phases: *warm-up* and *suggestion*. The new workflow of the interaction process is shown in Fig. 5.

During the warm-up phase, the system introduces itself, and prompts users to share their preferences about both items and their properties. This phase is reminiscent of the profile elicitation strategy employed by the original system. Once enough ratings are provided, the *suggestion phase* is activated. The system asks the user to express a preference to several popular or interesting movies, with the objective of fine-tuning its understanding of the user's taste. The advantage of natural language interaction is that the user is not limited to simple like/dislike answers, but can also provide more detailed feedback. In fact, users can *criticize* the suggested item, by giving specific feedback to a property of the item. In practice, system suggestions become a starting point for further conversation. In Fig. 5, the system presents the question *"Have you watched Ready Player One? Do you like it?"*, and the user answers with *"I like that movie, but I don't like sci-fi movies that much"*.

Ideally, the items shown during the suggestion phase should be chosen by maximizing the quantity of information that can be elicited from the user, while minimizing interaction effort. In this sense, *Active Learning* [49] is definitely relevant. It is defined as "the process of guiding the sampling process by querying for certain types of instances based on previously seen data". Its objective is to select

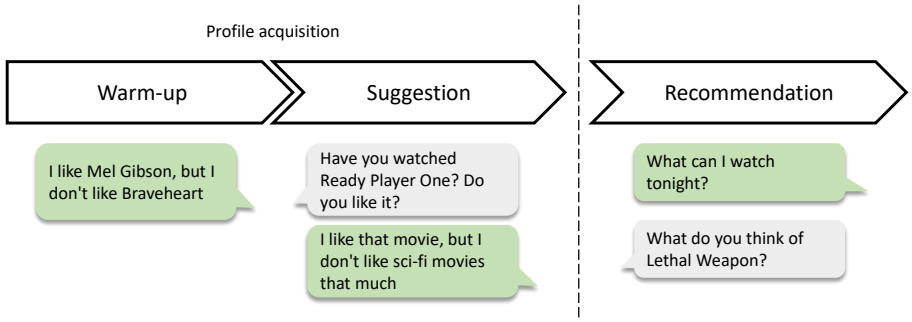

Fig. 5 Workflow of the CoRS

the training data to feed to a machine learning technique, in order to improve the efficiency of the data. The examples that are most useful for the prediction task are selected, while the least informative ones are discarded.

Many Active Learning strategies exist for recommender systems [12], each of which focuses on maximizing one aspect of the suggested items, such as their diversity, or the likelihood that the user can confidently provide a rating on them. Choosing a strategy over all others is not trivial, as there is no *one-size-fits-all* solution for achieving good results.

When dealing with natural language, the effect of a specific suggestion strategy becomes more nuanced. For example, suggesting items that are well-known increases the probability that the user will be able to generate more complex feedback. On the other hand, presenting a small set of diverse items can potentially lead to a more informative user profile. However, if the suggestions become too specific or obscure, users will not know what to say about them, making the interaction more frustrating.

In this section, we describe the Active Learning strategies that we have investigated for empowering the suggestion phase of our CoRS.

4.1 Popularity

This strategy selects the most popular items, i.e., those with the highest number of ratings [43, 12]. It belongs to the *single-heuristic non-personalized* category, since it does not take the user's previous preferences into account. It also falls into the category of *attention-based* strategies, since it focuses on finding items that have received the most *attention* among users in the past.

The principle behind this approach is simple: popular items are more likely to be also known by the current user, which in turn decreases the chance that he/she does not know how to rate it. This is important in the context of a CoRS, since it means that users have to spend less time rating items, and can potentially receive a recommendation with less interaction turns.

The conversational interface can also bring out another potential advantage: in fact, users are not limited to provide a simple *like/dislike* rating, but can also write more complex feedback via text. Popular items can be a good starting point for

the user to provide more feedback, simply because he/she may have more things to talk about them.

A problem commonly observed when using this strategy is the *prefix bias* [43]: because the user profile is mainly acquired through popular items, the recommender system is also biased towards recommending other popular items, thus ignoring the long tail of less popular items.

4.2 Random Popularity

Random Popularity is one of the first examples of *combined-heuristic non personalized* Active Learning strategy, which was first introduced in the MovieLens recommender system [12]. In the original definition, preference elicitation was performed by showing a list of movies that users could rate. This list was composed of randomly selected movies, plus one that was selected from a manually-sourced repository of popular movies. This increases the probability that users have experienced at least one of the proposed movies, while avoiding the most common pitfalls of a strategy purely based on popularity.

Some modifications are required in order to fit this approach into our conversational interface, because suggestions are provided one after another, and users are asked to provide a preference to each of them. First, we extracted the Top 250 most popular movies from IMDb.[5] During preference elicitation, the system can do two things:

1. Ask the user to rate a movie from the popular list. The movie is chosen randomly (among those that were not already rated by the user);
2. Ask the user to rate a completely random movie.

Both options have a 50% chance of being selected, therefore the user will be asked to provide preferences to a balanced combination of popular and less popular items. Suggesting random movies can however lead to some problems, as the likelihood that the user can express an opinion on them is significantly lower. The effectiveness of the suggestion strategy is thus reduced, as well as the efficiency of the interaction.

4.3 $\sqrt{Popularity} \times Variance$

This strategy tries to combine the effect of the popularity score with heuristics that take into account the informativeness of the ratings [18]. It is a *combined-heuristic non-personalized* strategy that selects popular items that also have a large variance of ratings. Preferences on these items may provide more useful (discriminative) information about the user's profile.

The strategy is a variant of the $log(popularity) \times entropy$ one. Applying logarithm or square root allows to reduce the weight of popularity, which usually has an exponential distribution, in order to avoid the dominance on entropy or variance when the two values are multiplied. Research shows that this is preferred to a strategy that only selects items based on either variance or entropy, which

[5] `https://github.com/jberkel/imdb-movie-links/blob/master/top250.txt`

tends to select unpopular items that are less likely to be relevant to the current user's interests [19]. Given a candidate item i and R_i, the set of ratings associated to i, we calculate the score as follows:

$$score(i) = \sqrt{|R_i|}\frac{\sum_{r \in R_i}(r - \mu_{R_i})^2}{|R_i|}, \tag{1}$$

where μ_{R_i} is the average of the ratings in R_i. We then select the item with with the highest score, and present it to the user.

4.4 Item-to-Item

This is a *single-heuristic personalized* strategy, which takes into account the item similarity to the user's previously rated items. To calculate this similarity, two elements are needed: a way to represent items, and a function that calculates the distance/similarity between two items. The implementation of the item-to-item strategy found in [12] uses a standard item-based collaborative representation, in which an item is described by the ratings it received by all users. The Pearson correlation coefficient calculates the similarity between items by analyzing the subset of users that rated both items.

We instead opted for a content-based representation of the items, which is also used by the recommendation algorithm. Each item is described by a set of *properties*. In the movie domain, properties can be things such as the actors, the genre, and the director. We then measure the distance between two items by calculating the Jaccard Index between the sets of properties that describe them. The Jaccard Index is often used to gauge the diversity of two sets [45]. Given two items i and j, and their respective set of properties P_i and P_j, we calculate the similarity using the following formula:

$$sim(i, j) = \frac{|P_i \cap P_j|}{|P_i \cup P_j|} \tag{2}$$

Essentially, the similarity score takes into account the properties that are shared between the two items. For example, two movies could be considered similar because they share the same director, or belong to the same genre, or because one or more actors participated in both.

During the interaction, we calculate the similarity between the candidate items and those already in the user profile, using the aforementioned formula. The warm-up phase described earlier ensures that the similarity can be calculated correctly. Differently from [12], we do not ask the user to rate the most similar movies. Instead, we chose to select the *least similar items*, among those that meet a minimum similarity threshold.

This decision was made to differentiate this strategy from the ones that are described earlier in this section. While the popularity-based strategy focuses entirely on the *familiarity* of suggested items, here we employ a strategy that is dedicated to maximizing their *diversity*. A potential advantage is that this diversity can help build a more accurate user profile [38].

4.5 Highest Predicted

This is another *personalized*, *single-heuristic* approach, which falls specifically in the *prediction-based* category. In fact, this strategy relies on asking users to express a preference to items that he/she probably likes. In order to find those items, a *rating prediction* algorithm is adopted, and the item with the highest predicted rating is returned. Hence, this strategy exploits the output of a recommender system during the preference elicitation step.

In our case however, we decided not to use our recommendation algorithm, and instead generate suggestions using collaborative filtering, in order to avoid redundancy between the output of the item selection strategy and that of the recommendation phase.

Specifically, we used a matrix factorization model based on Singular Value Decomposition (SVD++) [32]. To sum up, SVD++ is used as rating prediction algorithm and the item with the highest predicted rating that has not been rated by the user will be suggested, because it is more likely to be interesting to the user, and worth to be rated.

5 Experiment

5.1 Experimental Protocol

We performed a between-subjects in-vivo user study that involved 219 participants (women = 14.48%, medium-high interest in movies = 92.76%), most of which were University students. Participants were collected via voluntary sampling. The experiment involved six configurations of the recommender system described in Section 3. Five configurations use one of the item selection strategies that were described in Section 4. The sixth configuration disables the suggestion phase, thus profile elicitation is performed exclusively by asking users to provide preferences on their own, without any assistance from the system. This will serve as the baseline configuration. The participants were randomly divided into six groups, each group having a number between 34 and 40 participants. Each participant tested only one of the configurations, and performed the following steps:

1. Before starting the experiment, each participant is required to read the instructions, which describe the purpose of the test, the tasks, and some examples of use.
2. At the beginning of the test, the system introduces itself to the participants. Then, the warm-up phase begins: each participant has to provide at least three ratings to movies or properties on his/her own. For the baseline configuration, we decided to increase the minimum number of user-given ratings during the warm-up phase to six. This was chosen to decrease the difference in profile size with the other configurations, in which the user profile always contains at least eight ratings.
3. After the warm-up phase, the suggestion phase begins. The system proposes a set of five movies, chosen using one of the item selection strategies described in Section 4. The participants can rate each item, skip it, or provide additional feedback. This step is ignored for the baseline configuration.

4. After enough ratings have been elicited, participants can ask the system to recommend some movies. At this point, the recommendation phase is started, and five recommendations are generated. Each recommendation can be accepted, rejected, or skipped. Also, participants can request more details, the trailer, or an explanation.
5. At the end of the test, participants are required to fill out a post-test questionnaire.

5.2 System configuration

We configured the conversational recommender system for the movie domain. To do this, we generated a movie knowledge base by extracting movies and their properties from Wikidata. As a result, the knowledge base contains 17,155 movies and 372,557 properties. The Natural Language Understanding component is trained to recognize 61,559 distinct entities in the movie domain (movies, actors, directors, genres, etc.).

For the *Popularity*, $\sqrt{Popularity} \times Variance$ and *Highest Predicted* configurations, which rely on previous ratings, we decided to bootstrap the database of user profiles by including ratings collected from the Movielens *latest-small* dataset[6]. This is a necessary step to ensure that all strategies are treated fairly, i.e. by providing enough data to function correctly. This also provides the added benefit of increasing the consistency of the suggestions across all users, because they will not be strictly dependent on the interactions collected from previous users.

5.3 Research Questions

- **RQ1: What is the effect of Active Learning techniques on the recommendation quality of a CoRS? How does it compare against a model based only on user-given preferences?** The suggestion strategies described in Section 4 use different criteria to decide which items to use for profile elicitation. Therefore, the profiles that will be captured will possess different characteristics, which in turn should result in different recommendations. We are interested in understanding how these strategies affect the users' perception of the recommendations, and if they can actually improve the recommendation process compared to a preference elicitation strategy that relies entirely on the user. Accordingly, we collect feedback on the output of the recommender system, as well as the users' opinion through the answers to the questionnaire.
- **RQ2: What is the effect of Active Learning techniques on the interaction cost of a CoRS? How does it compare against a model based only on user-given preferences?** Suggesting different items leads to differences in the quality of the interaction as well. We collect a set of metrics that measure the users' attitude in using the CoRS, as well as record their opinion on the user experience through the questionnaire. This will help us understand if the aforementioned strategies are beneficial in making the interaction more efficient and pleasant for users.

[6] `https://grouplens.org/datasets/movielens/`

5.4 Metrics

During the experiment, we collected a set of metrics that we use to determine the answer to the aforementioned research questions. In order to answer RQ1, we collected a set of state-of-the-art metrics for evaluating the quality of the recommendations:

- **Hit Rate (HR):** It is the average number of *hits*. For each user, we register a hit if at least one item in the recommendation list was deemed satisfactory. It assesses the system's ability to reach the user's goal, i.e. find a good recommendation.
- **Accuracy (ACC):** It measures the ratio between liked recommendations and all recommendations presented by the system.
- **Mean Average Precision (MAP):** The average precision of the recommendation lists, calculated by taking into account the items liked by the user and their ranking. It measures the quality of the ranking of the recommendations [47].
- **normalized Discounted Cumulative Gain (nDCG):** This is another metric of ranking quality commonly used in Information Retrieval (IR). It measures the utility of each recommendation, which is logarithmically discounted based on its position in the recommendation list [28].
- **Aggregate Diversity (AD):** It measures the recommender systems performance based on the top-N recommended items lists that the system provides to its users. It is computed as the total number of distinct items recommended across all users [1]. In order to carry out a fair comparison among configurations involving different number of users, we also report the aggregate diversity as a percentage of the maximum total number of possible diverse recommendations. For example, if we recommend 5 items to 40 users, the maximum number of possible diverse recommendations is 200, while with 20 users we could recommend a set of 100 possible diverse recommendations. Hence, an aggregate diversity of 100 items corresponds to a percentage of 50% and 100% for the the two cases, respectively.
- **Gini Index (GI):** It measures the degree to which recommendations are concentrated on a few popular items (i.e., low diversity), or conversely, how much they are equally distributed across all candidate items [15]. A Gini index equal to 1 means that recommendations are concentrated on a single item in the knowledge base, while an index close to 0 means that recommendations are evenly distributed on all recommendable items.

In order to answer RQ2, we measured the cost of interacting with the system via a set of metrics commonly used for evaluating conversational agents.

- **Number of Questions (NQ):** It counts how many questions are asked by the system during the profile elicitation step. Precisely, questions are asked either when the system prompts the user to give a preference to an item during the suggestion phase, or whenever some additional information is required in order to fully understand the user's message (e.g. when multiple concepts match with an entity written by the user). Asking too many questions could be an indicator of a more complicated user interaction.

- **Average time per question (TPQ):** It is the average time (in seconds) taken by the user to answer a question asked by the system. If the system asks difficult questions (e.g. by suggesting an unfamiliar item during profile elicitation), the time needed to answer this question may increase.
- **Interaction Time (IT):** It is the average time (in seconds) taken by the user to complete the experiment.
- **Query Density (QD):** Inspired by Glass et al. [16], it is defined as the average number of new concepts (in our case, ratings to movies or properties) introduced in a user message. A higher QD means that the user can provide more complex ratings. The QD is computed by the following formula:

$$QD = \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{N_u(i)}{N_m(i)}, \tag{3}$$

  where $N_u(i)$ is the number of distinct concepts introduced by the user u in the dialogue i, $N_m(i)$ is the number of messages of the user in the dialogue i, and N_d is the number of dialogues.
- **Conversation Length (CL):** It measures the average number of *interaction turns* that take place during the experiment. Longer conversations could be a potential indicator of a less efficient interaction.

We have also collected the answers to the questionnaire, which is based on the ResQue model [42]. The questionnaire aims to assess the quality of each configuration from the users' point of view. Questions are organized into eight constructs: *Ease of use, Control, Transparency, Interaction Adequacy, Recommendation Accuracy, Intention of use, Perceived usefulness, Overall Satisfaction*. Answers are provided on a 5-point Likert Scale, with 1 meaning *Strongly Disagree*, and 5 meaning *Strongly Agree*.

6 Experiment Results

The values of the metrics collected during the experiment are presented in Table 2. We also performed statistical tests to verify the significance of the differences between the configurations. In particular, we used the *MANOVA* test, followed by *t-test for independent samples* to check significant differences between pairs of configurations. The t-test is performed for all metrics except for AD and GI, since they are aggregate. For both tests the significance level is 0.05, however, we mitigate the effect of multiple comparisons by applying Bonferroni correction to the t-test results [10]. Since each group has a sample size that is between 34 and 40, we can safely assume a normal distribution for the metrics. Moreover, we also report the statistical power of each test.

The MANOVA test confirms that the differences in the metrics between all configurations are significant ($p < 0.001$, Pillai's Trace $= 0.56$, $F = 3.31$, $NumDF = 40$, $DenDF = 1050$).

Metric	No sugg.	Pop.	Rand Pop.	Pop.Var.	I2I	Highest Pred.
NQ	**3.73**	11.94	12.14	12.57	8.89	22.03
TPQ	39.84	23.32	24.54	29.21	31.60	**20.80**
IT	790.42	**653.64**	751.41	805.40	834.50	862.91
QD	1.28	1.26	1.08	**1.29**	1.08	1.26
CL	**19.35**	26.25	25.86	28.71	23.78	35.24
HR	0.788	0.882	0.919	0.890	0.919	**1.000***
Accuracy	0.398	0.518	0.516	**0.533**	0.430	0.529
MAP	0.288	**0.422**	0.400	0.402	0.301	0.397
nDCG	0.603	0.728	0.767	0.695	0.718	**0.818***
AD	**174**	130	152	124	160	141
%AD	**87%**	72%	82%	71%	86%	83%
GI	**0.120**	0.241	0.158	0.256	0.122	0.157

Table 2 Results of the metrics for the user study. The highest scores are highlighted in bold. *Means that the configuration performs significantly better than the *No Suggestions* baseline.

6.1 Interaction Quality Metrics

The first noticeable result is that the No Suggestions configuration asks less questions compared to the others. This is expected, since, as we said earlier, we counted each movie suggestion as a question. The t-test confirms that the difference between No Suggestions and all Active Learning strategies is significant ($p < 0.001$, $t > 7.12$, statistical power $> 99\%$).

We can also see that the Item-to-Item configuration seems to ask a lower number of questions compared to Random Popularity, and $\sqrt{Popularity} \times Variance$ and Highest Predicted strategies. This can happen because the suggestion phase may be interrupted when no suitable movies can be suggested by this strategy. The t-test confirms the difference in NQ between Item-to-Item and Random Popularity ($p = 0.0028$, $t = 3.10$, statistical power $= 87.2\%$), Item-to-Item and $\sqrt{Popularity} \times Variance$ ($p = 0.0025$, $t = 3.20$, statistical power $= 78.1\%$), Item-to-Item and Highest Predicted ($p < 0.001$, $t = 4.82$, statistical power $= 99.7\%$).

Out of all Active Learning configurations, Highest Predicted reports the highest number of questions asked. By analyzing the interaction logs in detail, we discovered that few users skipped a large amount of suggestions, thus increasing the number of questions asked by the system, since it has to provide more suggestions before the recommendation phase can start. It is likely that in these cases the item selection strategy did not receive adequate information during the warm-up phase, in which case the SVD++ algorithm returned irrelevant or uncommon movies. Nevertheless, this has not affected recommendation performance.

In terms of TPQ, Highest Predicted achieved the quickest answer time, while Item-to-Item was the slowest. This suggests that the personalized suggestions made by the Highest Predicted approach make it easier for users to quickly rate. Popularity achieved a close second for this metric. When diversity is prioritized, the answer time becomes longer, as users will be faced with unfamiliar items more often. Despite this, the t-test did not find any significant differences in TPQ.

Users reported a lower IT for the Popularity-based technique, while the highest value is reported by the Highest Predicted configuration. This is again influenced by the fact that some users skipped a large number of suggestions, thus the profile elicitation phase required more time before enough data was collected. The differ-

ence in IT between Popularity and Highest Predicted is significant ($p = 0.0044$, $t = 2.95$, statistical power = 83.8%). This gives credit to the hypothesis that popular items are easier to rate for users, thus allowing them to complete the preference elicitation phase quicker. By introducing variance with the Random Popularity, $\sqrt{Popularity} \times Variance$, and Item-to-Item configurations we observe an increase in IT, which is also consistent with the previous hypothesis.

In terms of the overall conversation length, we can see that the Active Learning-based configurations require on average more messages before the interaction can be completed. This was expected, as participants had to rate at least five movies proposed by the Item Selector. The t-test confirms that the difference in CL between No Suggestions and all other configurations is significant ($p < 0.003$, $t > 3.1$, statistical power $> 86.8\%$). Additionally, the Highest Predicted configuration reports a higher CL than Popularity ($p = 0.008$, $t = 2.74$, statistical power = 77.7%), Random Popularity ($p = 0.003$, $t = 3.11$, statistical power = 86.3%), and Item-to-Item ($p < 0.001$, $t = 3.64$, statistical power = 94.9%). This is justified by the higher number of skipped suggestions.

Query Density is very close across configurations, with only the Item-to-Item and Popularity strategies scoring slightly lower. On one hand, this is an indication that the proposed strategies do not negatively affect the efficiency of the interaction. However, it also means that the hypothesis that system suggestions can help users provide more complex feedback cannot be confirmed. In fact, participants often limited themselves to rating the suggested item itself. The t-test confirms that the differences in QD are not significant.

6.2 Recommendation Quality Metrics

As for recommendation quality metrics, we can observe that the Highest Predicted configuration managed to achieve a Hit Rate of 1, which means that all participants managed to find at least one satisfactory item within the 5 generated by the recommender system. In general, all Active Learning-based configurations managed to obtain a higher HR compared to the baseline. However, only the difference between No Suggestions and Highest Predicted is significant ($p = 0.003$, $t = 3.05$, statistical power = 88.6%). Similar considerations are made for the nDCG metric: all Active Learning configurations score higher than the baseline, with Highest Predicted obtaining a significant improvement ($p = 0.001$, $t = 3.34$, statistical power = 92.9%). Thus, we can say that the profile acquired through the use of Active Learning strategies can lead to a higher quality of both the recommendations and their ranking. Therefore, providing suggestions to users during the preference elicitation phase of a CoRS is indeed beneficial to its effectiveness, because it supports users in coming up with a set of preferences that is representative of their tastes. Moreover, tailoring the suggestions based on what the user has provided during the warm-up phase seems to increase this benefit, as demonstrated by the fact that the Highest Predicted configuration scored the highest for HR and nDCG.

The $\sqrt{Popularity} \times Variance$ configuration obtained scores similar to Popularity for the recommendation quality metrics. This suggests that introducing variance in the suggestion strategy does not produce an appreciable increase in the informativeness of the acquired user profile. Indeed, no significant differences were found between the two configurations.

The recommendations generated by the Active Learning configurations have not achieved higher diversity compared to the *No Suggestions* baseline, according to the AD and GI metrics. The results show that, by letting users provide preferences on their own, the recommender system is already able to achieve very high diversity. In fact, only 13% of the recommendations proposed by the baseline were shared across two or more users. Instead, the suggestion strategies that rely on familiarity introduce some *homogenization* in the acquired profiles, as more items are shared across profiles. This is also why the system is able to regain some diversity by using randomized suggestions, promoting their diversity, or by personalizing them.

6.3 Answers to the Questionnaire

The results of the questionnaire are shown in Table 3. We compare the approaches by calculating the mean of the answers to each question. Due to the low sample size, the MANOVA test was unable to confirm that the difference in the responses are significant. Nevertheless, we can state that all configurations are rated positively by participants for all constructs. The highest score is consistently achieved by the Popularity and Random Popularity strategies. Specifically, the Popularity configuration was rated the highest in terms of Use Intentions, Overall Satisfaction, Recommendation Accuracy, Perceived Usefulness, and Control.

The answers to the Recommendation Accuracy and Perceived Usefulness constructs show a picture that is different from the one described by the metrics. Indeed, despite the fact that Highest Prediction obtained the best results for both Hit Rate and nDCG, participants rated it lower than most other approaches. Of course, this highlights the discrepancy between the relevance of the recommendations, and their *usefulness* as perceived by users, which is already discussed in the literature [14,42,29]. We propose to investigate this phenomenon further: if it is validated statistically, it could mean that personalized suggestions favor relevance over the usefulness of the items.

Participants rated positively both Popularity and Random Popularity configurations for the Ease of Use construct. This corroborates the hypothesis that the use of popular items makes profile elicitation easier. On the contrary, the $\sqrt{Popularity} \times Variance$ and Item-to-Item configurations scored lower for this construct, which can be attributed to the fact that they tend to suggest movies that are uncommon or controversial.

Construct	No sugg.	Pop.	Rand Pop.	Pop.Var.	I2I	Highest Pred.
Transparency	4.03	4.00	**4.22**	3.83	3.70	3.71
Use Intentions	3.88	**3.92**	3.73	3.63	3.65	3.53
Overall Satisfaction	4.10	**4.25**	4.24	3.97	3.97	3.82
Recommend. Accuracy	3.83	**4.17**	3.95	3.57	3.70	3.62
Perceived Usefulness	3.65	**4.06**	3.84	3.54	3.51	3.35
Ease of Use	4.10	4.22	**4.30**	4.16	4.07	4.19
Control	3.70	**3.97**	3.84	3.80	3.62	3.75
Interaction Adequacy	3.96	3.98	**4.01**	3.94	3.95	3.98

Table 3 Answers to the questionnaire, organized by construct. The highest scores are highlighted in bold.

All configurations score similarly in the Interaction Adequacy construct, which means that the participants generally agree that the dialogue model employed by the CoRS is adequate for reaching their recommendation goals. This suggests that the slight increase in interaction time and conversation length does not negatively impact user experience.

7 Discussion

Based on the experimental results, we can now answer the research questions described in Section 5.3:

RQ1. What is the effect of Active Learning techniques on the recommendation quality of a CoRS? How does it compare against a model based only on user-given ratings?

The results described in Table 2 show that incorporating user profile elicitation strategies based on Active Learning into a CoRS can have a positive effect on the quality of the recommendations. Indeed, the statistical tests confirmed that the Highest Predicted strategy significantly improves both Hit Rate and nDCG, compared to a strategy based only on user-given feedback. On the other hand, the Popularity and Random Popularity approaches tend to score better in terms of the perceived quality, usefulness, and trust in the recommended items. However, these results are not statistically significant, and will require further studies in order to be fully understood.

Contrary to our expectations, the introduction of system suggestions did not result in increased diversity of the recommended movies. In fact, most approaches involved in this study rely on proposing popular or familiar items, which are more likely to be shared across profiles. This in turn increases the probability that the same movie will be recommended to multiple users. Despite this, we show that a decrease in overall diversity is not accompanied by a loss in recommendation quality.

These results help us better understand how people interact with a CoRS using natural language. Since users prefer interacting using short, direct messages, the problem of obtaining an informative user profile with few interactions becomes a fundamental factor for the success of a CoRS. The user-driven strategy employed in the baseline allows users to tailor the profile to their specific preferences, which explains the high diversity. However, users may be inclined to only rate a set of closely related items, thus narrowing down their profile into a narrow view of the entire item space. This results in *overspecialized* recommendations, as the outputs of the recommender are also related to the ones mentioned in said profile. This limits their usefulness, as shown in the results in Tables 2 and 3. The item selection strategies investigated in this study help reduce the risk of narrow profiles, which allows the PPR algorithm to find more suitable connections between movies. The choice of the specific approach is also important: we discovered that personalized Active Learning strategies can improve the relevance of the recommendations, however this does not directly translate into a higher perceived value of the recommendations. Therefore, this choice must be made carefully.

RQ2. What is the effect of Active Learning techniques on the interaction cost of a CoRS? How does it compare against a model based only on user-given ratings?

The statistical tests failed to find any significant improvements to the interaction cost over the baseline. On the contrary, they show that Active Learning strategies suffered a slight increase in conversation length, as users were required to write more messages on average before completing the experiment. However, it can be argued that measuring the effects of user profile acquisition strategies on interaction cost is hard, even more so if the measurement is done in a real world scenario.

Moreover, the results suggest that system suggestions do not help users extract more complex preferences, who instead limited themselves to one item per message. Indeed, the fact that users prefer writing short queries is known, has been discussed in similar areas of research such as Web search [51]. Therefore, more research is needed in order to understand how to elicit complex text feedback from users.

Nevertheless, the results show that there are clear differences in the way each Active Learning strategy affects user experience. The configurations that rely on popularity tend to decrease the time per question and the overall time needed to complete the experiment. On the other hand, the strategies that prioritize diversity tend to make the preference elicitation step harder. The opinions recorded from the questionnaire also seem to suggest that users consider the Popularity and Random Popularity strategies as slightly easier to use than the others. Indeed, popular movies are easier to rate, because users are more familiar with them. Therefore, suggesting items to rate based on their popularity can be a simple yet effective strategy for acquiring preferences more easily. However, a more comprehensive study must be performed in order to fully validate this hypothesis.

8 Conclusion

In this paper, we presented an investigation on the application of several item selection strategies based on Active Learning into a Conversational Recommender System, and evaluated them in an experiment. We compared each strategy against a completely user-driven baseline. The experiment was performed in a realistic scenario, in which actual users interacted with a working recommender system, and evaluated the recommendations. We measured the effects on both recommendation quality and interaction cost. The experiments prove that the integration of item suggestion strategies based on Active Learning has a positive effect on the recommendation quality of the system, despite a sensible decrease in overall diversity of the recommendations. Moreover, we discovered that while personalizing the items suggested during profile elicitation increases the likelihood of finding satisfactory recommendations, focusing on popular items tends to improve the perceived quality of the recommendations.

A limitation of this experimental study is the relatively small sample size compared to the number of configurations, which reduced the significance of the results. Additionally, the Bonferroni correction applied to the t-test results further reduced the amount of significant differences due to its highly conservative behavior [10]. As future work, we plan on extending this study by increasing the number of participants. This will help increase the significance of the results, and will give more confidence to the observations made in this paper. We will further investigate the new hypotheses that emerged from this experiment. First, we propose to perform an experiment focused on measuring the quality of interaction of the preference

elicitation step. We will continue studying the trade-off between the accuracy of recommendations and their usefulness as reported by users. In order to widen the range of applicability of our findings, we will replicate the study using other recommendation algorithms. Finally, we propose to investigate aspect-based active learning techniques, which can elicit judgements on aspects of the items instead of items themselves, in order to improve the profile acquisition process.

Declarations

– **Funding** No funding was received for conducting this study.
– **Conflict of Interests** The authors declare that they have no conflict of interests.

References

1. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Trans. Knowl. Data Eng. **24**(5), 896–911 (2012). DOI 10.1109/TKDE.2011.15
2. Basile, P., Musto, C., de Gemmis, M., Lops, P., Narducci, F., Semeraro, G.: Content-based recommender systems+ DBpedia knowledge= semantics-aware recommender systems. In: Semantic Web Evaluation Challenge, pp. 163–169. Springer (2014)
3. Bertomeu Castelló, N.: Finding Optimal Presentation Sequences for a Conversational Recommender System. In: S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo, R.R. Yager (eds.) Advances in Computational Intelligence, vol. 300, pp. 328–337. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). DOI 10.1007/978-3-642-31724-8_34. URL http://link.springer.com/10.1007/978-3-642-31724-8_34. Series Title: Communications in Computer and Information Science
4. Brill, T.M., Munoz, L., Miller, R.J.: Siri, Alexa, and other digital assistants: a study of customer satisfaction with artificial intelligence applications. Journal of Marketing Management **35**(15-16), 1401–1436 (2019). DOI 10.1080/0267257X.2019.1687571. URL https://www.tandfonline.com/doi/full/10.1080/0267257X.2019.1687571
5. Burke, R.D., Hammond, K.J., Yound, B.: The findme approach to assisted browsing. IEEE Expert **12**(4), 32–40 (1997)
6. Carenini, G., Smith, J., Poole, D.: Towards more conversational and collaborative recommender systems. In: Proceedings of the 8th international conference on Intelligent user interfaces, pp. 12–18 (2003)
7. Catherine, R., Cohen, W.: Transnets: Learning to transform for recommendation. In: Proceedings of the eleventh ACM conference on recommender systems, pp. 288–296 (2017)
8. Christakopoulou, K., Radlinski, F., Hofmann, K.: Towards Conversational Recommender Systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, pp. 815–824. ACM Press, San Francisco, California, USA (2016). DOI 10.1145/2939672.2939746. URL http://dl.acm.org/citation.cfm?doid=2939672.2939746
9. Chuan, C.H., Morgan, S.: Creating and evaluating chatbots as eligibility assistants for clinical trials: An active deep learning approach towards user-centered classification. ACM Transactions on Computing for Healthcare **2**(1), 1–19 (2020)
10. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7**, 1–30 (2006). Publisher: JMLR. org
11. Elahi, M., Ricci, F., Rubens, N.: Active learning strategies for rating elicitation in collaborative filtering: A system-wide perspective. ACM Transactions on Intelligent Systems and Technology (TIST) **5**(1), 1–33 (2014). Publisher: ACM New York, NY, USA
12. Elahi, M., Ricci, F., Rubens, N.: A survey of active learning in collaborative filtering recommender systems. Computer Science Review **20**, 29–50 (2016). Publisher: Elsevier
13. Følstad, A., Brandtzaeg, P.B.: Users' experiences with chatbots: findings from a questionnaire study. Quality and User Experience **5**(1), 1–14 (2020)

14. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: Proceedings of the fourth ACM conference on Recommender systems - RecSys '10, p. 257. ACM Press, Barcelona, Spain (2010). DOI 10.1145/1864708.1864761. URL http://portal.acm.org/citation.cfm?doid=1864708.1864761
15. Gini, C.: Meausrement of inequality and incomes. The Economic Journal **31**, 124–126 (1921)
16. Glass, J., Polifroni, J., Seneff, S., Zue, V.: Data collection and performance evaluation of spoken dialogue systems: The MIT experience. In: Sixth International Conference on Spoken Language Processing (2000)
17. Goker, M., Thompson, C.: The adaptive place advisor: A conversational recommendation system. In: Proceedings of the 8th German Workshop on Case Based Reasoning, pp. 187–198. Citeseer (2000)
18. Golbandi, N., Koren, Y., Lempel, R.: On bootstrapping recommender systems. In: Proceedings of the 19th ACM international conference on Information and knowledge management, pp. 1805–1808 (2010)
19. Golbandi, N., Koren, Y., Lempel, R.: Adaptive bootstrapping of recommender systems using decision trees. In: Proceedings of the fourth ACM international conference on Web search and data mining - WSDM '11, p. 595. ACM Press, Hong Kong, China (2011). DOI 10.1145/1935826.1935910. URL http://portal.acm.org/citation.cfm?doid=1935826.1935910
20. Greco, C., Suglia, A., Basile, P., Semeraro, G.: Converse-et-impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems. In: Conference of the Italian Association for Artificial Intelligence, pp. 372–386. Springer (2017)
21. Hammond, K.J., Burke, R.D., Lytinen, S.L.: A case-based approach to knowledge navigation. In: IJCAI, pp. 2071–2072 (1995)
22. Haveliwala, T.H.: Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. IEEE transactions on knowledge and data engineering **15**(4), 784–796 (2003). Publisher: IEEE
23. Hernández-Rubio, M., Bellogín, A., Cantador, I.: Aspect-based active learning for user preference elicitation in recommender systems. In: I. Cantador, M. Chevalier, M. Melucci, J. Mothe (eds.) Proceedings of the Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020), Samatan, Gers, France, July 6-9, 2020, *CEUR Workshop Proceedings*, vol. 2621. CEUR-WS.org (2020). URL http://ceur-ws.org/Vol-2621/CIRCLE20_16.pdf
24. Iovine, A., Lops, P., Narducci, F., de Gemmis, M., Semeraro, G.: Improving preference elicitation in a conversational recommender system with active learning strategies. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing, pp. 1375–1382 (2021)
25. Iovine, A., Narducci, F., Semeraro, G.: Conversational Recommender Systems and natural language:: A study through the ConveRSE framework. Decision Support Systems p. 113250 (2020). DOI 10.1016/j.dss.2020.113250. URL http://www.sciencedirect.com/science/article/pii/S0167923620300051
26. Jannach, D., Kreutler, G.: Rapid development of knowledge-based conversational recommender applications with advisor suite. J. Web Eng. **6**(2), 165–192 (2007)
27. Jannach, D., Manzoor, A., Cai, W., Chen, L.: A Survey on Conversational Recommender Systems. arXiv preprint arXiv:2004.00646 (2020)
28. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS) **20**(4), 422–446 (2002). Publisher: ACM New York, NY, USA
29. Kaminskas, M., Bridge, D.: Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. ACM Transactions on Interactive Intelligent Systems **7**(1), 1–42 (2017). DOI 10.1145/2926720. URL https://dl.acm.org/doi/10.1145/2926720
30. Kang, J., Condiff, K., Chang, S., Konstan, J.A., Terveen, L., Harper, F.M.: Understanding How People Use Natural Language to Ask for Recommendations. In: Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17, pp. 229–237. ACM Press, Como, Italy (2017). DOI 10.1145/3109859.3109873. URL http://dl.acm.org/citation.cfm?doid=3109859.3109873

31. Kiseleva, J., Williams, K., Jiang, J., Hassan Awadallah, A., Crook, A.C., Zitouni, I., Anastasakos, T.: Understanding user satisfaction with intelligent assistants. In: Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval, pp. 121–130 (2016)
32. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 426–434 (2008)
33. Lee, S.L.: Commodity recommendations of retail business based on decisiontree induction. Expert Systems with Applications **37**(5), 3685–3694 (2010). Publisher: Elsevier
34. Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. In: Proceedings of the 20th ACM conference on Hypertext and hypermedia - HT '09, p. 73. ACM Press, Torino, Italy (2009). DOI 10.1145/1557914.1557930. URL `http://portal.acm.org/citation.cfm?doid=1557914.1557930`
35. Merialdo, A.K.B.: Improving collaborative filtering for new-users by smart object selection (2001)
36. Musto, C., Lops, P., de Gemmis, M., Semeraro, G.: Context-aware graph-based recommendations exploiting personalized pagerank. Knowledge-Based Systems **216**, 106806 (2021)
37. Narducci, F., de Gemmis, M., Lops, P., Semeraro, G.: Improving the user experience with a conversational recommender system. In: International Conference of the Italian Association for Artificial Intelligence, pp. 528–538. Springer (2018)
38. Narducci, F., de Gemmis, M., Lops, P., Semeraro, G.: Improving the user experience with a conversational recommender system. In: C. Ghidini, B. Magnini, A. Passerini, P. Traverso (eds.) AI*IA 2018 - Advances in Artificial Intelligence - XVIIth International Conference of the Italian Association for Artificial Intelligence, Trento, Italy, November 20-23, 2018, Proceedings, *Lecture Notes in Computer Science*, vol. 11298, pp. 528–538. Springer (2018)
39. Nickel, M., Rosasco, L., Poggio, T.A., et al.: Holographic embeddings of knowledge graphs. In: The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), pp. 1955–1961 (2016)
40. Parapar, J., Radlinski, F.: Diverse user preference elicitation with multi-armed bandits. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 130–138 (2021)
41. Priyogi, B.: Preference Elicitation Strategy for Conversational Recommender System. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 824–825 (2019)
42. Pu, P., Chen, L., Hu, R.: A user-centric evaluation framework for recommender systems. In: Proceedings of the fifth ACM conference on Recommender systems - RecSys '11, p. 157. ACM Press, Chicago, Illinois, USA (2011). DOI 10.1145/2043932.2043962. URL `http://dl.acm.org/citation.cfm?doid=2043932.2043962`
43. Rashid, A.M., Albert, I., Cosley, D., Lam, S.K., McNee, S.M., Konstan, J.A., Riedl, J.: Getting to know you: learning new user preferences in recommender systems. In: Proceedings of the 7th international conference on Intelligent user interfaces, pp. 127–134 (2002)
44. Rashid, A.M., Karypis, G., Riedl, J.: Learning preferences of new users in recommender systems: an information theoretic approach. Acm Sigkdd Explorations Newsletter **10**(2), 90–100 (2008). Publisher: ACM New York, NY, USA
45. Real, R., Vargas, J.M.: The probabilistic basis of Jaccard's index of similarity. Systematic biology **45**(3), 380–385 (1996). Publisher: JSTOR
46. Rubens, N., Sugiyama, M.: Influence-based collaborative active learning. In: Proceedings of the 2007 ACM conference on Recommender systems, pp. 145–148 (2007)
47. Schröder, G., Thiele, M., Lehner, W.: Setting goals and choosing metrics for recommender system evaluations. In: UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA, vol. 23, p. 53 (2011)
48. Sun, Y., Zhang, Y.: Conversational Recommender System. arXiv:1806.03277 [cs] (2018). URL `http://arxiv.org/abs/1806.03277`. ArXiv: 1806.03277
49. Tong, S.: Active learning: theory and applications, vol. 1. Stanford University USA (2001)
50. Wang, R., Ma, X., Jiang, C., Ye, Y., Zhang, Y.: Heterogeneous information network-based music recommendation system in mobile networks. Computer Communications **150**, 429–437 (2020)
51. Weld, H., Huang, X., Long, S., Poon, J., Han, S.C.: A survey of joint intent detection and slot-filling models in natural language understanding. arXiv preprint arXiv:2101.08091 (2021)

52. Williams, J., Raux, A., Henderson, M.: The Dialog State Tracking Challenge Series: A Review. Dialogue & Discourse **7**(3), 4–33 (2016). URL `http://dad.uni-bielefeld.de/index.php/dad/article/view/3685`
53. Xu, K., Yang, J., Xu, J., Gao, S., Guo, J., Wen, J.R.: Adapting user preference to online feedback in multi-round conversational recommendation. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 364–372 (2021)