Integrating a cognitive assistant within a critique-based recommender system

Marc Güell[a], Maria Salamó[a], David Contreras[b], Ludovico Boratto[c]

[a] *Dept. Matemàtiques i Informàtica, Facultat de Matemàtiques i Informàtica, UBICS research institute, Universitat de Barcelona, Barcelona, Spain*

[b] *Facultad de Ingeniería y arquitectura, Universidad Arturo Prat, Iquique, Chile*

[c] *Data Science and Big Data Analytics, EURECAT, Centre Tecnològic de Catalunya, Barcelona, Spain*

Abstract

Recommender systems are cognitive computing systems designed to support humans in their decision-making processes through convincing, timely product suggestions. In the field of recommender systems, critique-based recommenders have been widely applied as an effective approach for guiding users through a product space in pursuit of suitable products. To date, no critique-based approach has included an assistant that support users in their search in a pleasant way. In this paper, we describe how we integrate an assistant within a critique-based recommender. We consider the proposed assistant to be cognitive because its reasoning process when recommending products is based on a cognitively-inspired clustering algorithm. The proposal is evaluated by users and compared with a non-assistant approach. The results of this research demonstrate that the integration of a cognitive assistant within the recommender improves the user experience and increases the performance of the recommendation process, i.e., users need fewer cycles to achieve the desired product or service.

Keywords: Recommender System, Cognitive Assistant, Cognitive Systems

---

1. Introduction

Cognitive computing refers to a hardware or software solution that mimics in some way human intelligence capabilities and helps to improve human decision-making (Kelly III, 2016). Indeed, decision-making is a human activity based mainly on cognitive information. Nowadays, users are searching for products and services among a large volume of content more than ever before. This very large volume and variety of products and their specific and varied characteristics make it difficult for a user to find the proper product or service.

Recommender systems (Ricci et al., 2011) are cognitive computing systems that assist users in making the best possible decisions in a timely manner. In the literature, recommender systems use many and varied recommendation techniques, from collaborative filtering (Koren and Bell, 2011; Elahi et al., 2014) to content-based techniques (Pazzani and Billsus, 2007), among others. However, in high product domains where the products are very expensive and in which users are likely to search for and buy products for the first time, the task of navigating in order to locate a desired item among a large set of options is intimidating for the average users. In these domains, critique-based recommenders (Pu et al., 2011b; Chen and Pu, 2012; Salamó and Escalera, 2012) are widely recognized as effective preference-based search and recommendation technology. The conversational nature of critique-based recommenders helps them to guide users through a product space in pursuit of suitable products. This is done by using a cyclical recommendation process (i.e., a dialogue), alternatively making suggestions and eliciting user feedback, in order to refine their needs and preferences, based on recent recommendations. Users provide feedback by critiquing features of the currently recommended product. A critique is a directional preference over one of the features, for example, "*like this but cheaper*". Note that in this way, users are not obliged to formulate formal queries but rather express their preferences in an easier and more natural way. Critiquing is based on the idea that it is easier for a user to critique a product recommendation, rather than to construct formal queries (Burke et al., 1997). Consequently, critique-based recommenders are able to help customers with ill-defined preferences both to navigate through complex[1] product spaces and to better understand their own buying preferences.

Most of these recommenders, however, fail to enable users to interact with the recommender in a more pleasant way. In this article, we propose the integration of a cognitive assistant within the critique-based recommender.

[1]In complex product spaces, users require a good knowledge of the large number of characteristics of the products and their relationship with the different available options in order to make a decision.

This cognitive assistant captures the preferences of the users in natural language and helps them in the search process. The assistant's behavior when helping users is based on a cognitively-inspired clustering algorithm (Contreras and Salamó, 2018). The objective is to identify clusters of similar products based on their description, learn and reason based on the user's interactions and adapt these clusters in order to use them as a mechanism for proposing search directions to the user with the aim of both guiding the user in the search space and learning their preferences more rapidly. Note that clusters are adapted according to users' interactions, which reflect their evolving requirements. Thanks to the integration of the assistant within the critique-based recommender, the recommender is able to focus appropriately on the products the user is interested in and offer quality product recommendations that might otherwise be ignored, thereby, making smarter recommendations and reducing session lengths. We carried out the evaluation with real users and compared it with a non-assistant approach. We evaluate both the recommendation efficiency and the usability perceived by the users. This evaluation produced adequate results concerning usability and indicated an improvement in the efficiency of the recommender compared to a non-assistant approach.

In this paper, our hypothesis is that a cognitive assistant can improve both the recommendation process and the definition of the user model in critique-based recommenders. With this hypothesis in mind, the contribution of this paper is three-fold:

1. We propose an assistant whose rationale is based on a cognitive user model.
2. We integrate the assistant within the critique-based recommendation process, and the unified framework is detailed in depth: the conceptual architecture, the recommendation process with the integrated assistant, and the process of the cognitive assistant.
3. We carry out an exhaustive evaluation with real users to validate our initial hypothesis. Our evaluation focuses on a comparison of the framework with the non-assistant approach versus the assistant-based approach. With these experiments we show the positive influence of integrating an assistant within a recommendation framework. The results of our in-depth evaluation confirm our hypothesis and indicate that

our proposal improves on the efficiency[2] of previous non-assistant approaches employed in critique-based recommenders.

The rest of this paper is structured as follows: Section 2 presents related work. Section 3 presents the recommendation environment that integrates the cognitive assistant within a critique-based recommender. Section 4 presents the live-user evaluation. Section 5 concludes the paper.

2. Related Work

In this section, we analyze the main approaches that integrate assistants and recommender systems. Although both the assistants[3] and the recommender systems are based on various machine learning strategies, there is a large technological gap between them, as described by Rafailidis and Manolopoulos (2019). There are many studies of assistants and recommender systems, but very few approaches have focused on integrating them to form a unified framework that also considers users' interactions with the recommender system via conversations.

Recently, there has been a growing interest in employing conversational systems in the recommender systems community (Christakopoulou et al., 2016), as they have a major impact on human-computer interaction. Conversational recommender systems (Zhang et al., 2018; Sun and Zhang, 2018) aim to find and recommend the most relevant information (e.g., products or services) for users based on textual or spoken dialogues. Research in this area can be divided into two main categories.

The first category corresponds to the research such as Zhang et al. (2019) that investigates the core algorithms used for recommendation generation, where the recommender is in charge of asking users about their interests. In this approach, users can communicate with the system, answering questions in natural language conversations an approach that has proven to be more effective than classical recommendation (Ricci et al., 2011) approaches. However, users are unable to freely specify their own and evolving preferences.

The second category corresponds to those conversational recommenders that use an assistant to help users to express their preferences. There are

[2]Analyzed by means of the Average Session Length, which measures the number of cycles that a user must go through before being presented with their ideal target product.

[3]An assistant can be any kind of conversational interface, such as a virtual assistant or a chatbot.

many and varied applications of assistants in the literature but not all of them use a recommender system. For example, Massai et al. (2019) proposed a semantic assisting engine for suggesting a list of local POI's and services but this approach retrieves the list by exploiting the Km4City smart city knowledge base. Costa et al. (2016) proposed the inclusion of a persuasive module based on a case-based argumentation approach, in the iGenda cognitive assistant built upon a multi-agent system. iGenda helps care-receivers and caregivers in the management of their activities in daily life by resolving scheduling conflicts and promoting active aging activities.

In the second category, there is also research on web-based virtual assistants that use a recommender system to improve the assistant's performance. In this vein, Sobecki et al. (2006) defined a cooking assistant, that recommends recipes, based on a hybrid recommendation algorithm. Tavčar et al. (2016) described a web service for virtual museum tours that is based on a virtual assistant that uses a simple content-based recommender to provide suggestions regarding exhibits targeted for a specific user.

When considering the integration of assistants within the recommender system, very few studies have focused on exploiting assistants as a recommendation interface. Note that most of the proposals in this vein use a chatbot[4]. Atzori et al. (2017) presented the *Tourisitific* project, in which the aim is to create a recommender system for travel information that supports lightweight access through a chatbot, but the paper only focused on the interaction between the recommender and the chatbot and it lacks a demonstrator. The proposal does not contain details about the architecture and the recommendation process or how effective this integration is. Nica et al. (2018) presented a chatbot for e-tourism that uses model-based reasoning for enhancing the user experience during the chat.

Also in area of integrating assistants within recommender systems, there are some approaches that have considered the use of a conversational recommender system. Kucherbaev et al. (2017) outlined their vision of chatbots that facilitate interaction between citizens and policy-makers at the city scale. Lee et al. (2018) presented a web-based conversation application that provides personalized travel recommendations. This is one of the few approaches to present an overall architecture for developing a recommender

[4]Chatbots are automated programs used as a medium to interact with humans via textual or auditory means.

system interacting with a dialogue system. Narducci et al. (2018) studied user interaction with a content-based recommender system (i.,e, based on PageRank algorithm) in a movie recommendation scenario, implemented as a Telegram chatbot. The authors evaluated different interaction modes. The results of the user study demonstrated that when users can type their preferences, the recommender shows the best trade-off between accuracy and cost of interaction.

To this point, most of the studies regarding the integration of assistants within recommender systems have been limited to the description of the interaction between them. Differing from these approaches, the research discussed in this article takes into consideration the description of the unified framework as well as the interaction aspects of the framework, just to keep the gap noted by Rafailidis and Manolopoulos (2019). In addition, note that none of the previous approaches regarding the integration of assistants within a recommender system can be considered to be cognitive, since the assistant is considered to be a mere information acquisition tool. The proposal in this paper describes a cognitive assistant based on a cognitively-inspired clustering algorithm (Contreras and Salamó, 2018). Moreover, to the best of our knowledge, no cognitive assistant within a critiquing-based recommender system has ever been proposed previously.

3. Description of the recommendation environment with an integrated assistant

This section describes some required definitions, the conceptual architecture of the recommender system integrated with a cognitive assistant, the recommendation process, and the process followed by the cognitive assistant when providing recommendations and assisting the user in their search.

3.1. Required definitions

We first define what a critique is in our recommendation framework. In addition, we define all of the data structures involved in both the cognitive assistant and the cognitive recommender: the case base, the session base, the cognitive user model, the clustering model and the concept base.

In critique-based recommenders, the most common feedback mechanisms are *unit* and *compound* critiques. In the former, users are allowed to critique a single feature of a product at a time (McCarthy et al., 2006; Ricci and Nguyen, 2007), whereas in compound critiques, each critique can be a combination of multiple unit critiques (Reilly et al., 2007; McGinty and Reilly, 2011; Pu et al., 2011b). In the literature, most of critique-based recommenders use unit critiques.

Definition 1. Critique. A critique c_i, is represented as a triple $(f_i, type, v)$, where f_i refers to a feature of the recommended product, $type$ is the type of critique c_i (i.e., typically $<$, $>$, $<>$), and v is the current value of f_i.

For example, "a cheaper smartphone", when the price of the current recommendation is \$500, implies a critique $(price, <, \$500)$. Another example is $(manufacturer, <>, Lg)$, which represents the user critique "I do not like Lg smartphones".

Definition 2. Case Base. The case base, CB, is a set of products for recommendation, described as $CB = \{p_1, ..., p_n\}$, where p_i is the ith product and the set of features that describes each product is defined as $F = \{f_1, ..., f_m\}$. In addition, each product is assigned to one cluster which is the range of 1 and k clusters of the data set, $cl_{\{1:k\}}$. Specifically, we use the k-prototype (Huang, 1998) method for partitioning the case base.

Definition 3. Session Base. The session base, SB, is a data set of past critiquing sessions by other users defined as $SB = \{s_1, ..., s_l\}$, where s_i is a sequence of recommendation-critique pairs (r_i, c_i), where r_i is the recommendation and c_i is a critique. A recommendation r_i is a product that has been recommended to the user during the session s_i and has not been accepted because the user has made another critique c_i in order to receive a new recommendation. Note that each session culminates in a terminal product, denoted by $term(s_i)$, which is the accepted recommendation in session s_i.

The cognitive recommender stores a history of past sessions in the session base, SB, and maintains the current recommendation session of the user in the cognitive user model, $CogUM$.

Definition 4. Cognitive User Model. The cognitive user model, $CogUM$, is a set of recommendation-critique pairs, defined as $CogUM = \{u_1, ..., u_k\}$, where $u_i = (r_i, c_i, CM_i)$ is a particular recommendation cycle with r_i representing the recommended product, c_i representing the critique applied to r_i and CM_i is formally described by Definition 5.

Definition 5. Clustering Model. Each CM_i is the clustering model in the ith recommendation cycle, which is defined as $CM_i = \{cm_1, cm_2, ..., cm_k\}$, where cm_j represents the number of cases in a particular cluster j.

Note that CM_i stores the information relating to all clusters in a particular cycle. Moreover, the clustering model is updated during each recommendation cycle. In the first cycle, the case base contains the complete data set and during the recommendation cycles the products in each cluster are maintained if they satisfy the latest critique.

Definition 6. Concept Base. The concept base, CoB, is a data set of keywords defined as $CoB = \{K_1, ..., K_t\}$, where K_i is a tuple that contains a keyword that can represent a feature (e.g., price, manufacturer, etc.) or a type of feature (e.g., more, less, etc.) and the different ways of describing this keyword in natural language.

The concept base is used for interpreting the user's input and relating it to one or more critiques. Moreover, we also use it to decide what question to ask users when they request help. The concept base contains a set of words related to each one of the features or the type of the features that may exist in the domain. Moreover, it can be modified according to the different languages used by the assistant. At the moment our framework considers the English and Spanish languages.

In the English version for the smartphone domain, a few examples of the contents of the concept base are given below:

K_1: $< Price, [price, monetary, cost, money, damage] >$

K_2: $< Manufacturer, [manufacturer, maker, company] >$

K_3: $< Plus, [plus, more, best, better] >$

K_4: $< Less, [less, lowe, small, little, lower] >$

It is important to remark that one role for the CoB is to suggest questions that intrinsically contain a critique in order to shorten the search space (a role detailed later in Section 3.4.2). To this end, the concept base is sorted out according to the keywords. For example, in the smartphone domain, the order is: price, camera, size, RAM, weight, manufacturer, operating system, and CPU, with price being the most important feature in this domain and the CPU the least important one. This particular order was defined based on a previous live-user study in a virtual world (Contreras et al., 2018) and we have ordered it running from the features that the users critique most to the least critiqued ones.

3.2. Conceptual Architecture

This section describes the conceptual architecture of the proposal. In particular, Figure1 illustrates the client-server conceptual architecture, which consists of three main components: the **Client**, **Server**, and **Recommender** layers.

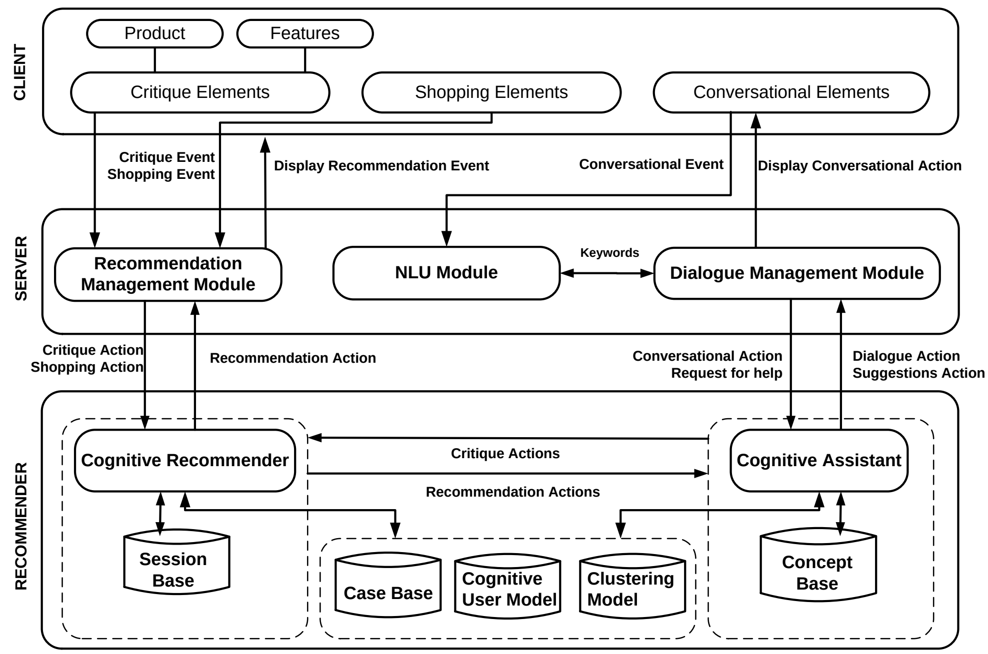

Figure 1: Conceptual Architecture of the client-server proposal with three components: the Client, Server and Recommender layers

First, the **Client** is mainly devoted to being the viewing layer. It is in charge of capturing the users' interactions with the assistant or the recommender and displaying the recommendation products (i.e., the products and their technical features). In this layer, users can interact with the recommender by means of *Critique Elements*, *Shopping Elements*, and *Conversational Elements*. Critique Elements allow users to perform critiques, Shopping Elements let users either start or finish (activate, buy, or abandon) a recommendation process, and Conversational Elements allow users to interact in natural language with the assistant via a text chat. The user's interactions in the Client layer trigger three types of events that are sent to the Server layer. These events are: the CritiqueEvent, the ShoppingEvent, and the ConversationalEvent. The CritiqueEvent occurs when the user performs a critique by pushing the critique elements. The ShoppingEvent is activated when users either start or finish a recommendation process. On the other hand, the ConversationalEvent is activated when the user performs a conversational interaction using natural language on a Conversational Element. The Client layer also receives two actions from the Server layer: the DisplayRecommendationAction, which is in charge of displaying a new product recommendation, and the DisplayConversationalAction, which displays the answers given by the cognitive assistant from the recommender layer in a text chat.

Second, the **Server** layer is responsible for the communication between the Client and the Recommender layer. Basically, it has three components. The first component is the *Recommendation Management Module*, which is responsible for mapping critique events to critique actions when users interact with critique elements and, in reverse, recommendation actions to display recommendation actions. The *Natural Language Understanding* (NLU) module is in charge of identifying keywords from the text chat in the Conversational Event. The keywords serve to identify whether the user needs help or if she has expressed a critique in natural language. These keywords are sent to the *Dialogue Management Module*, which allows for the interaction with the Cognitive Assistant in the Recommender layer. Note that the dialogue management module identifies two different actions: a Conversational Action and a Request for help. The Conversational Action indicates that the user has made one or more critiques. For example, in the smartphone domain, if the current product recommendation has 3GB of RAM and the user writes the phrase *I would like more memory*, it means a preference over the feature "RAM" that is converted into the following critique: $RAM > 3$. The conversational Action goes back to the client as a Dialogue Action and also generates a recommendation action. That is, a new product is shown to the user considering their current preferences. On the other hand, the request for help indicates that the user asks the assistant for help. This action goes back to the client as a Suggestion Action. A suggestion is a critique that is built from the information stored in the Cognitive User Model and the Clustering Model during the session and which is presented to the user in the form of a question. With the answer to this question, a new conversational action will be generated and, as a result, the Cognitive User Model, the clustering model, will be updated, and a new recommendation will be shown to the user.

Finally, the **Recommender** layer is composed by two modules: the *Cognitive Recommender* and the *Cognitive Assistant*. It is important to note that both the recommender and the assistant make use of: (1) a case base

(CB) of products for recommendation, where each product is assigned to one cluster in the first recommendation cycle based on their similarity with other products; (2) a cognitive user preference model ($CogUM$) that stores all the critiques made by the user during the session, which learns from interactions with the user and adapts its content to the user's evolving requirements during the session; (3) a clustering model (CM) that stores the number of cases in each cluster every time the user performs a critique and is therefore updated during each recommendation cycle.

In particular, the *Cognitive Recommender* extends the Incremental Critiquing (IC)[5] algorithm and incorporates a cognitively-inspired clustering algorithm in the recommendation process (Contreras and Salamó, 2018). Note that this algorithm, called HGR-CUM-I, adds an adaptive clustering process to a critique-based recommender, thereby adapting the recommendation process with a new cognitive user model based on both the preferences received by the user and the adaptive clusters. On the other hand, the *Cognitive Assistant* builds a complex dialog thanks to the inclusion of the IBM Watson Assistant[6] service and the syntax analyzer of the Google Natural Language[7]. The *Cognitive Assistant* is responsible for mapping text to critique actions by using a concept base (see Definition 6) that contains a set of keywords associated to critique terms. When one or more keywords are identified, they are sent as critiques to the *Cognitive Recommender* with the aim of obtaining a new recommendation product (see the *Critique* and *Recommendation* actions between the recommender and the assistant in the Recommender layer). However, the cognitive assistant is also in charge of focusing the user when she requests help, by analyzing the Cognitive User Model and the Clustering Model it prepares a critique and transforms the critique into a question. This question is sent to the user, who may accept or decline it. When the user responds to the question, a new Conversational Action begins with the critique hidden in the question if the response is positive, or an opposite critique is sent if the response is negative.

[5]The IC is one of the most well-known critiquing-based recommenders (Reilly et al., 2005).

[6]Details at https://www.ibm.com/cloud/watson-assistant/

[7]For details, look at https://cloud.google.com/natural-language/

3.3. Recommendation process

Users change between different states during their interactions with the recommender and the cognitive assistant. These states and their interactions can be defined in a finite-state-machine (see Figure 2).

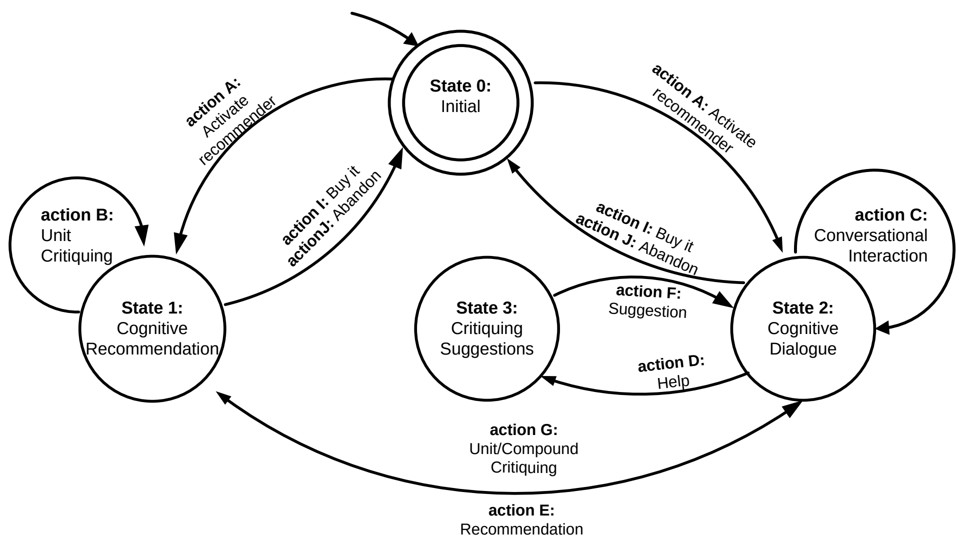

Figure 2: Users' states during their interactions with the recommender and the cognitive assistant

The set of states and their transactions are detailed below.

State 0: Initial. The user is able to interact with the cognitive recommender and the cognitive assistant. The user can then start a recommendation session (i.e., the *activation* action) by using the Shopping Element in the Cognitive Recommender or the Conversational Element in the Cognitive Assistant **(action A)**. When the user starts the session, the cognitive recommender creates a cognitive user model for the user and the recommendation process is moved to **State 1** (Cognitive Recommendation state) or **State 2** (Cognitive Dialogue).

State 1: Cognitive Recommendation. In this state, the recommender returns a product to be shown in the client layer. The recommendation is based on: the current recommended product, the current critique, the cognitive user model, the relevant sessions that are similar to the current cognitive user model, and the clustering model in the current recommendation cycle. Specifically, we apply our cognitively-inspired clustering algorithm

HGR-CUM-I as described in Contreras and Salamó (2018). We have chosen it because the HGR-CUM-I algorithm enhances significantly the recommendation efficiency of well-known state-of-the-art algorithms: IC (Reilly et al., 2005) and HGR (Salem et al., 2014).

The user can perform four actions at this state.

1. *Unit critiquing* **(action B)**, consists in making unit critiques through the Critique Elements. For each critique, the recommender provides a new product recommendation and the cognitive user model is updated to include the new critique and the new recommendation. Moreover, the number of products for each cluster in the clustering model is updated in the current recommendation cycle. This action maintains the user in the same state (State 1).
2. *Buy it* **(action I)** occurs when the user has found a suitable product, and it is activated by means of a Shopping Element, as depicted in Figure 1.
3. *Abandon* **(action J)**, which is activated by means of a Shopping Element. This action happens when the user explicitly finishes the recommendation process because she does not find a suitable product. After action I and J, the user returns to **State 0** (Initial), where she can start a new recommendation.

State 2: Cognitive Dialogue. A user changes to this state when she interacts with the cognitive assistant. In this state, the cognitive assistant uses the NLU module to analyze the user's text chat from the user and identifies the keywords that allow it to guide the dialogue between the user and the assistant. The user can perform four actions in this state.

1. *Conversational interaction* **(action C)**, consists of a conversational process between the user and the assistant by means of Conversational Elements. During the dialogue, the cognitive assistant guides the user in searching a suitable product. This action maintains the user in the same state (State 2).
2. *Help* **(action D)**, which is activated when a user writes *help* in the text chat. This action occurs when a user wants a suggestion from the cognitive assistant. After this action, the user changes to **State 3** (Critiquing Suggestions state).
3. *Unit/Compound Critiquing* **(action G)**, occurs when the cognitive assistant identifies one or more critiques in the user interaction with the

text chat. Both unit and compound critiques are sent to the cognitive recommender and the user changes to **State 1** (Cognitive Recommendation).

4. *Buy it* **(action I)** and *Abandon* **(action J)** have been previously described in **State 1**.

State 3: Critiquing Suggestions. A user changes to this state in two situation. First, when she needs a suggestion during the cognitive dialogue and she actives action D. In this situation, the cognitive assistant suggests to the user a critique from the data stored in the cognitive user model and the clustering models, and asks the user if the inferred critique describes a feature of interest or not. Second, when the cognitive assistant receives a new recommendation from the cognitive recommender **(action E, Recommendation)**.

3.4. Cognitive assistant process

This subsection is divided into two parts. First of all, we briefly describe how the assistant processes the user's input, how the assistant collects the users' input, and how it generates "unit" or "compound" critiques. Secondly, we describe the methodology used to provide recommendations and to ask cognitive questions based on the user's interaction.

3.4.1. Processing user's input

The process of the cognitive assistant is based on IBM Watson and Google Natural Language services, as mentioned in Section 3.2. The Watson Assistant service makes it possible to guide the conversation with users using a flow of nodes and child nodes. In particular, a node is an inflection point in the dialogue that contains a condition and according to how this condition is satisfied, three different actions may occur: (1) it generates a result (i.e., a critique or a new question for the user if she has requested help), (2) it moves to another node that may be a child node, or (3) it jumps to any point in the dialogue (i.e., that is, it goes to any other node, not necessarily the next one in the dialog). The cognitive assistant proposed contains a dialog consisting of 15 nodes, in which each node includes a title, a condition for entry into the node, one or more answers for the user and the node action. In addition to the initial and the last nodes used to fix the beginning and the end of the dialog, the remaining ones are devoted to processing the different types of critiques made by the user (i.e., the numerical critiques: greater than or lower than a feature value, and the nominal ones) and to provide information and a question for the user when she requests help.

Every node analyzes the text introduced by the user in the assistant and decides which action should be performed. Google Natural Language is used by the *Natural Language Understanding* module to detect key concepts of interest for the cognitive assistant. Specifically, we used the syntax analyzer, in which we extracted the dependencies between lemmas. For example, it is important for the assistant to know that the lemma "less" is related to the lemma "RAM" and not to another product feature. NLU passes the relationship between lemmas as keywords to the *Dialog Management Module* as keywords (see Figure 1) and the latter sends a *Conversational Action* including the related keywords to the cognitive assistant. The assistant then builds a critique, if possible, based on the keywords. To this end, the cognitive assistant has a concept base that contains several keywords associated to each critique scope (see Def. 6). The scope of a critique is the feature name and its type and value (see Def. 1). For example, the concept base contains product specifications such as the price and the name of the manufacturer, as well as the different ways used to refer to these specifications, and it also stores the different terms used to describe each type of critique, such as more, plus, best, less, and others. It is important to remark that users may describe several critiques in one text and the cognitive assistant is therefore enabled to detect more than one critique in a unique user's interaction. Note that when the cognitive assistant receives a *Conversational Action*, the dialogue is moved to a specific node in the IBM Watson and a request for the cognitive recommender is activated with the critiques built by the assistant and a new message is sent to user.

3.4.2. Providing recommendations and asking questions to the user

The cognitive assistant is in charge of providing recommendations and asking users questions, see Figure 3. It provides recommendations when the assistant receives a conversational action and when it detects one or more critiques into the text typed by the user. It asks the user a question when it receives a request for help (i.e., the user writes *help* in the text chat). We will now explain both situations in detail.

In the former, the assistant receives a conversational action (see Figure 3). The result of this action is that the assistant shows a new recommendation that is generated by the cognitive recommender, which is based on the HGR-CUM-I algorithm (Contreras and Salamó, 2018). It combines CUM-I (Contr-

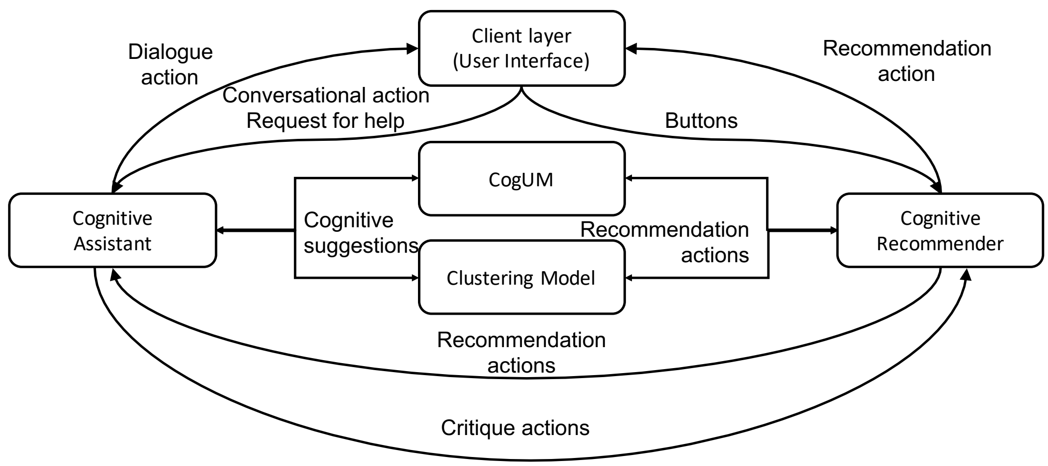

Figure 3: Cognitive assistant process for providing recommendation and asking questions to the user

eras and Salamó, 2018) with a well-known history-based recommender named HGR (Salem et al., 2014). To the best of our knowledge, HGR-CUM-I is the algorithm that improves recommendation efficiency the most when compared to IC (Reilly et al., 2005) and HGR (Salem et al., 2014), both of which are well-known state-of-the-art algorithms. In particular, HGR-CUM-I stores a history of past sessions in the session base, SB, and maintains the current user recommendation session in the cognitive user model, $CogUM$ (see Def. 4). Specifically, the recommendation process consists of four phases: (1) HGR-CUM-I identifies the set of relevant products[8] and updates the number of products for each cluster in the CM_j cluster model, based on the set of relevant products in the recommendation session, j; (2) the recommender ranks the candidate products using the quality measure defined in Contreras and Salamó (2018). Specifically, the quality measure evaluates how the current critique influences the cluster model (CM_j) in the cognitive user model ($CogUM$); (3) the recommender identifies a set of relevant sessions based on overlapping computation, which are history sessions in the SB that overlap with the user's current partial critique session, stored in $CogUM$. If there are relevant sessions, the recommender ranks candidates (i.e., the accepted recommendation in the session) for the next recommendation based on the number of sessions in which the candidate has accepted a recommendation

[8]Relevant products is a subset of the case base where each product satisfies the last critique.

and the candidate products are stored in a list r_F. If there are no relevant sessions the recommender reverts to the CUM-I algorithm (Contreras and Salamó, 2018), which proposes an alternative list of products r_F; (4) finally, the best ranked product, p_r, which is obtained using r_F, is recommended to the user. The full information corresponding to the recommended product is obtained from the case base, CB, and the recommender product p_r is temporally removed from the CB. The product information then is sent to the client layer as a *Recommendation Action* (see Figure 1).

In the second situation, the cognitive assistant receives a request for help, as shown in Figure 3. The assistant processes and suggests a question to the user, which is based on an inferred critique, $c_s = (f_i, type_i, v_i)$, built from the data stored in the cognitive user model, $CogUM$, and the clustering model, CM. Figure 1 also shows the relationship between the cognitive assistant and these models.

The process for building a question is as follows. First, the cognitive assistant selects the feature for critiquing, f_i. From the $CogUM$, the assistant learns which features have been critiqued and which ones have not yet been used by the user during the session. It assigns the highest score to the non-critiqued features in $CogUM$. The score is based on the order defined in the concept base, discarding those features that have been critiqued previously during the session. With the user's answer to a question that includes a feature not previously critiqued by her, both the assistant and the recommender learn about the user's preferences regarding features that have not been specified because the $CogUM$ improves its content and it allows the recommendation process to focus on relevant cases while ignoring irrelevant data. Second, apart from choosing a feature to suggest a question, the assistant needs to suggest a value and a type (lower, greater or different). Moreover, taking into account the fact that the next recommendation should be close to the current recommended product, we use the cluster of the current recommended product for selecting both the value v_i and the type of suggested critique, $type_i$. To this end, the assistant again uses the $CogUM$ (i.e., in particular the CM stored in it) by considering the cluster to be the current recommended product in the $CogUM$. If the selected feature is numerical, we calculate the average of the values for products in the cluster of the current recommended product (cl_{avg}) and use it as the value of the suggested critique, v_i. Next, for selecting the type of critique ($<$ or $>$), we compute the number of products in the cluster, using CM_i, with both higher v_h and lower v_l values than the average, cl_{avg}. With the aim of reducing the product space, we apply the following conditions: if $v_h >= v_l$ then $type_i$ is $<$ else $type_i$ is $>$. On the other hand, if the selected feature is nominal we compute the most frequent value and it is used as v_i, while for nominal features the $type_i$ is always $<>$. Finally, the cognitive assistant converts the critique into a more pleasant expression in natural language and asks the user the critique in the form of a question regarding whether she is interested in a product with such a characteristic. If the answer is 'yes', the cognitive assistant sends the critique to the cognitive recommender and the process is the same as the one described in the first situation. If the answer is 'no', the cognitive assistant asks the user to describe a preference or seek further help. Note that the assistant is cognitive because it uses information stored in the cognitive recommender, which is updated at each cycle of the recommender, according to the user's preferences stored in $CogUM$ and CM_i.

4. Live-user evaluation

In this section, the efficiency of the integrated assistant within the critique-based recommender is assessed and compared to a non-assistant approach. In addition, we will evaluate the usability of the proposal. Before detailing the evaluation, we will first describe the interface used in our experiments.

4.1. Interface of the recommendation environment

Figure 4 depicts the interface screen shown to the user in their browser. It corresponds to the client side depicted in the conceptual architecture shown in Figure 1. The example interface shown is set in the smartphone domain. It is divided into two main areas, each one with a specific interaction modality (i.e., buttons or typing) and functionality.

On the left hand side of Figure 4 the image of the current product recommendation is shown and, under it, one of the targets the user must reach in our live user study is shown. The target is only shown to the specific live-user evaluation we have performed in this paper.

The first area, just in the middle of Figure 4, represents the critique elements in the conceptual architecture shown in Figure 1. This area is devoted to *button interaction*. Here, a product is described in terms of its features and the particular value of each feature. Additionally, each of the features contains one or two buttons for performing critiques (i.e., these are the cri-

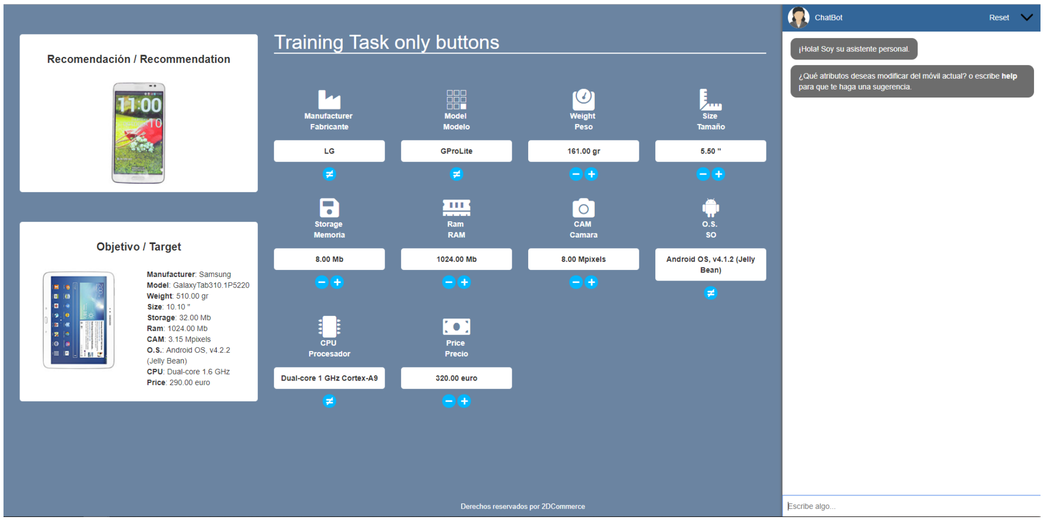

Figure 4: The main interface with a smartphone view

tique elements in Figure 1). The user is able to make a critique[9], which gives her an opportunity to provide informative feedback. The user feedback represents the *CritiqueAction* in the Space Server layer (see Figure 1). This feedback is introduced to the cognitive user model (*CogUM*). Next, the cognitive recommender uses this informative feedback about the user's preferences and responds to this action by replacing the product displayed with a new recommendation that better matches the preference expressed (i.e., the *RecommendationAction* that maps to the *DisplayRecommendationEvent* in the Space Server layer described in Section 3).

The second area in the interface is for the *assistant*, which is displayed on the right hand side of Figure 4. This area displays the conversation between the user and the assistant and a text box to let the user express her needs and preferences in natural language. Note that the user is free to use the buttons area or the assistant area at any time.

4.2. Setup and methodology

A summative or quality assurance test, which is usually performed during the middle or near-end stages of development, was used. It focuses on

[9]With a critique the user expresses a preference regarding a specific feature in line with their personal requirements (e.g., *cheaper* or *higher star rating for hotel*, etc.).

gathering both qualitative and quantitative data (Bowman et al., 2002). We developed a wizard-like online web application called *CritiqueAssistant*[10], which contains all of the instructions, interfaces and questionnaires that participants need to perform the evaluation remotely.

The online evaluation procedure consisted of the following phases. First, a *Pre-test* phase during which participants were asked to input their background information and their previous experience with recommender systems. This was followed by a *Training* phase during which participants performed a training task consisting of finding a predefined target product by using buttons (i.e., to make critiques) or typing on the chat (to specify their preferences and request help). The predefined target product was randomly selected from the case base. In particular, the features of the target product are shown to the user on the lower left corner of the *CritiqueAssistant*. Following the training, users started the *Efficiency Test* phase in which users performed three tasks featuring a predefined target product with the aim of evaluating the recommendation efficiency of the different interaction modalities. The tasks performed are as follows:

1. Users search for the predefined target product using only the buttons that enable them to make critiques. They are not allowed to use the assistant.
2. Users search for the predefined target product using only the assistant and the buttons were disabled.
3. Users search for the predefined target using the complete interface and they freely decide to use the buttons or the type what they want in the assistant.

We evaluated the different tasks by computing the *Average Session Length* (from now on ASL), which measures the number of cycles that a user must work through before being presented with their ideal target product. In addition, we calculate the *Percentage of Benefit* measured as:

$$\text{Benefit}(x, y) = (1 - \frac{y}{x}) \cdot 100 \tag{1}$$

[10]The application is available at http://131.72.237.141:8083/WebRec/signin.htm and a video featuring the most important parts of our interface is available at https://youtu.be/JRuFRXku4Qw

where y and x stand for the ASL of the compared task and the baseline, respectively. In this paper, the baseline is Task 1. It is important to remark that the ASL metric has been widely used in evaluations of critique-based recommenders, such as McCarthy et al. (2005); Zhang et al. (2008); Salamó and Escalera (2012); Salem et al. (2014); Contreras et al. (2018); Contreras and Salamó (2018, 2019, 2020), among others.

Finally, in the last phase of our evaluation, participants answered a questionnaire that is shown in Table 1. The questionnaire, which is based on previous works (Ricci and Nguyen, 2007; Pu et al., 2011a), consisted of 19 questions using a seven-point Likert scale (i.e., in which 1 corresponded to "strongly disagree" and 7 to "strongly agree") and a last question to see which part of the interface participants preferred (i.e., the buttons or the assistant). After the test, we collected data from logs and questionnaires. We then analyzed these data to extract relevant information concerning test objectives.

Forty test subjects, comprising both male and female participants ranging in age from 22 to 32 and having varied levels of computer skills and experience in assistants environments, were recruited. The test was conducted by a moderator and an observer, and it was performed using a SMARTPHONE[11] data set that contained 1721 products with two types of features: numeric (Weight, Size, Storage, Ram, Camera, and Price) and nominal (Manufacturer, Model, OS, and CPU). In addition, we adopted the methodology used in Salem and Hong (2013); Salem et al. (2014) to automatically generate the largest session base, SB, featuring 10000 past critiquing sessions based on the behavior of rational users. It is important to remark that previous works demonstrate that a large session base improves the recommendation efficiency (Salem and Hong, 2013; Salem et al., 2014; Contreras and Salamó, 2019, 2020).

The experiment was designed as a within-subject test, i.e., the same group of participants was used for all three tasks. These tasks were rotated to mitigate carryover effects. Participants were divided into two groups. One group of 20 participants that performed Task 1, Task 2, and Task 3 and another group of 20 participants who performed Task 2, Task 1, and Task 3.

[11]This data set was obtained from the gsmarena website (https://www.gsmarena.com) and it is available on demand. Moreover, this data set has been used in previous works (Contreras et al., 2014, 2018; Contreras and Salamó, 2019, 2020).

4.3. Recommendation Efficiency

Figure 5 shows the distribution by the number of cycles performed to complete a task, represented vertically, and the tasks performed by users, horizontally. The box plot highlights that the shortest number of cycles, in these experiments is achieved by the approach that let users to use both the buttons and the assistant (i.e., the Task 3). In fact, for Task 1, the median of cycles was 17 and the lower and upper quartiles of cycles ranged from 13 to 21. For Task 2, the median of cycles was 12.50 and the lower and upper quartiles of cycles ranged from 11 to 16; 1 outlier of 27 cycles. Finally, the median for Task 3 was 11 cycles and the lower and upper quartiles of cycles ranged from 9 to 14; 1 outlier of 22 cycles. In addition, the red line in Figure 5 shows the average of number of cycles (ASL) for the three tasks, with an ASL of 17.05 for Task 1 and 13.83 for Task 2, while the lowest value corresponds to Task 3 with 11.94 cycles.

In our experiments, we also computed the average time used by the participants to complete the tasks (see the green line in Figure 5). Participants used less time (51.27 seconds) with the interface based on buttons (Task 1) and the longest time (101.58 seconds) with the interface based exclusively on the assistant, which is to be expected since participants needed time to type their needs and the buttons are a faster interaction mechanism. In Task 2, the users spent much longer (152.80 seconds) to complete the task but the benefit (defined in zEquation 1) is still close to 20% in comparison with Task 1. That is, users spent more time defining their preferences but their sessions featured fewer recommendation cycles, as shown in Figure 5. Our direct observation of the users during the experiments suggests that they need to take a certain amount of time to type in their preferences and this effort made them consider carefully which preference/s they should write about. On the other hand, Task 3 corresponds to the interface that integrates both buttons and the assistant and, as expected, the time achieved lies in between that of Task 1 and Task 2, while achieving the highest benefit (30%).

We applied the ANOVA statistical method and the Bonferroni test (Bland and Altman, 1995) to analyze whether the differences in efficiency recorded between Task 1, Task 2, and Task 3 are statistically significant. In particular, we obtained a *p-value* of 0.015 between Task 1 and Task 2, a *p-value* of $0.49E{-}4$ between Task 1 and Task 3, and a *p-value* of 0.122 between Task 2 and Task 3. These *p-values* mean that there are significant differences in the efficiency when we use the cognitive assistant (i.e., Task 2 and Task 3) compared to the non-assistant approach (Task 1), but not between the tasks that use the assistant (i.e., between Tasks 2 and 3).

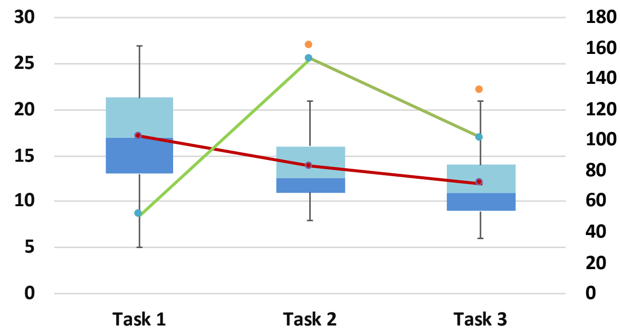

Figure 5: Efficiency obtained at each task

In Task 3, we also analyze the percentage of use of the different interaction modalities, as shown in Figure 6. The results were very similar in percentage terms. Button use by the participants in Task 3 was 52% and 48% of their interactions were carried out by typing their preferences or soliciting help from the assistant.

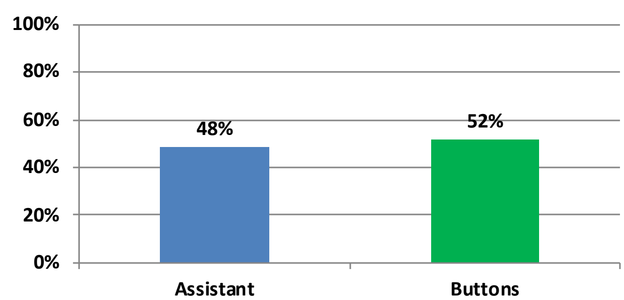

Figure 6: Percentage of use of the different interaction modalities

Finally, we evaluated the number of interactions corresponding to the different interaction modes featured in the cognitive recommender. Our results, which are shown in Figure 7, indicate that participants carried out unit critiques in 45% of the interactions, typed multiple critiques (i.e., compound) in 36% of the interactions, and requested help in order to receive suggestions in 20%.

4.4. Analysis of usability

To analyze the usability of the new proposed framework, we focused on analyzing the following dimensions (see Figure 8): Quality, Utility, Inter-

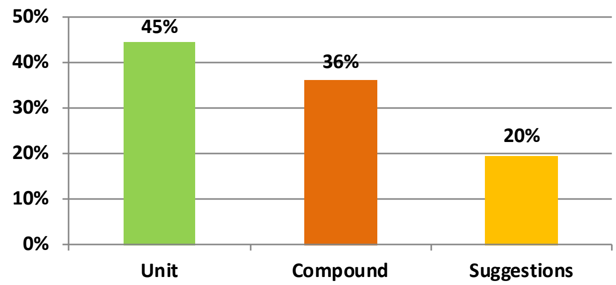

Figure 7: Percentage of use of the different interaction modes

action, Interface, Perceived Ease of User, and Intention of Use. First, the *Quality* dimension is the degree to which users feel the suggestions made by the assistant match their interests and preferences. Second, the *Utility* of the assistant is estimated by comparing the framework in Task 3 (which integrates a cognitive assistant) with the framework in Task 1 (an environment that lacks a cognitive assistant). Third, *Interaction* and *Interface* dimensions refer to their degree of satisfaction with the interface and its interactive elements. Fourth, the *Perceived ease of use* refers to how easy it is for the user to learn how to use the buttons and the assistant. Finally, *Intention of use* refers to attitudes regarding the use of the recommender system with an assistant in the future.

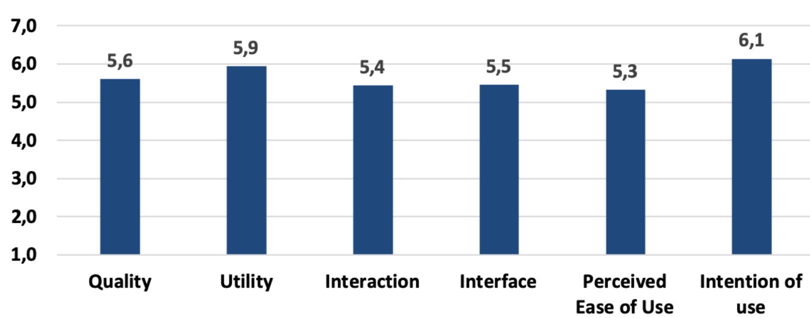

Figure 8: Average dimensions score value for the post-test questionnaire in a 7-point likert scale.

Participants perceive a high level of assistant accuracy with an average of 5.6 (see the Quality dimension in Figure 4.4) and an average of 5.9 in the Utility dimension. The results in these two dimensions indicate that the users' perception is that the new framework provides accurate support to them during the recommendation process.

In general, participants found the user interface to be easy to use, with an average score of 5.3 in the Perceived Ease of Use dimension and 5.5 in the Interface dimension. Moreover, participants rank highly the Interaction with the assistant (i.e., the average value obtained across all participants is 5.4). The good results on these three dimensions indicate a high level of usability for the assistant. Finally, it is important to highlight that participants evaluate with the highest score the behavioural intention of use in the future (depicted as Intention of Use in Figure 8) with an average of 6.1 in a seven point likert scale.

4.5. Discussion

Given the analysis performed with real users, we can now return to our initial purpose in this paper and determine the effect of integrating an assistant within a critique-based recommender. Moreover, we will discuss whether any lesson can be learned and applied to the constructions of new approaches in the future.

In our experiments, we analyzed a critique-based recommender that only uses buttons, one that collects user's preferences by means of an assistant, and a conversational recommender that integrates both the buttons and the assistant. Based on our results, which are shown in Figure 5, we can conclude that: (1) the shortest ASL is obtained by integrating both the recommender system and the assistant, and (2) users perceive the integration of a cognitive assistant in a conversational recommendation framework as being of high quality and utility. The lesson learned is that users need to be involved in the recommendation process and they need to express their preferences in an easy-to-use interaction mechanism based on natural language, as shown in the usability analysis. Moreover, our conclusion is that a conversational recommender that is able to capture user needs expressed in natural language and involve her cognitively in the decision-making process is able to both guide the user appropriately in the search space and to learn their preferences faster.

5. Conclusions

This paper, which is based on integrating an assistant within a critique-based recommender, defines the conceptual architecture and details the processes of both the cognitive assistant and the cognitive recommender. Both elements of the unified framework are cognitive because their rationale is based on a cognitively-inspired clustering algorithm that evolves according to the users' interactions. This integration enables richer user interaction models and more elaborate recommender systems become possible. Moreover, new user interaction models make it possible to process various types of user input. As a result, the integration of an assistant allows the recommender to make smarter recommendations and to reduce the session lengths. The proposal was evaluated with real users and compared with a non-assistant approach. The results show that there is a significant benefit of 29.9% in efficiency when integrating a cognitive assistant within a cognitive critique-based recommender system in comparison to the non-assistant approach.

Acknowledgments

This research was partially funded by project 2017-SGR-341, MISMIS-LANGUAGE (grant No. PGC2018-096212-B-C33) from the Spanish Ministry of Science and Innovation, NanoMoocs (grant No. COMRDI18-1-0010) from ACCIO, and by project DIAMOND funded by the European Union's Horizon 2020 research and innovation program (grant No. 824326).

References

Atzori, M., Boratto, L., Spano, L.D., 2017. Towards Chatbots as Recommendation Interfaces, in: Proceedings of the 2nd. Workshop on Engineering Computer-Human Interaction in Recommender Systems, pp. 26–31. URL: `http://ceur-ws.org/Vol-1945/paper_5.pdf`.

Bland, J.M., Altman, D.G., 1995. Multiple significance tests: the Bonferroni method. Journal of BMJ 310, 170. doi:10.1136/bmj.310.6973.170.

Bowman, D.A., Gabbard, J.L., Hix, D., 2002. A Survey of Usability Evaluation in Virtual Environments : Classification and Comparison of Methods. Journal of Presence: Teleoperators and Virtual Environments 11, 404–424. doi:10.1162/105474602760204309.

Burke, R.D., Hammond, K.J., Young, B.C., 1997. The FindMe Approach to Assisted Browsing. Journal of IEEE Expert: Intelligent Systems and Their Applications 12, 32–40. doi:10.1109/64.608186.

Chen, L., Pu, P., 2012. Critiquing-based recommenders: survey and emerging trends. Journal of User Modeling and User-Adapted Interaction 22, 125–150. doi:10.1007/s11257-011-9108-6.

Christakopoulou, K., Radlinski, F., Hofmann, K., 2016. Towards Conversational Recommender Systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, USA. pp. 815–824. doi:10.1145/2939672.2939746.

Contreras, D., Salamó, M., 2018. A Cognitively Inspired Clustering Approach for Critique-Based Recommenders. Journal of Cognitive Computation 12, 428–441. doi:10.1007/s12559-018-9586-5.

Contreras, D., Salamó, M., 2019. History and Opinion Collaborative Recommender: An Approach for Social Recommendations, in: Artificial Intelligence Research and Development. IOS Press. Frontiers in Artificial Intelligence and Applications, pp. 285–294. doi:10.3233/FAIA190135.

Contreras, D., Salamó, M., 2020. Data-driven decision making in critique-based recommenders: from a critique to social media data. Journal of Intelligence Information Systems 54, 23–44. doi:10.1007/s10844-018-0520-9.

Contreras, D., Salamó, M., Rodríguez, I., Puig, A., 2014. An Approach to Improve User Experience with Conversational Recommenders through a 3D Virtual Environment, in: Proceedings of the XV International Conference on Human Computer Interaction, Association for Computing Machinery. pp. 403–406. doi:10.1145/2662253.2662319.

Contreras, D., Salamó, M., Rodríguez, I., Puig, A., 2018. Shopping decisions made in a virtual world: Defining a state-based model of collaborative and conversational user-recommender interactions. Journal of IEEE Consumer Electronics Magazine 7, 26–35. doi:10.1109/MCE.2017.2728819.

Costa, A., Heras, S., Palanca, J., Novais, P., Julián, V., 2016. A persuasive cognitive assistant system, in: Ambient Intelligence- Software and Applications – 7th Int. Symposium on Ambient Intelligence, Springer. pp. 151–160. doi:10.1007/978-3-319-40114-0_17.

Elahi, M., Ricci, F., Rubens, N., 2014. Active learning strategies for rating elicitation in collaborative filtering: A system-wide perspective. Journal of ACM Transactions on Intelligent Systems and Technology 5, 13:1–13:33. doi:10.1145/2542182.2542195.

Huang, Z., 1998. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Journal of Data Mining and Knowledge Discovery 2, 283–304. doi:10.1023/A:1009769707641.

Kelly III, J.E., 2016. Computing, Cognition and the Future of Knowing, How Humans and Machines are Forging a New Age of Understanding, in: Computing Research News, Computing Research Association. pp. 8–9.

Koren, Y., Bell, R., 2011. Advances in Collaborative Filtering, in: Recommender Systems Handbook. Springer, pp. 145–186. doi:10.1007/978-0-387-85820-3_5.

Kucherbaev, P., Psyllidis, A., Bozzon, A., 2017. Chatbots As Conversational Recommender Systems in Urban Contexts, in: Proceedings of the International Workshop on Recommender Systems for Citizens, ACM, USA. pp. 6:1–6:2. doi:10.1145/3127325.3127331.

Lee, S., Moore, R., Ren, G.J., Arar, R., Jiang, S., 2018. Making Personalized Recommendation through Conversation: Architecture Design and Recommendation Methods, in: Proceedings of Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, pp. 727–730. URL: https://aaai.org/ocs/index.php/WS/AAAIW18/paper/view/17221.

Massai, L., Nesi, P., Pantaleo, G., 2019. PAVAL: A location-aware virtual personal assistant for retrieving geolocated points of interest and location-based services. Journal of Engineering Applications of Artificial Intelligence 77, 70 – 85. doi:10.1016/j.engappai.2018.09.013.

McCarthy, K., Reilly, J., McGinty, L., Smyth, B., 2005. Experiments in Dynamic Critiquing, in: Proceedings of the 10th International Conference on Intelligent User Interfaces, Association for Computing Machinery. pp. 175–182. doi:10.1145/1040830.1040871.

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P., 2006. Group Recommender Systems: A Critiquing Based Approach, in: Proceedings of the 11th International Conference on Intelligent User Interfaces, Association for Computing Machinery. pp. 267–269. doi:10.1145/1111449.1111506.

McGinty, L., Reilly, J., 2011. On the evolution of critiquing recommenders, in: Recommender Systems Handbook. Springer, pp. 419–453. doi:10.1007/978-0-387-85820-3_13.

Narducci, F., de Gemmis, M., Lops, P., Semeraro, G., 2018. Improving the User Experience with a Conversational Recommender System, in: Proceedings of AI*IA 2018 – Advances in Artificial Intelligence, Springer. pp. 528–538. doi:10.1007/978-3-030-03840-3_39.

Nica, I., Tazl, O.A., Wotawa, F., 2018. Chatbot-based Tourist Recommendations Using Model-based Reasoning, in: Proceedings of the 20th International Configuration Workshop, pp. 25–30. URL: http://ceur-ws.org/Vol-2220/05_CONFWS18_paper_31.pdf.

Pazzani, M., Billsus, D., 2007. Content-based recommendation systems, in: The Adaptive Web: Methods and Strategies of Web Personalization, pp. 325–341. doi:10.1007/978-3-540-72079-9_10.

Pu, P., Chen, L., Hu, R., 2011a. A User-centric Evaluation Framework for Recommender Systems, in: Proceedings of the Fifth ACM Conference on Recommender Systems, Association for Computing Machinery. pp. 157–164. doi:10.1145/2043932.2043962.

Pu, P., Faltings, B., Chen, L., Zhang, J., Viappiani, P., 2011b. Usability Guidelines for Product Recommenders Based on Example Critiquing Research, in: Recommender Systems Handbook. Springer, pp. 511–545. doi:10.1007/978-0-387-85820-3_16.

Rafailidis, D., Manolopoulos, Y., 2019. Can virtual assistants produce recommendations?, in: Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics, Association for Computing Machinery, New York, USA. pp. 1–6. doi:10.1145/3326467.3326468.

Reilly, J., McCarthy, K., McGinty, L., Smyth, B., 2005. Incremental critiquing. Journal of Knowledge-Based Systems 18, 143–151. doi:10.1007/1-84628-102-4_8.

Reilly, J., Zhang, J., McGinty, L., Pu, P., Smyth, B., 2007. A Comparison of Two Compound Critiquing Systems, in: Proceedings of the 12th International Conference on Intelligent User Interfaces, Association for Computing Machinery. pp. 317–320. doi:10.1145/1216295.1216356.

Ricci, F., Nguyen, Q.N., 2007. Acquiring and Revising Preferences in a Critique-Based Mobile Recommender System. Journal of IEEE Intelligent Systems 22, 22–29. doi:10.1109/MIS.2007.43.

Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), 2011. Recommender Systems Handbook. Springer. URL: https://link.springer.com/book/10.1007/978-0-387-85820-3.

Salamó, M., Escalera, S., 2012. Increasing retrieval quality in conversational recommenders. Journal of IEEE Transactions on Knowledge and Data Engineering 24, 1876–1888. doi:10.1109/TKDE.2011.116.

Salem, Y., Hong, J., 2013. History-aware critiquing-based conversational recommendation, in: Proceedings of the 22nd International Conference on World Wide Web Companion, Association for Computing Machinery. pp. 63–64. doi:10.1145/2487788.2487813.

Salem, Y., Hong, J., Liu, W., 2014. History-guided Conversational Recommendation, in: Proceedings of the 23rd International Conference on World Wide Web, Association for Computing Machinery. pp. 999–1004. doi:10.1145/2567948.2578844.

Sobecki, J., Babiak, E., Slanina, M., 2006. Application of Hybrid Recommendation in Web-based Cooking Assistant, in: Proceedings of the 10th International Conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part III, Springer. pp. 797–804. doi:10.1007/11893011_101.

Sun, Y., Zhang, Y., 2018. Conversational recommender system, in: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Association for Computing Machinery, USA. pp. 235–244. doi:10.1145/3209978.3210002.

Tavčar, A., Csaba, A., Butila, E.V., 2016. Recommender system for virtual assistant supported museum tours. Journal of Informatica 40, 279–284. URL: http://www.informatica.si/index.php/informatica/article/view/1433.

Zhang, J., Jones, N., Pu, P., 2008. A Visual Interface for Critiquing-Based Recommender Systems, in: Proceedings of the 9th ACM Conference on

Electronic Commerce, Association for Computing Machinery. p. 230–239. doi:10.1145/1386790.1386827.

Zhang, X., Xie, H., Li, H., Lui, J.C.S., 2019. Toward building conversational recommender systems: A contextual bandit approach. Computing Research Repository (CoRR) abs/1906.01219. URL: http://arxiv.org/abs/1906.01219.

Zhang, Y., Chen, X., Ai, Q., Yang, L., Croft, W.B., 2018. Towards Conversational Search and Recommendation: System Ask, User Respond, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, USA. pp. 177–186. doi:10.1145/3269206.3271776.

Table 1: Post-test questionnaire

Dimension	Statement
Quality: Accuracy	I liked the items recommended by the system.
Quality: Accuracy	Each of the recommended products was relevant to me.
Quality: Enjoyability	I enjoyed the items recommended to me.
Quality: Accuracy	The items recommended to me matched my interests.
Quality:Novelty	The recommender system helps me discover new products.
Utility:Perceived Usefulness	The recommended items effectively helped me find the ideal product.
Utility: Attitudes	Overall, I'm satisfied with the assistant.
Interaction Adequacy	The recommender provides an adequate way for me to express my preferences.
Interaction Adequacy	The assistant suggestions help me to decide my preferences.
Interface Adequacy	The recommender interface provides sufficient information.
Interface Adequacy	The labels used by the recommender interface are clear and adequate.
Interface Adequacy	The assistant provides me with valuable information.
Perceived Ease of use: initial learning	I became familiar with the recommender system very quickly.
Perceived Ease of use	I found it easy to tell the system about my preferences.
Perceived Ease of use	I found it easy to tell the system about my preferences with the buttons.
Perceived Ease of use	I found it easy to tell the system about my preferences with the assistant.
Perceived Ease of use: Ease of decision making	Finding an item to buy(target) with the help of the recommender is easy.
Behavioural intention of use: Continuance and frequency	I would use this recommender system for buying products in the future.
Behavioural intention of use	If a recommender such as this existed, I would use it to find products to buy.
Interface preferences	What has been more useful for you to express your preferences? 1. buttons, 2. the assistant