MyrrorBot: A Digital Assistant Based on Holistic User Models for Personalized Access to Online Services

CATALDO MUSTO, Università degli Studi di Bari, Italy
FEDELUCIO NARDUCCI, Politecnico di Bari, Italy
MARCO POLIGNANO, MARCO DE GEMMIS, PASQUALE LOPS, and GIOVANNI SEMERARO, Università degli Studi di Bari, Italy

In this article, we present MyrrorBot, a *personal digital assistant* implementing a natural language interface that allows the users to: (i) access online services, such as *music, video, news,* and *food recommendation*s, in a *personalized* way, by exploiting a strategy for implicit user modeling called *holistic user profiling*; (ii) query their own user models, to inspect the features encoded in their profiles and to increase their awareness of the personalization process.

Basically, the system allows the users to formulate *natural language* requests related to their information needs. Such needs are roughly classified in two groups: *quantified self-related* needs (e.g., *Did I sleep enough? Am I extrovert?*) and *personalized access* to online services (e.g., *Play a song I like*). The *intent recognition* strategy implemented in the platform automatically identifies the intent expressed by the user and forwards the request to specific services and modules that generate an appropriate answer that fulfills the query.

In the experimental evaluation, we evaluated both *qualitative* (users' acceptance of the system, usability) as well as *quantitative* (time required to complete basic tasks, effectiveness of the personalization strategy) aspects of the system, and the results showed that MyrrorBot can improve the way people access online services and applications. This leads to a more effective interaction and paves the way for further development of our system.

CCS Concepts: • **Information systems** → **Recommender systems**; **Personalization**; • **Computing methodologies** → **Natural language processing**; • **Human-centered computing** → **User models;**

Additional Key Words and Phrases: Chatbots, user models, personalization, recommender systems, personal digital assistants

ACM Reference format:
Cataldo Musto, Fedelucio Narducci, Marco Polignano, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2021. MyrrorBot: A Digital Assistant Based on Holistic User Models for Personalized Access to Online Services. *ACM Trans. Inf. Syst.* 39, 4, Article 46 (August 2021), 34 pages.
https://doi.org/10.1145/3447679

Authors' addresses: C. Musto, M. Polignano, M. de Gemmis, P. Lops, and G. Semeraro, University of Bari Aldo Moro, Department of Computer Science, Via E.Orabona 4, 70125, Bari, Italy; emails: {cataldo.musto, marco.polignano, marco.degemmis, pasquale.lops, giovanni.semeraro}@uniba.it; F. Narducci, Dipartimento di Ingegneria Elettrica e dell'Informazione, Via E.Orabona 4, 70125, Bari, Italy; email: fedelucio.narducci@poliba.it.

1046-8188/2021/08-ART46 $15.00
https://doi.org/10.1145/3447679

1 INTRODUCTION

The rise in popularity of **Personal Digital Assistants (PDAs)** is one of the most important trends we are currently witnessing. This tendency, which is mainly due to the advancements in natural language processing and speech recognition techniques [Deng et al. 2013], is also confirmed by the huge investments made by big companies in that direction.[1] Examples of such systems are Amazon's Alexa, Apple's Siri, Google's Assistant, and Microsoft's Cortana. The main characteristic of these systems is the capability to interpret requests expressed in natural language and to generate an answer that fulfills users' information needs [Sarikaya 2015].

At first sight, PDAs show an *open-domain knowledge* (e.g., they can provide services and information about music, movies, weather, jokes, games, recipes, news, and so on). However, this is a very challenging capability [Huang et al. 2020], which is typically obtained just at an *integration* level. Indeed, many different domain-specific modules, each of which is specialized in a particular goal, are typically combined in a single system [Wang et al. 2014]. In terms of system architecture, PDAs represent a sort of meta-layer of *intelligence* that acts as an hub for apps and external services. In a nutshell, a generic PDA typically works in two steps: first, it understands the *intent* expressed by the user, next it invokes specific services that implement the algorithms that can fulfill the request [Sarikaya et al. 2016].

Even if these systems can support the users in a broad range of tasks in a very effective way (e.g., to set the temperature, to play relaxing music, to switch on the kitchen light, and so on), the current state of the art in the area of PDAs still shows a huge room for improvement.

As an example, the development of PDAs that include some kind of *personalization mechanisms* is at a very embryonic stage [Rafailidis 2018], since most of the PDAs typically work in a *cold-start* scenario [Lam et al. 2008], that is to say, the user who is interacting with the device is partially or completely anonymous. This issue is mainly due to the objective difficulty to build a user profile for *each* of the services the user interacts with. In fact, to provide a PDA with personalization mechanisms, the device should acquire information about users' interests and preferences in almost *every* domain. Unfortunately, this is a very time-consuming and boring task, and this is not feasible for a technology that is supposed to be immediately *ready to use*.

As a consequence, if a user asks Alexa to recommend a book to read, she obtains answers that may result as very *trivial* or far from her interests. Generally speaking, even when some sort of personalization mechanism exists, it is either very basic (e.g., based on heuristics, such as the popularity of an item) or very *opaque*, that is to say, the user is not aware of which personal information about her are hold by the device and how they are used to identify suitable suggestions.

In this article, we fit into this research line and we tackle both these issues by proposing a strategy to provide users with *personalized access* to the services available in a PDA. The main hallmark of our methodology is the adoption of an *implicit* user profiling strategy based on the ***holistic user modeling*** **(HUM)** paradigm [Musto et al. 2020b] that allows to introduce personalization mechanisms without an explicit and time-consuming acquisition of interests and preferences. In our opinion, the adoption of this paradigm perfectly fits the needs of the users who interact with a PDA, since they need a *ready-to-use* technology that can provide *open-domain* personalization without the need of an explicit elicitation of their preferences.

In a nutshell, a HUM is a comprehensive representation of the user, which is based on seven different facets, that is to say, *demographics data, interests, affects, psychological traits, behaviors, social connections, and health data*. These facets are inspired by those defined by Cena et al. [2018], who in turn inherited ideas previously presented in the area of *generic user modeling* [Kobsa 2001],

[1] https://www.invest.mywallst.com/post/2-reasons-why-big-tech-is-so-invested-in-voice-assistants.

ubiquitous user modeling [Heckmann et al. 2005], and *social semantic user modeling* [Plumbaum et al. 2011]. Each facet is populated by merging and processing heterogeneous data gathered from social networks, smartphones and personal devices through machine learning and natural language processing algorithms. For the sake of brevity, we can state that each service implements personalization mechanisms by using a different portion of the information available in a HUM of the user. As an example, food recommendations will rely on *health data* along with *height* and *weight* of the user, while music recommendations will exploit *mood* and *preferences*. More details about the whole process will be provided next.

Moreover, another distinctive trait of this work is the introduction of the concept of *queryable user model*, since we allow the users to *query* their own HUMs in natural language and to inspect which information are encoded in their own profiles. This characteristic, inspired by the popular paradigm of Quantified Self [Swan 2013] and fostered by recent regulations as the European GDPR, is supposed to make the personalization process more *transparent*, since the users can: (i) improve their self-awareness, by asking the PDA information about themselves (e.g., *Did I sleep enough tonight?*); (ii) increase the knowledge about the data and the information that are held by their profiles.

Generally speaking, *queryable user modeling* can be seen as an extension of the traditional *question answering* over personal data. In particular, we encoded some background knowledge that can be exploited to answer in a personalized manner to questions such as *"Shall I make more physical exercise?"* or *"Is it good for me to eat more sweets?"* Indeed, to answer these questions the system requires to take into account both personal characteristics of the users as well as some background information that can drive the generation of the answers.

Both these intuitions are implemented in MyrrorBot, a personal digital assistant based on *holistic user profiles* that provides personalized access to online services and allows the user to query their own user models. In the current version of the prototype, *seven* different services are available: music recommendation based on Spotify, video recommendation based on YouTube, news recommendation based on Google News, recipe recommendation through a knowledge-aware food recommender system [Musto et al. 2020c], weather information, workout recommendation based on data available on Darebee,[2] and tv-programs recommendation. As we will show in Section 3, all the services are available in both *retrieval* (e.g., *"Play a U2 song"*) and *recommendation* (e.g., *"Play a song I like"*) mode.

To sum up, through this article, we provide the following contributions:

- We propose a strategy to provide *personalized* access to the online services available in a PDA. Personalization mechanisms are based on an implicit user modeling strategy and exploit *holistic user profiles*. In particular, different strategies to provide users of a PDA with movies, video, news, music, workout, TV programs, and recipes recommendations are introduced in this work;
- We introduce the concept of *queryable user model*, that is to say, we designed a strategy to allow the users to query their own profiles through natural language requests. This contribution can be seen as an extension of the traditional *question answering* over personal data, since we also encode some background knowledge that is used to generate personalized answers. In our opinion, this can be useful to increase users' self-knowledge and self-awareness and to make personalization process more transparent;
- We validate our approach by designing a modular PDA that allows the users to interact with such *personalized* services and to *query* their own profiles;

[2] https://www.darebee.com/.

- We evaluate our design choices through a user study in which we investigated both *qualitative* (users' acceptance of the system, usability) as well as *quantitative* (time required to complete basic tasks, effectiveness of the personalization strategy) aspects of the system.

The remainder of the article is organized as follows: in Section 2, we present Related Work in the area. Next, Section 3 focuses on the description of the platform. First, we will introduce the profiling strategy we exploited in this work, and then we will describe the external services we integrated into the system and we will provide details of our strategy to interpret users' requests. Finally, in Section 4, we discuss the findings of the experiments, and Section 5 identifies future research directions and concludes the work.

2 RELATED WORK

This work investigates two different research lines, that is to say, conversational agents that provide users with (personalized) access to online services and strategies for transparent and scrutable user modeling. In this section, we provide an overview of related work in the areas, by emphasizing the distinguishing aspects of the current work with respect to relevant literature.

2.1 Conversational Agents

Conversational Agents is a broad concept that includes all those systems that interact with the user through a dialog. According to the goal of the dialog and the task the agent is able to perform, we can distinguish chit-chat from goal-oriented (e.g., to book a flight, to turn on a light, to listen to some music) conversations [Bordes et al. 2016]. This is not only a distinction that impacts on the final outcome of the conversation, but it influences the whole architecture implemented by the system.

In the literature, we can identify two main architectural approaches for Conversational Agents: *modular* or *end-to-end* systems [Dodge et al. 2015]. *End-to-end CAs* are mainly based on Deep Learning techniques and each system module is trainable from data (collection of dialogues) [Wen et al. 2016]. Once the training of the CA has been completed, the system is able to understand user's messages and to generate answers. End-to-end systems work definitively well for chit-chat conversations, but there are still a few works that effectively applied them in goal-oriented scenarios. Indeed, one of the problems in a goal-oriented conversation is to catch precise information that is required for accomplishing the task (e.g., from and to which city for a flight booking).

An overview of end-to-end language understanding and dialog management for personal digital assistants is provided in Sarikaya et al. [2016]. In the article, the authors focus their analysis on Microsoft Cortana and discuss the intent recognition strategy, performed as a multi-label classification task, as well as the ability of the system to switch among nine different domains.

Even if such an open domain knowledge also regards our work, we did not chose an *end-to-end* architecture. Indeed, the implementation of an end-to-end PDA requires the availability of a collection of dialogues that is very hard to obtain and update. With the exception of some attempts [Ren et al. 2020], this issue made the literature of end-to-end conversational agents very limited, especially for open domain systems. Indeed, just a few examples of available datasets have been discussed in literature. These include ReDial [Li et al. 2018], which consists of 10,000 conversations regarding movie recommendations, and Converse [Iovine et al. 2019], a dataset of real dialogues for movie, music, and book recommendations. To our knowledge, no dialogues in the area of workout recommendation, food recommendation, news recommendation, and video recommendation actually exist. This made this class of systems poorly suitable for our goals.

To summarize, even if these systems represent a promising solution, they still have to face some challenges before the potentialities of goal-oriented conversations can be entirely exploited. Accordingly, we preferred to design our system as a *modular* PDA. Indeed, MyrrorBot falls into

the category of Goal-oriented systems [Bordes et al. 2016], since the objectives regard the different services that are integrated in the platform, that is to say, music, video, food, news recommendation, and so on. A more detailed overview of the services will be provided in Section 3.4.

Modular agents (or *pipeline-based* agents), as the word itself suggests, consist of a set of components, each one devoted to a specific role [Liu and Lane 2017; Truong et al. 2017; Zhao and Eskenazi 2016]. These components are put together in a unique architecture to form a pipeline where the output of a module becomes the input of another. One of the first attempts that follow this idea is due to Rudnicky et al. [1999], while this setting has been recently adopted by Narducci et al. [2019] and by Iovine et al. [2020], as well.

Modular CAs typically consist of four different modules: a *Natural Language Understanding* component, which interprets user messages, a *Dialog State Tracker*, which keeps track of the state of the dialog, a *Dialog Policy*, which identifies the next action to be performed in a given dialog state, and a *Natural Language Generator*, which generates the natural-language messages sent to the user.

Unlike the above-mentioned classic structure, in this work, we also introduced a *profiling* and a *personalization* module, whose goal is to tailor the access to different services to the preferences and needs of the user. As previously introduced, personalization mechanisms rely on the *holistic user modeling* paradigm [Musto et al. 2020b]. This is the first distinctive trait this work. Indeed, up to our knowledge, even *commercial solutions* such as Alexa and Google Home adopt very rough personalization strategies and poorly exploit personal information to tailor the answers they generate.

Moreover, it should be pointed out that MyrrorBot distinguishes from **Conversational Recommender System (CoRS)**. Indeed, the distinguishing characteristic of a CoRS compared to a standard recommender is the capability of establishing a multi-turn dialog [Jannach et al. 2020], which is mostly devoted to the elicitation of user preferences. This step is not needed in MyrrorBot, since our PDA has the huge advantage of exploiting the user profile built through the HUM paradigm, and this drastically reduces the preference-elicitation step.

To conclude, we can state that this work belongs to the wide category of researches that investigated CAs and PDAs from an *human-computer-interaction* perspective. In particular, the dimensions that have been so far investigated in the literature, are related to *objective* metrics such as number of interactions, interaction time, number of utterances exchanged [Jin et al. 2019; Mahmood and Ricci 2009; Tsumita and Takagi 2019] and *subjective* metrics such as understandability, user-friendliness, usability score on the **System Usability Scale (SUS)**, user control, transparency [Ghosh et al. 2018; Jin et al. 2019; Li et al. 2017; Pecune et al. 2019]. Objective metrics are generally computed by recording and analysing the user interaction with the agent, thus they provide a *quantitative* analysis of the system. Subjective metrics bring out the user perception of the system and questionnaires are generally administrated to users to this purpose, thus these metrics provide a *qualitative* evaluation of the system.

The evaluation of MyrrorBot has been inspired by that proposed in Narducci et al. [2019]. In particular, we used *task completion time* as objective metric, while *subjective* aspects were evaluated by exploiting state-of-the-art questionnaires for usability evaluation and perception of the quality of the recommendations, such as the previously mentioned SUS [Brooke et al. 1996], User Experience Questionnaires [Laugwitz et al. 2008], AttrakDiff [Hassenzahl et al. 2003], and ResQUE [Pu et al. 2011].

2.2 Scrutable and Transparent User Modeling

The concept of *scrutable* (or *transparent*) user modeling [Kay et al. 2002] concerns the idea of allowing the users to inspect and to become aware of the information encoded in their own user profiles.

Even if this concept was originally spread in the e-learning domain [Bull and Kay 2010; Kay and Lum 2005], the first system implementing this principle was the PersonisAD [Assad et al. 2007]. Next, Kyriacou proposed in Kyriacou [2008] a user profiling architecture aiming at building transparent user profiles. Next, many relevant attempts of introducing transparent user profiling strategies have been proposed throughout the years. As previously stated, the e-learning domain was one of the most active. In this area, we can cite both the contributions by Sateli et al. [2016] and Guerra-Hollstein et al. [2017] who propose the adoption of an *open learner model* to represent the characteristics of students and learner in a scrutable way. Similar attempts were proposed in the *restaurant* domain [Wasinger et al. 2013] and in the *news* domain [Ahn et al. 2007; Wongchokprasitti and Brusilovsky 2007]. In a nutshell, all these attempts share the common idea of adopting a representation of the user that is transparent and can be inspected through classical tools, such as web-based interfaces.

More recently, this topic got new interests thanks to the European GDPR regulation [Voigt and Von dem Bussche 2017], which emphasized the *users' right to explanation* and fostered the research in the area of algorithmic transparency. It is not by chance that recent work by Google researchers [Balog et al. 2019] focused on the development of a strategy for explainable and transparent user modeling, that inherits the same ideas previously spread by El-Arini et al. [2012] and Gianforme et al. [2009]. A strategy for scrutable user modeling is also presented in Narducci et al. [2013], where the authors propose the use of semantic representation techniques for a transparent modeling of users' interests. Recently, in Musto et al. [2020b] the authors introduce the concept of *holistic user modeling*, a transparent and comprehensive representation of the user, which is obtained by processing and merging rough data coming from several heterogeneous sources (social networks, wearable devices, etc.). In our work, this paradigm was chosen as user profiling strategy due to its *unobtrusiveness* and the overall *transparency* of the process, since each user is provided with a full control of which personal data are encoded in the profiles and which ones have to remain private. In this work, we extended the interaction mechanisms presented in Musto et al. [2020b] by designing a natural language interface to query the information encoded in a HUM, to improve the way people access to the plethora of personal data available in such profiles.

Generally speaking, regardless the comprehensiveness and the transparency of the resulting profiles, all the above mentioned work share the idea of allowing the access to personal data through standard strategies, such as web interfaces and classical data visualization techniques [Tsai and Brusilovsky 2017]. In this area, effective attempts are those presented in Bakalov et al. [2010] and in Büring and Reiterer [2005], where the adoption of scatter plot to access to the information handled by personal digital assistants is discussed. Similarly, in Le-Phuoc et al. [2014] the authors present a specific architecture that allow to inspect users' personal data through query languages such as SPARQL.

Unlike the above-mentioned research, in this work, we investigate the intuition of exploiting *a natural language interface* to query a user model and to allow the user to inspect the information encoded in her own profile. It should be pointed out that this article significantly extends the findings already presented in Musto et al. [2020a], where a preliminary investigation of the adoption of queryable user modeling is presented. In particular, this article introduces several new contributions such as: (i) a detailed discussion of the personalization strategies we implemented for each service integrated in the system; (ii) a more extensive experimental evaluation, that also includes a user study evaluating the effectiveness of our personalization strategies; (iii) a more detailed discussion of the motivations of the current work as well as of related literature. To conclude, we want to emphasize that, up to our knowledge, the design and the development of a *queryable user model* is a relatively new research direction in the area. In our experiments, we will evaluate to

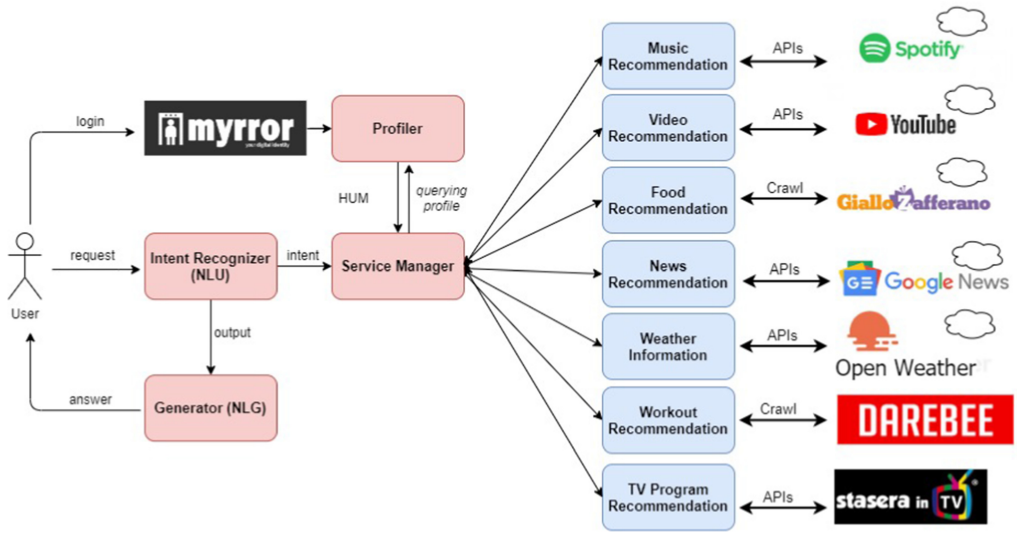

Fig. 1. Architecture of Myrrorbot.

what extent this strategy can be helpful to increase the transparency of user profiling process and to allow the users to improve their self-awareness.

3 DESIGN AND IMPLEMENTATION OF MYRRORBOT

The architecture of MyrrorBot is sketched in Figure 1. As shown in Figure 1, our PDA inherits the structure of *modular CAs* we have introduced in Section 2.1. Indeed, the system consists of four main components: two of these are typical of CAs, such as the Intent Recognizer, whose goal is to interpret the requests of the user, and the Generator, that returns the natural language answer fulfilling the information need. Next to these components, we also implemented a Profiler module, whose goal is to acquire and exploit the profiles of the users, and a Service Manager that acts as a dispatcher that routes users' requests to specific services and modules. To build the profiles of the users, we relied on the data exposed by Myrror,[3] a platform that implements the principles of *holistic user modeling* we have previously discussed.

Each of the different *intents* our system can handle is managed by invoking external services through their APIs or by crawling publicly available data. As previously stated, seven different *personalized* services are available in this release: *music, video, food, news, weather information, workout, and tv-programs*.

Moreover, we point out that *user login* is not mandatory. If the user wants to connect her Myrror profile, then she will receive personalized recommendations; otherwise, she can interact with the services that are available in MyrrorBot without any personalization (e.g., "Play a U2 song").

In the following, we will introduce all the modules that compose the architecture of MyrrorBot. In particular, we will first focus on *what* the system can do (that is to say, which *intents* can catch and what *answers* it returns); next, we will discuss *how* the system provides personalized access to online services, by discussing our *user profiling* and *recommendation* strategies.

[3]For a screencast showing the interaction with Myrror, we refer the reader to https://www.youtube.com/watch?v=3YRlcUhNZnQ&t=2s.

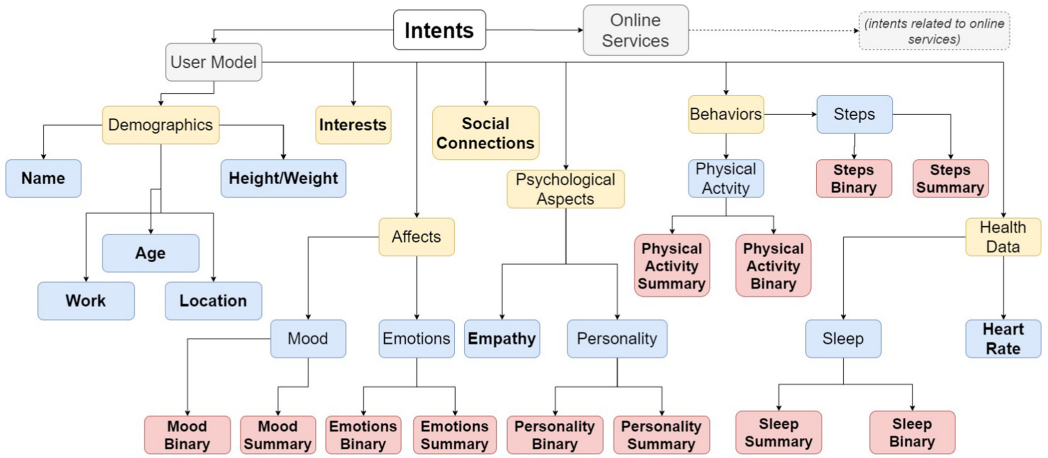

Fig. 2. Hierarchy of User Model-related Intents. Facets and Features encoded in HUMs are reported in yellow and blue, respectively. Some intent is further split according to the type of requests the users can forward (red boxes). Intents are reported in **bold**.

3.1 Intent Recognizer

Every time a user interacts with PDAs, she is asking to fulfill a particular information need. Clearly, each information need can be expressed through several utterances, thus it is necessary to implement a component whose goal is to: (i) understand the *meaning* of each request expressed by the user; (ii) forward such information to the module that dispatches the request to the correct service. In our case, this role is played by an Intent Recognizer.

The design and the implementation of an intent recognizer revolves around the concept of *intent*, which is a representation of a particular information need of the user. As for MyrrorBot, examples of possible *intents* include asking for a music recommendation, reading a specific news, checking the weather for the weekend and so on.

Formally, given a user request r and given a set of intents $i_1 \dots i_n$, an intent recognizer assigns a score $s(r, i_k)$ to each intent i_k, with $k = 1 \dots n$. Such score represents the likelihood that a particular intent meets the request conveyed by the user. In our case, given a input request r (e.g., *"Play a U2 song"*) the intent i_j having the highest score (e.g., *music recommendation*) is returned by the IR, which in turn will invoke the service that will handle the request.

The intent recognizer currently implemented in MyrrorBot is based on DialogFlow,[4] a Google toolkit that allows to build *conversational interfaces*. The recognition strategy implemented in DialogFlow is based on *machine learning*. Hence, for each *intent*, the system needs to be *trained* by feeding the intent recognizer with an adequate set of *input examples* that cover the different expressions the user can use to formulate a particular request. In our case, we fed the intent recognizer with approximately 20–30 examples for each intent.

An overview of the intents that can be handled by MyrrorBot is reported in Figures 2 and 3. Due to the high number of intents our system can recognize, for the sake of readability, we have split the diagram into two figures: the first one shows *user model-related* intents, while the second one focuses on *services-related* intents.

The goal of user-related intents (see Figure 2), is to query all the facets of the *holistic user profiles*. As previously introduced, a HUM is based on seven different facets: *demographics data, interests,*

[4] https://dialogflow.com/?authuser=1.

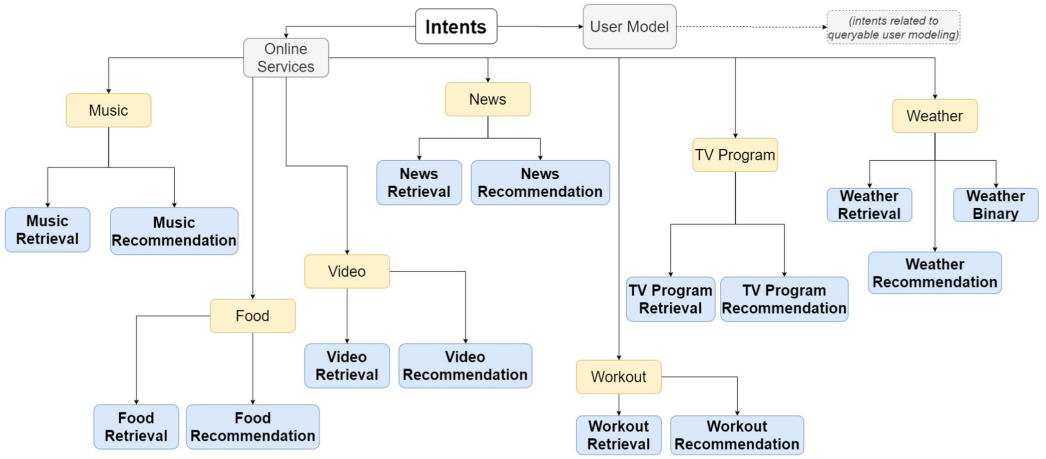

Fig. 3. Hierarchy of Services-related Intents. Available services are reported in yellow, while blue box describe the intents the system can catch. Intents are reported in **bold**.

affects, psychological traits, behaviors, social connections, and health data. In particular, the goal of these intents is to allow the users to: (i) ask which information about them are encoded in the profiles (e.g., *What are my interests?*); (ii) increase their self-awareness and acquire new knowledge (e.g., *Am I an extrovert person?*). Of course, the answers returned by MyrrorBot depend on the data that are available in the profile.

Even if this functionality is based on a simple *retrieval* of the features encoded in the profile, the novelty of the approach lies in the opportunity to *query* a profile of the user *in natural language*. Up to our knowledge, this is a completely new research direction in the area of PDAs and user modeling strategies. Our choice is supposed to make both profiling and personalization processes more transparent and easy to understand for the users, since we provide the users with a complete control of the information about them, which are encoded by the platform. More information about the pipeline that is run to build and obtain HUMs is reported next.

As shown in Figure 2, such intents are basically mapped to the facets (*yellow* boxes in Figure 2) and to the features (*blue* boxes) available in each HUM. Moreover, it should be pointed out that some features were further split into two more fine-grained intents (*red* boxes), depending on whether the user is interested to ask a summary of the information encoded in the facet (e.g., *How much did I sleep tonight?*) or to acquire new knowledge (e.g., *Did I sleep enough?*). In the first case, the generation of answers is based on the extraction of the corresponding (in this case, sleep time) from the HUM. In the latter, the actual value is also compared to a threshold value, which is based on common-sense knowledge (e.g., 8 h per day). Based on the comparison, a different template is instantiated (e.g., "Yes, you slept enough" vs. "No, you should sleep more") and is returned by the user. More details about this will be provided next.

Similarly, services-related intents (see Figure 3) are mapped to the online services connected to MyrrorBot. In this case, we have reported the services with *yellow* boxes and the intents with *blue* boxes. As shown in Figure 3, each service can be invoked in a *retrieval* and *recommendation* mode, that is to say, by explicitly forwarding a request to MyrrorBot (e.g., *I want to read a news about Kobe Brant*) or by asking for a recommendation (e.g., *Give me some news I'm interested in*). Finally, we underline that a third intent is also available for weather service, whose goal is to provide a binary answer about weather forecasts (e.g., *Will tomorrow be cloudy?*).

In total, our system is able to catch 36 different intents (reported in **bold** in Figures 2 and 3).

3.2 Generator

The goal of the Generator is to acquire the output returned by the services and to further process it to finally provide the user with the *answer* that fulfills her information need.

In our case, the answers are presented in *natural language* and they are obtained through a *natural language generation* strategy [Reiter and Dale 2000], which is completely unsupervised. In particular, each answer is based on a template that consists of a *fixed* part, which is the same for all the users, and a *dynamic* part, which depends on the output of the services.

Some examples of the *answers* returned by the system are provided in Table 1. As well as for the Intent Recognizer, we have split the answers into *services-related* answers and *user model-related* answers. Of course, *user model-related* intents are available only for the users who have an active Myrror profile, while *services-related* intents can be formulated also by guest users. In this case, they will receive non-personalized answers.

Each line of the Table provides an example of the interaction between a user and MyrrorBot. As an example, *music service* presents its output by using the string *"I am playing"* as *fixed part*, while the dynamic part depends on the song or the playlist returned by the recommendation component. A similar behavior can be noted for the other services. In most of the cases, natural language answers are accompanied by some *multimedia* content, such as a *cover image* for a recommended recipe, or a widget that allows the user to interact with the recommended item, as it happens for music recommendation. As for *user model-related* intents, it should be pointed out that some requests require that the system encodes some background knowledge about *well-being* and *healthy habits*. As an example, if the user asks whether she slept enough or whether she made enough physical activity, the answer of the system is dynamically generated by comparing the features extracted from the HUM of the user with some *threshold* value (e.g., 8 h per day, as for average sleep time).

A more detailed overview of the output returned by the Generator is discussed in Section 3.4, where some screenshots showing the interaction between a user and MyrrorBot is provided.

3.3 Profiler

Most of the intents we have presented in Sections 3.1 and 3.2 need information about the current user to personalize the interaction with the system and to allow to inspect the features encoded in the profile. In MyrrorBot, the Profiler is the component that is devoted to the acquisition and the management of the profiles of the users.

As previously stated, the novelty of this work mainly lies in the conceptualization and in the design of such a component, since the PDAs currently available are based on very *naive* personalization and recommendation strategies. Conversely, the distinctive trait of this work is the adoption of a representation of the user than allows to provide personalized and *open-domain* recommendations by exploiting a comprehensive and heterogeneous set of personal characteristics, including preferences, mood, emotions, personality, health data, and so on.

Generally speaking, techniques to build a *user profile* can be roughly split into two classes: *explicit* and *implicit user modeling*. In the first case, the user is asked to *explicitly* provide information about her (demographics, interests, etc.), while in the latter such data are automatically inferred by analyzing user behavior or are obtained by exploiting external *user modeling services*.

As previously clarified, explicit user modeling is not feasible for general-purpose PDAs, since the agent should acquire a huge set of information about users' interests and preferences in almost *every* domain, and this task is particularly boring and time-consuming. As a consequence, we focused our attention on methodologies for implicit user profiling, and we chose the *holistic user modeling* as strategy to obtain a multi-domain representation of the users. In particular, we relied

Table 1. Services-related and User Model-related Requests Defined in MyrrorBot

	Intent Class	User Message	MyrrorBot Answer
SERVICES	Music	*(i) Play U2 - One - (ii) I need a playlist for a party*	*(i) I am playing U2 - One - (ii) I am playing Fiesta! playlist*
	Video	*(i) Find videos about Kobe Bryant - (ii) Recommend me a video*	*(i) Enjoy this video: Kobe Brant - Best Plays - (ii) - This is a video you might like Top 50 Impossible Shots of the 2017 NBA Season*
	Food	*(i) I need a recipe with chicken - (ii) Can you suggest me a recipe?*	*(i) I suggest American Fried Chicken - (ii) I suggest Pumpkin Risotto*
	News	*(i) The latest news about basketball, please - (ii) Recommend me a news!*	*(i) What Bowen Hardman's commitment means for Ohio State basketball https://tinyurl.com/yan9tdtf - (ii) I think you should read this article: https://tinyurl.com/tlwkm2g*
	Weather	*(i) What will the weather be in Bari?*	*(i) The weekend will be clear or very slightly cloudy*
	Workout	*(i) I want a cardio workout! - (ii) Recommend me a workout*	*(i) I recommend you 30 minute-walk (ii) I recommend you Fresh Start*
	TV Program	*(i) I want to watch a documentary - (ii) Recommend me a TV Program!*	*(i) On Italia 1 at 21.00 will go on air The Last Dance - (ii) On Rai 1 at 20.35 will go on air I Dieci Comandamenti*
USER MODEL	Demographics data	*(i) How old am I? - (ii) Where do I live? - (iii) Where do I work?*	*(i) i) You're 30! - (ii) You live in Bari - (iii) You work in research at University of Bari*
	Health Data & Behaviors	*(i) Tell me about my physical activity? - (ii) Do I burn enough calories? - (iii) What is my heart rate?*	*(i) According to my data, you did 30 minutes of very active activity! - (ii) No, you should burn more calories - (iii) Your heart rate is 91bpm*
	Interests	*(i) What are my interests?*	*(i) Your interests regard recsys, inter, u2, music*
	Affects	*(i) What is my mood?*	*(i) According to the available data, your mood is neutral*
	Psychological traits & Social connections	*(i) Tell me about my personality - (iii) What are my main contacts on Twitter?*	*(i) You are curious, organized and energetic. Moreover, you are also friendly and compassionate - (ii) Your main contacts are semeraro_g and SWAP_research*

For the sake of readability, Some intents regarding different facets are merged in a single line.

on the profiles exposed by Myrror, a platform that implements the principles of *holistic user modeling*.

Myrror [Musto et al. 2020b] works as an *aggregator* of user personal data gathered from heterogeneous sources, such as social networks, smartphones, and wearable devices. In particular, the current implementation of Myrror allows the user to link six different digital identities: *Facebook, Twitter, LinkedIn, Instagram, FitBit devices, and Android smartphones*. Obviously, it should be pointed out that the user has to *explicitly* grant the access to the data source and has to allow the extraction of the data.

All the information extracted from the sources are processed through natural language processing and machine learning techniques and are mapped to the seven different facets that compose a HUM. As previously stated, these facets include *demographics, interests, affects, psychological aspects, behaviors, social connections, physical states*. The next table provides an overview of the facets that compose a HUM, along with some examples of the features encoded in each facet. As can be seen, most of the features are mapped to the *intents* the system can handle.

Facet	Features
Demographics	Name, Surname, Sex, Age, Height, Weight, Nationality, Last Location, etc.
Interests	Preferences and Topics discussed
Affects	Mood and Emotions
Psychological Aspects	Personality traits and Empathy
Behaviors	Visited places, Amount Daily Activity
Social Connections	Contacts and Interactions
Health Data	Amount of Sleep, Heart Rate, Diseases, etc.

3.3.1 Building a Holistic User Profile. A thorough discussion of the machine learning and natural language processing pipeline that is run to populate the different facets of holistic user profiles is outside the scope of the current article, and we suggest to refer to Musto et al. [2020b] for more details about the process. For our goals, we can assume that a HUM of the target user is built and can be exploited to provide personalized services in MyrrorBot.

However, to make the article more self-consistent, in this section, we briefly introduce how the features that compose a HUM are obtained.

Demographics: Our HUM includes several demographic features, such as *name, surname, e-mail, location, height, weight, working position*, just to name a few. This is a fundamental facet to build an effective user model [Dong et al. 2017], and it is populated by *mapping* the demographics information available in each data source (e.g., Facebook, LinkedIn) to the corresponding feature. To populate this facet is simple and straightforward, since analogous features (e.g., gender or location) typically have the same (or a very similar) name on different data sources. The strength of our approach lies in the fact that different social networks encode different characteristics of the user: as an example, data about user height and weight are only available on a single data source (i.e., FitBit). In this case, Myrror works as an aggregator and allows to merge together the different demographics information about the person that is available in the different data sources.

Interests: Information about user interests is obtained by mining: (i) categories of the Facebook pages a user likes (e.g., *politics, technology, etc.*); (ii) categories of the apps the user frequently uses (e.g., *social networking, games, sport news, etc.*); (iii) *topics* that are typically discussed by the user as well as the *concepts* that are mentioned in her own posts.[5] Given these raw information, this facet is populated by extracting through natural language processing techniques: (i) the keywords describing Facebook pages the user likes; (ii) the keywords describing the apps used by the user; (iii) the keywords extracted from users' posts, along with the *concepts* and the *entities* returned by an entity linking algorithm [Derczynski et al. 2015].

[5]The term "posts" is used to indistinctly refer to Facebook posts, Instagram posts, and Tweets.

Affects: Affects, such as *mood* and *emotions*, are inferred from textual content. To populate this facet, we defined a simple mechanism that relies on the output of the returned by Sentiment Analysis [Basile and Novielli 2014] and Emotion Detection [Polignano et al. 2018] models that are integrated in the system. In both the cases, the input for the models is represented by the posts written by the user during the last day, and the output is the *main sentiment* (or the *main emotion*, respectively) returned by the machine learning model.

Psychological Aspects: Psychological aspects, such as *empathy* and *personality traits*, are inferred from textual content, as well. As for the personality traits, we used textual content as input to run a ML model for personality detection [Polignano et al. 2017], and we stored in Myrror the scores for her Big Five Personality traits (*Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism*) [Goldberg 1993], while as for the inclination to empathy [Hogan 1969], a categorical score (*high, medium, low*) is obtained and stored.

Behaviors: Information about users' behaviors can be obtained by exploiting FitBit or Android data. In particular, all the activities gathered from FitBit (running, walking, etc.) are collected and used to fill in this section of the profile. Alternatively, information coming from GPS sensors can be used to infer whether the user is making some activities.

Social Connections: Social connections are filled in by gathering data coming from both Android phone and social networks. Specifically, this facet is populated through a mechanism that executes the following two steps: (i) each contact extracted from all the data sources linked to the system is stored in the facet as a social connection; (ii) the strength of the tie between the user and the contact is calculated on the ground of the number of phone calls or on the number of interactions on social networks (*likes, favorites, retweet, etc.*) they have.

Health Data: This facet is filled through a simple mechanism that *maps* FitBit data to the attributes of our HUM. Specifically, all the information about *sleep* and *heart rate* are stored in this section of the profile. As for sleep, acquired data include the average number of *hours of sleep*, *time to get awake*, and so on. Similarly, data about heart rate include the average *heart rate*, *peak heart rate*, and so on.

Discussion: The previous paragraphs have shown the large pool of information we can acquire from a HUM. Clearly, the goal of this article is *not* to analyze how effective is our strategy to implicitly build holistic user profiles, since this was already investigated in our previous research [Musto et al. 2020b]. As previously stated, for our goals, we can assume that a HUM of the user is already populated and exists.

Rather than discussing strength and weaknesses of the methodology, we want to further stress the motivations that led to the choice of Myrror as external user modeling service. In our opinion, a PDA that is supposed to support the user in a personalized manner in several heterogeneous tasks needs to rely on a representation of the user that is as much comprehensive and rich as possible. In this sense, Myrror represents a good choice, since HUMs encode a huge and rich set of features that describe many different characteristics of the person, and this can be helpful to adapt the behavior of the PDA on the ground of such information. As an example, music recommendations could take into account *emotional state* and *interests* of user, while food recommendations could rely on *health data* and *health/weight* of the person, and so on. More details about the recommendation strategies implemented in the services we integrated in MyrrorBot is discussed next.

To conclude our overview, In Table 2, we report an example of a *populated* holistic user profile. This profile will be used in the next subsections as a *running example* to better explain the behavior of the personalization strategies implemented in MyrrorBot.

Table 2. Example of a Populated Holistic User Profile

Facet	Feature
Demographics	*name*: user; *surname*: demo; *email*: demo@demo.it; *location*: Bari *gender*: male; *language:* italian; *age*: 37; *work*: Researcher at University of Bari; *height*: 175; *weight*: 79
Interests	*preferences*: { nba, depeche mode, u2, kobe bryant, android, iphone, lebron james, rock music, indie music, business news, technology news, sport news, sport movies, crime movies; drama movies; talk show }
Affects	*mood*: neutral; *emotion*: none
Psychological Aspects	*personality*: { openness: high; consciousness: medium; extroversion: high agreeableness: medium; neuroticism: low }; *emphathy*: medium
Behaviors	*activity*: { light: 4 min; fair: 10 min; active: 25 min }; *caloriesBurnt*: 250
Social Connections	*contacts*: { semeraro_g: 1; cataldomusto: 5, swap_research: 3 }
Health Data	*restingHearthRate:* 91; *sleep*: { asleep: 400 min, inBed: 100 min }

Will be used as running example throughout this section.

3.3.2 Querying User Profile. As previously stated, one of the hallmarks of the current work lies in the adoption of a *natural language interface* that allows the user to inspect and query her own profile. This functionality, which enables the vision of a *queryable user model*, is designed to fulfill several information needs. As an example, a user may have *informative* queries related to the understanding of the information that is encoded in her profile: she can ask what *interests* the system is currently inferring about her, or which *personality traits* have been predicted by analyzing her posts. The goal of these queries is to make both profiling and personalization processes more transparent, since the user can inspect in every moment the current representation of her profile held by MyrrorBot. This is a relevant design choice that distinguishes MyrrorBot from most of the PDAs currently available and from most of the online services the users interact with. Indeed, the profiles encoded by these systems are completely *opaque* and cannot be examined by the users. Conversely, we preferred to provide the users with a *transparent* access to their personal information. An example of such interactions, which are based on the *user profile* presented in Table 2, are reported in Figure 4.

Next to these queries, the user can also exploit the platform to increase her *self-awareness*. To this end, we designed some intents that are inspired by the previously mentioned Quantified Self principles, since they allow the user to *increase* her knowledge about personal lifestyle and personal habits. As an example, she may ask whether she *slept enough*, or to quantify the *amount of physical activities* she did today. An example of such interactions, which are based again on the *user profile* presented in Table 2, are reported in Figure 5.

By following this intuition, the user may use the platform to express many different requests related to her habits or her personal goals. As an example, the user may ask whether she is overweight or not, whether she has a correct lifestyle or whether she is eating too much sugar. This group of queries clearly distinguishes *queryable user modeling* with respect to the traditional *question answering* over personal data. Indeed, to correctly answer to these questions it is necessary to encode some background knowledge that drives the generation of the replies.

Currently, we are only able to catch a limited set of intents regarding quantified self-related goals, since their handling requires the acquisition and the exploitation of a huge set of background knowledge (e.g., *When can a lifestyle be labeled as correct? What is the correct amount of sugar?*

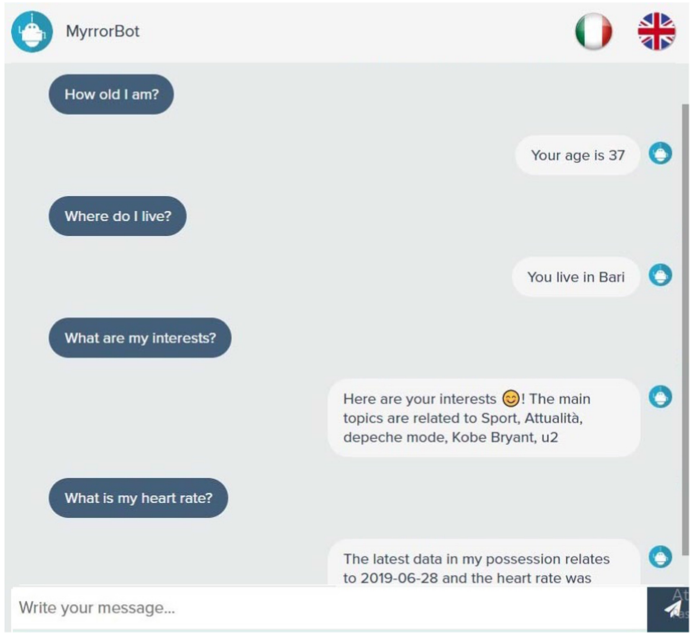

Fig. 4. Example of *informative* queries.

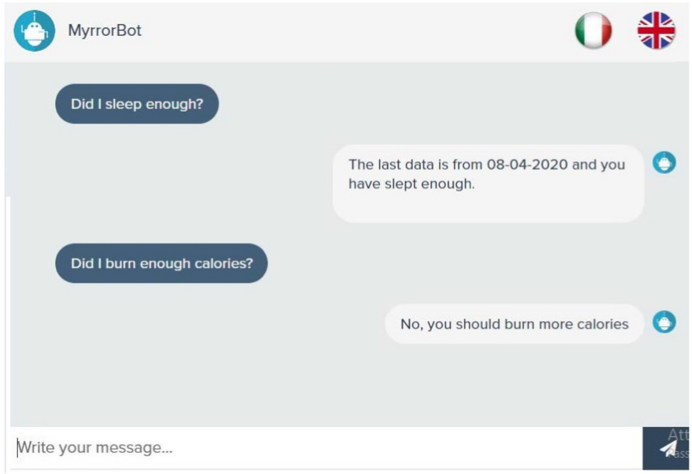

Fig. 5. Example of *quantified self-related* queries.

etc.), which is not easy to gather and encode. As future work, we will significantly extend the set of quantified self-related requests the user can forward to the platform, to investigate whether MyrrorBot can be effectively exploited to trigger *behavior changes* [Masthoff et al. 2014] and healthier habits.

3.4 Service Manager

Once the user's request has been interpreted by the Intent Recognizer, the information need is processed by the Service Manager. This module has the responsibility to properly query the

services connected to the platform and to forward to the Generator the elements returned by the services. For example, if the user asks to MyrrorBot to *"Play a U2 song,"* then the Service Manager will query a music service endpoint (e.g., Spotify) by asking for a U2 song.

As previously stated, MyrrorBot provides personalized access to seven different online services: *music, video, food, news, weather information, workout, and TV programs*. However, the modular structure of our framework easily allows to extend the catalogue of the services offered to MyrrorBot users. As future work, we plan to further extend the list oft he available services.

Each online service can be queried through a dual-modality: *retrieval* and *recommendation*. In the first case, the user explicitly formulates her information needs through specific and detailed questions (e.g., *"Play a video about NFL"*) and will receive a non-personalized answer that fulfills her request. In the latter, *logged* and *guest* users are provided with different answers: as for guest users, non-personalized recommendations based on simple popularity-based mechanisms are provided. Conversely, as for logged users, the large pool of information gathered from HUMs is taken into account to generate suggestions. As we will show in the next paragraphs, we developed a different recommendation strategy for each service. These strategies exploit a mix of techniques ranging from classic content-based algorithms [De Gemmis et al. 2015] exploiting user preferences to *stereotype-based* techniques [Rich 1979] that are based on the characteristics of the users that are available in HUMs. Collaborative filtering algorithms were not taken into account due to the *sparsity* problem, since each user independently interacts with the PDA, and the preferences of the other users who previously used MyrrorBot are not available.

Finally, we want to point out that we also provided MyrrorBot with some *explanation* facilities. In particular, when the user asks for a recommendation and she wants to know the motivation of that suggestion, she can forward to the PDA the question *"Why?"* In this case, a simple *content-based explanation strategy* [Friedrich and Zanker 2011] will identify the properties encoded in the HUM that have triggered the generation of the recommendation.

For the sake of completeness, we want to stress that a *demo version* of our system—based on the profile reported in Table 2—is available online and can be used in all the functionalities.[6] The interaction with MyrrorBot and the output returned by the system are also shown in a screen cast available on YouTube.[7]

3.4.1 Music Service. Music retrieval and recommendation services available in MyrrorBot rely on Spotify APIs. As for *retrieval-based* intents, the user can ask for a *playlist*, for songs played by a specific *artist* and for a specific *track* identified by its title. Moreover, the user can ask for a *context-aware* playlist tailored on specific tasks (e.g., a playlist listen to during a workout or during a party). It should be pointed out that such task-aware playlists are natively supported by Spotify; thus, we did not implement any personalization mechanism in this case.

Next, as for *music recommendation* intents, logged users can receive personalized recommendations of tracks and playlists. In particular, we designed three different recommendation strategies:

(1) strategy based on user preferences;
(2) strategy based on user's personal characteristics;
(3) strategy based on user's emotional state;

In case the user asks for a personalized music recommendation based on her preferences (e.g., *"Play a song I like"*), the platform will check the *"Interests"* section of her HUM and will query Spotify by looking for *artists* or *genres* that match one of the keywords stored in the profile. Next,

[6] MyrrorBot demo: http://90.147.102.243:8080/demo/chatbot.html.
[7] https://www.youtube.com/watch?v=C-iQUc8L8dc.

Table 3. Mapping between Age, Activity, and Music Genre, for Personalized Music Recommendation

	Physical activity	
Age range	**Limited**	**Frequent**
≤ 20	Pop/Rap	Punk/Metal
$20 < x < 50$	Relax/Romantic/Blues	Electronic/Summer/Rock
≥ 50	Classic/Jazz	Aged/Folk

Table 4. Mapping between User Emotional State and Playlists, for Personalized Music Recommendation

	Fear	**Anger**	**Sadness**	**Neutrality**
Playlists	Motivation mix Monday motivation Life sucks	No stress Young, wild and free The stress buster	Operazione buonumore! Happy beats Happy hits!	Hot hits Italia Power it Latin Pop classic
	Joy	**Disgust**	**Surprise**	
	Have a great day Wake me happy Just smile	Sad beats Alone again	Sorridi! Canto sotto la doccia Il caffè del buongiorno	

the most popular song for each artist/genre that matched the previous query is collected and the final recommendation is obtained by randomly picking one of those songs.

On the contrary, if the user asks for a recommendation based on their personal characteristics (e.g., *"Play a song that is suitable for me"*), MyrrotBot will suggest a song by following a *stereotype-based* personalization process that follows the schema presented in Table 3. Such stereotypes exploit heuristics and common-sense background knowledge and are based on *age* and amount of *physical activity* of the target user.

In particular, MyrrotBot will consider the *age* feature stored into the *"Demographics"* section of the HUM and the *activity* feature collected from the *"Behaviors"* section to drive the personalization. Next, given the age and the amount of physical activity (*frequent* if the amount of activity overcomes 30 min, *limited* instead) a personalized playlist is returned to the user. By referring to the running example presented in Table 2, i.e., a user 37 years old with less than 30 min of total physical activity in the last week, the system will suggest *relaxing, romantic, or blues* tracks.

Finally, MyrrotBot can also suggest a specific playlist based on the user's emotional state (e.g., *"Play songs based on my mood"*), which is a very novel functionality for a PDA. An example of such a request is shown in Figure 6. Given the latest emotion experienced by the user (according to the data available in her HUM, of course) a personalized playlist, chosen from a set of three previously defined playlists, is presented to the user. An example of the association rules that drive mood-based personalization is reported in Table 4. As shown in Figure 6, always based on our running example, the user is provided by a playlist that is based on her *neutral* emotional state.

For the sake of completeness, it should be pointed out that the personalization mechanisms we implemented in MyrrorBot represent an *extension* of those available when a user interacts with Spotify. Indeed, Spotify personalization algorithms are mainly based on user demographics, user preferences and listening habits. Conversely, our algorithms is based on HUMs, thus it can also take into account more sophisticated user features such as user mood and the amount of physical activities she carried out. In our opinion, this is supposed to provide more diverse and unexpected music recommendations, since our suggestions are based on personal information different from the simple users' preferences an listening habits.

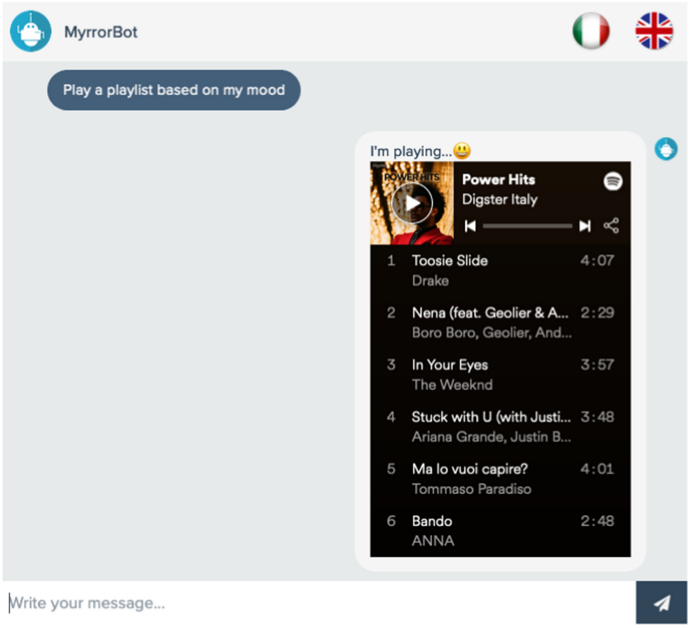

Fig. 6. Music recommendation based on user *emotional state* in MyrrorBot.

Moreover, we want to emphasize again that the user profile used by *Spotify* is completely opaque and not accessible through APIs outside the platform; thus, we could not directly exploit (or compare to) the recommendation algorithms available in Spotify. Indeed, when people interact with Spotify, they are not aware of which features are encoded in her profile nor how they are used to provide recommendations. Conversely, MyrrorBot provides a queryable and transparent representation that can be used to improve users' knowledge and self-awareness. Of course, the goodness of our recommendation strategies will be thoroughly evaluated in the experimental sessions.

3.4.2 Video Service. YouTube APIs are used to include *video retrieval and recommendation* in MyrrorBot. In this case, the user can ask for a specific video (e.g., *"Play a video about F1"*) or for a recommendation. The example in Figure 7 shows the correct identification of an entity (Formula 1, in this case) in a *retrieval* scenario. Such entity is used as a filter and it is exploited to query YouTube APIs to find a suitable video.

Next, as for *video recommendation*, suggestions are generated by using the same principles of music recommendations. Indeed, the suggestions can rely on the *interests* of the user as well as on the *emotional state* that is encoded in the HUM. In this case, similar to what we have previously explained for music recommendations, we defined associations between moods and video categories, summarized in Table 5 to provide users with video recommendations based on their emotional state.

As we already did for our music service, also for video recommendation it is important to point out again that: (i) we could not directly use YouTube recommendation algorithms and user profiles, since they are not made available through APIs to external services; (ii) we exploited a larger pool of information about the user (e.g., user mood) that is not typically used in video recommendation services. Thus, we can state that our approach is based on a richer and a more comprehensive representation of the user.

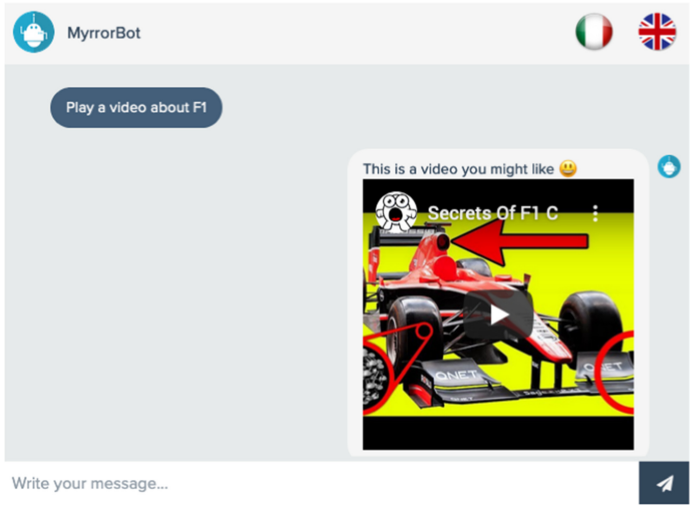

Fig. 7. Video Retrieval on MyrrorBot.

Table 5. Mapping between User Emotional State and Video Categories, for Personalized Video Recommendation

	Fear	Anger	Sadness	Joy	Disgust	Surprise	Neutrality
Video Categories	Motivational videos	Relaxing videos	Funny videos	Funny videos	Pleasant videos	News	News

3.4.3 Food Service. GialloZafferano is a very popular Italian community discussing about food and recipes. GialloZafferano data are used in MyrrorBot to provide users with recipe recommendations. In this case, the system can support the user in several tasks, such as: (i) asking for a recipe with a specific ingredient, (ii) asking for a recipe that matches her food requirements or dietary goals (e.g., vegan, lactose-free or healthy recipe), (iii) asking for a particular type of recipe (e.g., main course, second course, etc.).

Similarly to the other services, MyrrorBot also supports *food recommendation*: in particular, *guest* users will be provided with a popular recipe, while logged users will receive personalized recipes that will take advantage of the information that is available in HUMs such as *height, weight, mood, amount of sleep, and amount of physical activities*. In this case, it is necessary to point out that GialloZafferano platform does not provide any recommendation algorithm or personalization strategy. Accordingly, we just collected data from the community, and we used the information to design a recommendation service that is completely new.

The recommendation strategy we implemented in MyrrorBot relies on the *knowledge-aware* algorithm presented in Musto et al. [2020c], and we suggest to refer to the article for a complete discussion of the algorithm. For the sake of brevity, we can state that the personalization mechanism adopts a *knowledge-based reasoning strategy* that is based on common-sense background knowledge about healthy food and healthier lifestyle, which is encoded as *rules*. In particular, such rules increase/decrease the relevance score assigned to each recipe on the ground of the specific characteristics of the person. As an example, a user in perfect shape who did a lot of physical activity will receive as a recommendation a caloric recipe with a high amount of proteins. Conversely, a overweight and sedentary user will receive a lighter recipe that is more tailored to her food needs.

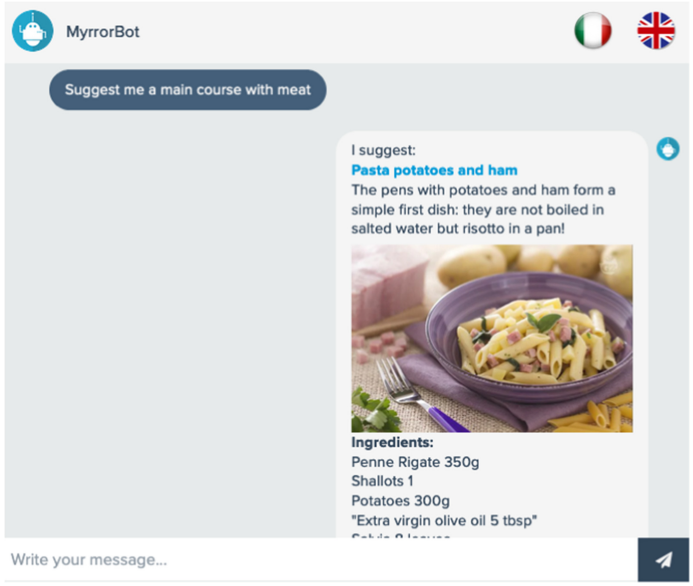

Fig. 8. Food retrieval on MyrrorBot.

Our expectation is that the usefulness of MyrrorBot as a personalized PDA will be mostly perceived when the user formulates requests regarding more *complex* information needs, such as identifying the most suitable recipe for a healthy dinner. An example of recipe suggestion is available in Figure 8.

3.4.4 News Service. MyrrorBot can also provide information about news by exploiting Google News APIs. In this case, the user can request the latest news about a particular category (e.g., Business, Health, Entertainment, Technology, Science, Sports) or about a particular topic (e.g., news discussing "Coronavirus"). In both the cases the service works in a *retrieval* mode, that is to say, all the users receive non-personalized answers based on their requests.

Figure 9 shows the answer of MyrrorBot for a request of news about *coronavirus COVID-19*. In particular, COVID-19 has been identified as a topic and the most recent news article about that topic is selected and shown to the user.

As for the other services, *news service* also supports *recommendations*. In this case, guest users will receive a popular news, picked among those collected from the top headlines of the day and the most important news of the week. Conversely, personalized recommendations are tailored on the ground of the preferences of the user that are encoded in her *holistic user model*. In particular, the recommendation strategy will look for articles matching the keywords extracted from the HUM of the user and the most popular one is finally presented.

As we already explained for other services, we were required to develop our own personalization strategy, since Google does not make available neither the user profile nor the recommendation algorithm that is used in Google News.

3.4.5 Weather Service. Information about weather is provided by exploiting OpenWeatherMap APIs. Basically, this service can be used to ask information about the weather.

If the current location of the user is available in the *"Demographics"* section of her HUM, then this feature is extracted from the profile and it is used to automatically retrieve weather

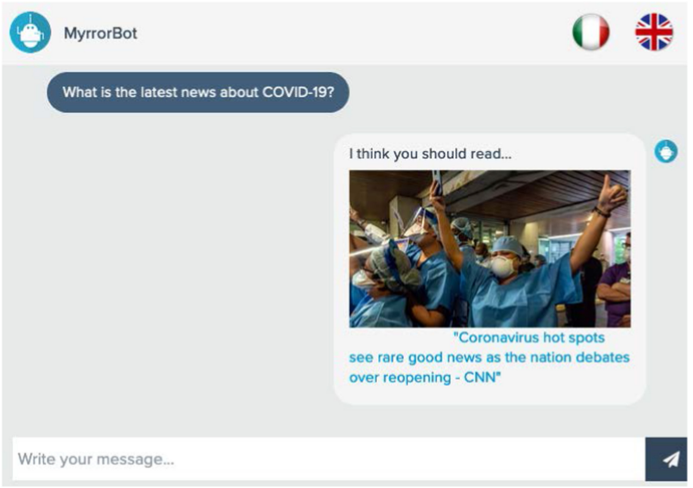

Fig. 9. News retrieval on MyrrorBot.

information regarding the *current* location of the user. Otherwise, the user has to explicitly indicate the city he is interested in.

3.4.6 Workout Service. MyrrorBot can also provide the users with *workout retrieval* and *recommendation*. Similar to what we did for our food recommendation service, we collected data from external services and we developed a completely new algorithm to provide users with personalized workout recommendations.

In particular, our algorithm is based on the data available on Darebee, which are split based on different body parts as well as on different difficulty levels. To summarize, ten different training programs are currently available in MyrrorBot: *cardio, high intensity (hi-it), combat, strength, wellness, stretching, yoga*, and exercises focused on *abdomens, upper body, and lower body parts*.

In *retrieval-based* intents, the user can request a particular workout, while when a *recommendation* is requested, different personal parameters extracted from the *"physical state"* facet of the HUM are acquired to provide user with a personalized suggestion. In particular, recommendations are based on the analysis of heart rate, daily sleep hours and emotional state. Moreover, user height and weight (available in the *"demographics"* facet) are used to assign the user to one of seven weight classes of the BMI classification [De Vriendt et al. 2009].

Generally speaking, workout recommendations exploit a *stereotype-based* strategy that relies on user health state and personal characteristics. As an example, a user who is overweight (high BMI) is assigned to a cardio workout, to allow her to lose her weight. Conversely, users in good shape are assigned to hi-it or strength workouts. Previous physical activity (available in the *"Behavior"* facet) is also exploited to properly tune the difficulty of the training. The higher the activity, the stronger the workouts. In our opinion, stereotype-based personalization can be effective in this particular settings, since they allow to provide users with accurate recommendations even in complete *cold-start* scenarios, that is to say, when no information about previous users' workout is available.

Moreover, BMI is also used in combination with heart rate to better tailor the recommendations. In this case, if the user has a BMI below 16.5 (severe underweight) and a heartbeat between 60 and 100, MyrrorBot recommends a strength training or stretching. Similar rules are applied for the other combination of BMI and heart rate. Similarly, the analysis of sleep regarding the previous night will be used to set the level of difficulty of the training to be recommended. If the user has

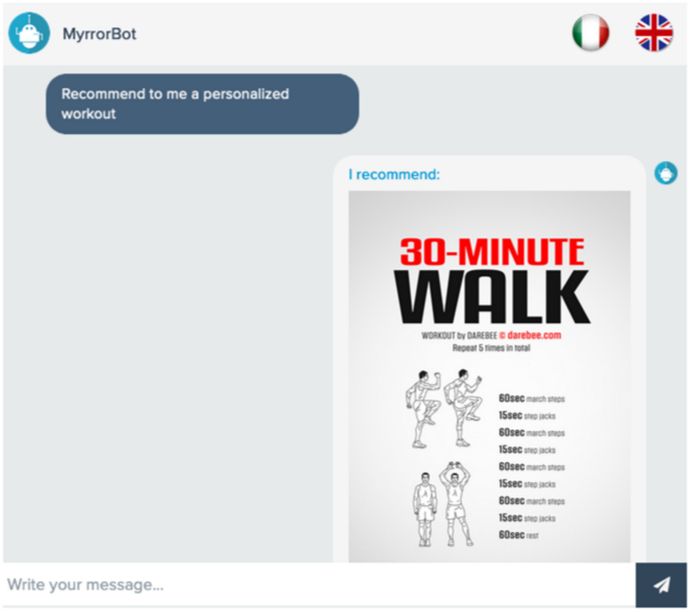

Fig. 10. Example of a request to MyrrorBot about a personalized workout.

not slept at all, then the system will advise the user not to train. Instead, if the user has slept a few hours, then the system will recommend a low difficulty workout to prevent to fatigue him further.

When all these information are not available (e.g., guest users) a workout randomly picked among cardio, wellness, and strength with a medium level of difficulty is selected. An example of the recommendations obtained by exploiting our workout service is provided in Figure 10. As shown in the figure, which always relies on our *running example*, the user is provided with a *cardio workout* recommendation due to its slightly high BMI. Moreover, due to the low amount of physical activity the difficulty level is set to low or medium. Of course, such recommendations are also provided with the *explanation* facility we have previously introduced, to better understand why the system has selected a particular workout.

As a future work, we plan to strengthen this feature by recommending a list of recommendations, that is to say, by arranging a personalized training program (obviously based on user characteristic) rated than a single training.

3.4.7 TV-program Service. TV-program retrieval and recommendation available in MyrrorBot is inspired by the Alexa's skill called *SuperGuidaTv*, which can be activated by pronouncing a sentence like: *"Alexa, what are you doing on TV tonight?"*

Similarly to SuperGuidaTV, *retrieval* intents are available for our TV-program service. In this case, it is possible to filter TV shows by using genres (action, thriller, adventure) and type of TV Programs (movies, TV shows, documentary). Moreover, even more sophisticated requests (e.g., a movie directed or starred by a particular person) are possible.

Moreover, our service extends the above mentioned SuperGuidaTV, since it provides *personalized* suggestion of TV shows. Indeed, if the user is logged into the platform, she can enjoy personalized TV program suggestions based on her mood and her preferences. First, the recommendation strategy will try to match the concepts encoded in the *"Interests"* facets of her HUM to the keywords that describe the TV Show. In this case, we want to emphasize that we designed this

recommendation strategy on our own, since no personalization mechanism is natively available on SuperGuidaTV.

If no match is found, then recommendations are based on user emotional state. In particular, if the user is experiencing sadness, fear, or anger, the system recommends a TV program whose genre is *entertainment, comedy, or satire* type. Conversely, if the user is experiencing joy or surprise or neutrality, the system recommends a random TV program.

4 EXPERIMENTAL EVALUATION

To validate the intuitions and the design choices behind this work, we carried out a user study ($N = 76$). In particular, our experiment was designed to answer to the following research questions:

(1) How effective is MyrrorBot at supporting the users in their daily tasks or at accessing personal information stored in their profiles? *(Research Question 1)*
(2) How accurate and satisfying are the recommendations based on HUMs, which are available in MyrrorBot *(Research Question 2)*
(3) What is the opinion of users regarding the overall usability and ease of use of the system? *(Research Question 3)*

4.1 Experimental Protocol

We designed a user study based on sample of 76 subjects, recruited through the *availability sampling strategy* (*male* = 61.8%, *age below 35* = 84.1%, *good experience with technology* = 85.5%, *previous experience with natural language interfaces and chatbots* = 35.5%).

In particular, we designed a *controlled experiment* run in our research laboratory consisting of the following steps:

(1) *Training:* The first session was run to provide the participants with the skills required to carry out the experiment. In particular, we introduced the concept of PDA, we showed the main functionalities of MyrrorBot, and we explained how to formulate natural language requests. Next, we discussed about the goal of the experiment and we showed how to create a holistic user profile in Myrror. Moreover, we also had a discussion with the users about privacy and trustworthiness of the system, and about how their personal data would have been used. At the end of the training, the users created their HUMs in Myrror;
(2) *Execution:* Next, we defined a set of 24 tasks to be carried out, and we asked the participants to complete them by using MyrrorBot. As for the tasks, we split them as follows: 9 tasks regarded the *retrieval* of information, 8 tasks regarded the *access* to the personal information stored in their HUMs and 7 tasks regarded the request of *recommendations*. Clearly, the tasks are roughly mapped to the *intents* the system can handle. For each task, we registered task completion time. To compare the performance of MyrrorBot with some baselines, we designed the following protocol: as for *retrieval* intents, we also asked the user to submit the same query on Google and we registered task completion time, as well. Conversely, as for *informative* intents, the users had to fulfill the request by interacting with the classic web interface of Myrror. For each user, the system to be used first was randomized. Finally, as for *recommendation* intents, we asked the users to compare the recommendations obtained by exploiting HUMs to the recommendations obtained by using a guest MyrrorBot profile (that is to say, without any form of personalization);
(3) *Evaluation:* Finally, we asked the users to fill in several post-usage questionnaires. The first one, inspired by usability questionnaires as the **System Usability Scale**[8] **(SUS)**, evaluated

[8] https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

Table 6. Overiew of the Metrics and Tools We Adopted to Answer to Each Research Question

Research Questions	Metric / Method	Baseline / Tools
RQ1	Task Completion Time	Searches on Google
RQ2	Quality of Recommendations	Recommendation with Guest Account
RQ3	Post-Usage Questionnaire	ResQUE (for RQ2) UEQ (Usability Questionnaire) AttrakDiff (Usability Questionnaire) SUS (Usability Questionnaire)

users' experience in terms of *learning curve, interaction cost, accuracy of the answers, likelihood of daily usage and likelihood of potential switching*. Next, we evaluated the overall *usability* of the system by using the AttrakDiff Questionnaire [Hassenzahl et al. 2003][9] and the UEQ Questionnaire [Laugwitz et al. 2008]. Finally, we evaluated the quality of the recommendations through the ResQUE questionnaire [Pu et al. 2011]. In all the cases, we chose state-of-the-art questionnaires in the area of usability evaluation and recommendation.

To answer to **Research Question 1 (RQ1)** and **Research Question 2 (RQ2)**, we carried out a *quantitative* analysis based on *objective* metrics such as *average* task completion time and *ratio* of explicit feedbacks we collected on the recommendations. As for task completion time, we registered time from the formulation of the request to the selection or the answer (on Google) or to the sending of the feedback (on MyrrorBot). As for **Research Question 3 (RQ3)**, we averaged over all the users the scores corresponding to the answers we collected for the questionnaires.

As for the selection of the baselines, it is necessary to point out that our system—as most of the chatbots currently available—was designed to support a classic text-based natural language interaction. Given that the current implementation does not support any voice-based interaction, we decided to exclude technologies such as Siri and Google Home from the potential baselines, since a comparison between a text-based and a voice-based PDA would not have been *fair*.

Indeed, it is necessary to recall that voice-based PDAs also exploit a *speech-to-text* component that acquires voice signals and converts them into text. Of course, this further processing requires some extra time and introduces some further *bias*. Unfortunately, this would have made the overall task completion times not comparable. As a consequence, we selected Google searches as a baseline, since they are based on a written text-based interaction and do not require any further processing. Moreover, as for the Internet connection, we can state that it remained *stable* throughout the experiment, so we can assume that each user did not suffer any significant change of the performance of the connection between the first and the second part of the experiment. Accordingly, we can state that our design led to a fair comparison between the different configurations.

To assess whether the gap between MyrrorBot and the baseline was significant, we also run Wilcoxon tests (with $p < 0.05$). In Table 6, we summarize all the tools and metrics we used for our experiment.

4.2 Discussion of the Results

Research Question 1. To investigate RQ1, we measured the average *task completion time* over a set of 17 tasks that cover most of the functionalities of MyrrorBot. To summarize, 9 of these tasks concerned the use of MyrrorBot as a retrieval tool, while the remaining 8 concerned the opportunity of inspecting the information encoded in the HUM of the user. To answer to RQ1, we

[9] http://www.attrakdiff.de/index-en.

compared the time needed to complete a task by interacting with MyrrorBot to the time needed to complete the same task by exploiting a baseline system. As previously stated, Google searches were adopted as baseline for *retrieval* intents, while the web-based Myrror interface was used to compare the completion time on *informative* intent. Result of this experiment are reported in Table 7.

As shown in the Table, the adoption of a natural language interface provides the user *with a significant decrease* in the task completion time. This is not a trivial result, since the users are more used to interact with search engines and web platforms rather than natural language interfaces, thus we did not expect a significant difference in the task completion time. Differently to what expected, gaps were significant ($p < 0.001$, by exploiting Wilcoxon tests), with the exception of the task related to the *weather* and *TV-program* searches. This is not surprising, since modern search engines already provide with some widgets to immediately check weather forecasts and to get a snapshot of the programs on air in a particular day. Conversely, for all the other tasks, we can state that the adoption of a natural language interface leads to a significant decrease of the task completion time.

The gaps we registered range from 5.96% to 39.2% for *retrieval intents* and from 19.05% to 47.2% for *informative intents*. As for retrieval intents, we can state that the highest gaps are noted for tasks related to *complex information needs*, such as searching a particular playlist, searching for a particular recipe and search for a specific training. This finding further confirms the intuition behind this work, since our conjecture was that the adoption of a natural language interface would have been helpful to support the users to fulfill less trivial information needs.

In general, the average gap we registered is equal to 21.41% for *retrieval* and to 40.29% for *informative intents*. This is another interesting result, since it shows that the natural language interface implemented in MyrrorBot allows the user to consult the information encoded in their HUM in about *half of the time*. This supports the ideas underlying *queryable user modeling*, since it shows that natural language interfaces can optimize and improve the access to personal data, thus tackling in an effective way the well-known *data explosion* problem.

Research Question 2. Next, as for RQ2, we carried out a *quantitative* analysis by collecting the feedbacks of the users after they received the recommendations. As previously stated, we asked the users to: (i) formulate a *recommendations* request (e.g., suggest me a song, suggest me a recipe for tonight, etc.) for each for the recommendation intents supported by the platform; (ii) provide an explicit binary feedback to evaluate the suggestion.

For this part of the experiment, we adopted a *between-subject* experimental design; that is to say, half of the sample received recommendations based on their HUM, while the others received non-personalized recommendations, by simulating the scenario in which a user is not known. In this way, we investigated the effectiveness of our personalization strategies and recommendation mechanisms based on HUMs. Of course, the users were not aware of the configurations they were assigned to.

In this case, we need to point out that we did not exploit any other baseline. Indeed, as previously explained, the personalization strategies natively encoded in some of the services (e.g., recommendations in Spotity or YouTube) were not available and could not be invoked by MyrrorBot. Accordingly, we chose a *guest* account as a unique baseline. Moreover, we want to emphasize that the exploitation of such a baseline provides us with a realistic experimental setting, since the PDAs currently available do not include personalization or profiling mechanisms, thus a setting based on a *guest* account replicates the interaction that concretely happens with most of the PDAs. The results of this experiment are summarized in Table 8.

Table 7. Task Completion Time for 17 Tasks

Service	Task	Baseline	MyrrorBot	Gap
Music	*Find a song by U2*	11.95	**10.52**	−11.96% $p < 0.001$
	Find a playlist for a party	16.37	**11.55**	−29.44% $p < 0.001$
Video	*Search for a video about Juventus*	14.66	**12.64**	−12.06% $p < 0.001$
News	*Search for an article about technology*	16.68	**12.17**	−27.03% $p < 0.001$
Food	Search for a lactose-free recipe	17.14	**13.22**	−22.87% $p < 0.001$
	Search for a second course with chicken	21.32	**13.75**	−35.51% $p < 0.001$
Weather	Ask for the weather in Bari	13.41	**12.61**	−5.96% $p > 0.05$
Workout	*Search for a Yoga training*	12.79	**7.76**	−39.32% $p > 0.001$
TV-Program	*Search for a TV Program for tonight*	10.20	**9.32**	−8.62% $p > 0.05$
Demographics	*Check your birthday*	10.31	**7.72**	−25.12% $p < 0.001$
Interests	*Check your interests*	22.11	**9.62**	−56.49% $p < 0.001$
Affects	*Check your latest mood*	20.02	**10.34**	−48.35% $p < 0.001$
Psychological Aspects	*Check your personality traits*	12.33	**9.98**	−19.1% $p < 0.05$
Behaviors	*Check your daily activities*	24.99	**13.25**	−47.2% $p < 0.001$
	Check if you burnt enough calories	23.17	**14.18**	−19.05% $p < 0.05$
Health Data	*Check if you slept enough*	19.20	**13.44**	−30.47% $p < 0.001$
	Check your heart rate	16.49	**10.20**	−38.14% $p < 0.001$

The double-line separates *retrieval* tasks from *informative* tasks. Time is reported in second.

As shown in the Table, the analysis of the feedbacks showed that our personalization and recommendation strategies based on HUMs provide the users with accurate and satisfying suggestions. Indeed, the amount of positive feedbacks we collected significantly overcomes those obtained by the suggestions generated when the user is not known.

It should be pointed out that the gaps are particularly large when the recommendation strategy is based on a richer set of personal information, as it happens for *workout* recommendation and *music* recommendation. Moreover, even if the personalization mechanisms we implemented in MyrrorBot are quite simple, we want to emphasize the importance of this outcome: first, this

Table 8. Percentage of Positive Feedbacks: HUM-based Recommendations vs. Guest

Task	HUM	Guest
Music Recommendation	**84.2%**	65.7%
Video Recommendation	**73.6%**	60.5%
News Recommendation	**81.5%**	68.4%
Food Recommendation	**77.6%**	71.0%
Weather Information	**100%**	61.8%
Workout Recommendation	**92.1%**	63.1%
TV-Program Recommendation	**89.4%**	86.8%

distinguishes our work from several *commercial* solutions that employ very basic personalization strategies (or do not employ them at all). Furthermore, *popularity-based* baselines (as those used to provide guest users with recommendations) often obtain good results in several scenarios, as shown in Cremonesi et al. [2010], and are usually hard to beat [Trattner et al. 2018], thus the superiority of a personalized recommendation strategy is not trivial.

Moreover, this experiment provided us with two further interesting findings: first, it showed the effectiveness of a personalization strategy for based on HUMs, that avoids the time-consuming task of explicit preference acquisition. This is an important outcome, since it allows to exploit recommendation algorithms in complete *cold-start* scenarios. This is particularly common for PDAs, since the users are typically not willing to provide their preferences in a broad range of different domains and services. Moreover, our work showed that the large pool of information available in a HUM (mood, personality, physical activity, etc.) leads to accurate recommendations. The positive effect of features different by interests and preferences on the overall accuracy of the recommendations—especially when they are obtained through an implicit user modeling strategy—is a poorly investigated research line, so we can state that our work provided a novel contribution in the area.

Of course, we are aware that several sophisticated strategies to provide users with recommendations exist. However, the analysis of these algorithms is outside the scope of the current article and it is left as future work. For our goals, the current findings already support our hypothesis concerning the effectiveness of *holistic user models* to provide users with personalized access to online services.

Research Question 3. After the *quantitative* analysis we have described in the previous paragraphs, we also carried out a *qualitative* analysis based on well-known state-of-the-art questionnaires to evaluate the quality of the suggestions generated through MyrrorBot and the overall usability of the system.

As a first step, we asked the users to answer to a set of five questions borrowed from the popular ResQUE questionnaire [Pu et al. 2011].[10] Each question was answered by using a five-point likert scale. The goal of these questions was to assess the ability of the recommender to match the interests of the user, to allow her to discover new products, to understand why the items were recommender and so on. The results are reported in Table 9. The percentage scores represent the average number of *"completely agree"* and *"agree"* answers.

[10]By referring to the short ResQUE questionnaire, we only investigated questions 1, 2, 10, 12, 14—reference to the questionnaire at https://hci.epfl.ch/wp-content/uploads/resque_short.pdf.

Table 9. Usability Evaluation of the System Inspired by ResQUE Questionnaire

Question	HUM	Guest
"The items recommended to me match my interests"	**81.5%**	63.1%
"The recommender systems helped me discover new products"	**68.4%**	60.5%
"I understood why the items were recommended to me"	**78.9%**	71.0%
"Overall, I am satisfied with the recommender"	**94.7%**	68.4%
"I will use this recommender again"	**77.6%**	60.5%

Scores represented the sum of "Agree" and "Completely Agree" answers to the questions.

As shown in the Table, this part of our experiment confirmed the outcomes already emerged for RQ2, since the users were more satisfied by the recommendations generated through our personalization strategy based on HUMs. In particular, larger gaps were noted for the overall satisfaction and the precision of the recommendations. Conversely, a smaller gap was noted for *novelty* and *explainability-related* questions (Questions 2 and 3, respectively). This was someway expected, since our strategies did not devote particular attention to the generation of novel or explainable recommendations. However, beyond this aspect, we can state that the answers we collected further confirmed the effectiveness of the recommendation strategy we implemented in MyrrorBot.

Next, we asked the users to answers to a set of five questions regarding the overall usability of the system. Such questions are inspired by those included in the SUS Usability Questionnaire [Brooke et al. 1996]. Table 10 summarizes the results.[11]

As shown in Table, that shows the distribution of the answers obtained by MyrrorBot and the original Myrror interaction style, the users were in general very satisfied with their experience with the new natural language interface, since almost 90% of the participants stated that they received the answers they were looking for. As for the *learning curve* and the *interaction method*, a large majority of the sample (around 90%) confirmed the ease of use and the ease of interaction. Given that only 35.5% of the participants had previous experience with chatbots and PDAs, this is a particularly interesting outcome that confirms how *natural language interfaces* can be used to provide users with a simpler and more natural interaction with web platforms and technology in general. Finally, we obtained interesting answers also for the likelihood of *daily usage* and likelihood of potential *switching*, since a large part of the sample confirmed that they would use this system (by even replacing the classical platforms they usually interact with) on a daily basis.

To further stress this finding, we can compare the results obtained by MyrrorBot with those obtained by Myrror. As shown in the last column of Table 10, MyrrorBot obtained a higher score for all the metrics, with a particularly relevant gap in terms of *Interaction* and *Learning Curve*. This is another encouraging finding, that further confirms the goodness of the design and the implementation of MyrrorBot as well as the effectiveness of chatbots and natural language interfaces to improve the way people access online services and information.

Finally, to further strengthen these outcomes, we also asked users to answer to the AttrakDiff [Hassenzahl et al. 2003] and UEQ Questionnaires [Laugwitz et al. 2008]. Both the questionnaires share a similar intuition, since they are based on a set of couples of *antithetical* concepts the users have to evaluate, by identifying the concept that is closer to her perception of the system. Such couples are then mapped to a set of *usability dimensions* that allow to obtain a snapshot of how the users perceive the system.

[11]The Switching metric was not calculated for Myrror, since the question refers to the comparison between a PDA and a web platform. The questions did not make sense, since Myrror was *already* a web platform, thus it has no terms of comparison.

Table 10. Usability Evaluation of the System Inspired by SUS

Metrics	Question	MyrrorBot			Myrror			Gap
		% Comp. Agree	% Agree	% Others	% Comp. Agree	% Agree	% Others	
Learning Curve	*"I became familiar with the system very quickly."*	**51.3%**	38.2%	10.5%	35.9%	**44.2%**	19.9%	**+9.4%**
Interaction	*"It was easy to communicate with the system and to formulate my requests."*	40.8%	**48.7%**	11.8%	29.4%	**43.1%**	27,5%	**+15.7%**
Accuracy	*"The system provided me with the answers I needed."*	**55.3%**	35.5%	9.2%	**55.0%**	31.9%	13.1%	**+3.9%**
Daily Use	*"I would use the system for my daily tasks."*	28.9%	**40.8%**	29.2%	25.4%	**40.1%**	34.5%	**+5.3%**
Switching	*"I would use the system instead of a web platform"*	**40.8%**	**40.8%**	18.4%	n.a.			

"Agree" and "Completely Agree" count the answers equal to 4 and 5 of 5. "Others" aggregates the remaining answers. Gap refers to the difference of the sums of "Completely Agree" and "Agree" answers between MyrrorBot and Myrror.

As for AttrakDiff, 28 different couples are used, while the number of couples used by UEQ is set to 26. Due to space reasons, in this article, we just present the results we obtained through AttrakDiff. This choice is also motivated by the similar outcomes we got from the questionnaire in terms of overall usability.[12] The results we obtained from the questionnaire are reported in Figure 11, that summarizes the average score provided by the users for each couple of concepts.

It should be pointed out that we asked the users to compile the questionnaire twice. The second round was used to evaluate the baseline systems we used throughout the experiments, as well. As shown in Figure 11, MyrrorBot obtained an higher average score for all the couples that compose AttrakDiff questionnaire. The higher gaps were noted for the couples *alienating-integrating* and *unpleasant-pleasant*. This shows that the idea of designing a comprehensive tool that allows the users to interact with several different services was correctly perceived by our experimental sample. Moreover, we can note that the lower results were obtained for the couple *undemanding-challenging*. This is an expected result, since the interaction with such a system requires a good knowledge of conversational systems and this skill is not trivial. The low results obtained by the couple *technical-human* further confirms this issue.

Next, all the couples were mapped to four usability dimensions, that is to say, *pragmatic quality*, *hedonic quality-identity*, *hedonic quality-stimulation*, *attractiveness*. Such dimensions represent the overall usability of the product, its ability to stimulate the identification of the users in the product and their interest in terms of content and presentation, and its overall perception of the quality, respectively. In this case, the average results obtained by each dimension are reported in Figure 12. In this case, we can note again that our system overcame both the baselines. Regardless of the behavior of competing systems, it should be pointed out that we obtained encouraging results for all the dimensions, and the higher score was obtained for the *attractiveness*. This confirms the goodness of the design choices behind MyrrorBot, since it shows that the quality of the system was correctly perceived by the users.

5 CONCLUSIONS AND FUTURE WORK

In this article, we presented MyrrorBot, a personal digital assistant that provides users with personalized access to online services, such as music, video, news, food recommendations, and so on. Moreover, the system also provides the users with the opportunity of *querying* their own user models in natural language, to inspect the features encoded in the different facets of their profiles.

[12]If requested, then the results obtained through UEQ can be reported in the article or in an Appendix.

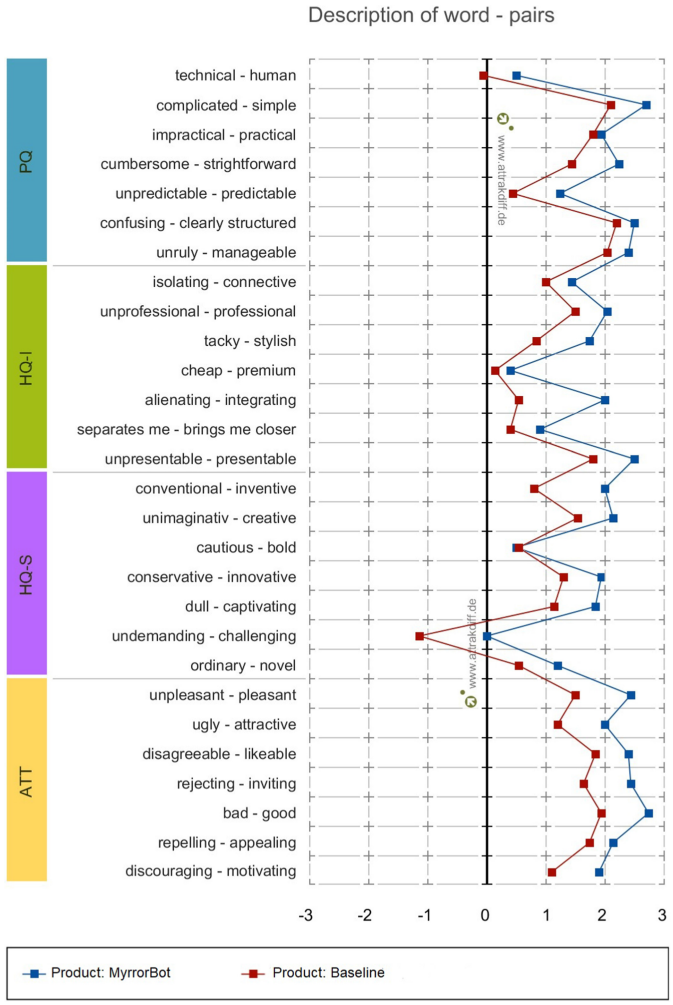

Fig. 11. Evaluation of word pairs, based on AttrakDiff model.

The architecture we propose is based on an *intent recognizer* that exploits Google DialogFlow to automatically interpret users' requests. Such requests are then forwarded to external services that support the users in fulfilling their information needs.

In the experimental evaluation, we investigated users' acceptance of this new interaction style and the results showed that people were generally satisfied with the system. Moreover, the experiments showed a significant gap in *task completion time* and *recommendation accuracy* when natural language interfaces and holistic user models were adopted, and this finally confirmed the effectiveness of the proposed method.

As future work, we will first extend the set of the available services to support the users in a broader range of tasks. Next, we will also improve the architecture of the system by including a *dialog state tracker* that keeps track of the requests of the user during a single session, to make the interaction even more effective and satisfying. Finally, we will integrate more recommendation algorithms based on user-generated content [Lops et al. 2009; Musto et al. 2009], and we will carry

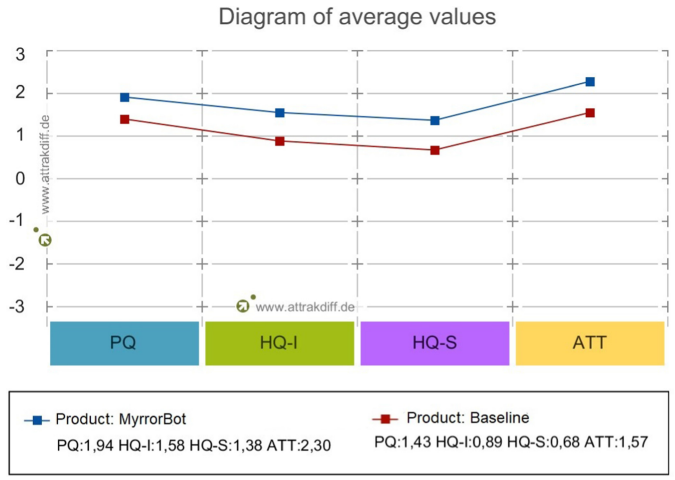

Fig. 12. Average score of the different aspects of the AttrakDiff model.

out new experiments, to evaluate whether the effectiveness of the personalization strategy differs on varying of the algorithm that is used to generate the suggestions.

REFERENCES

Jae-wook Ahn, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Yeon Syn. 2007. Open user profiles for adaptive news systems: Help or harm? In *Proceedings of the 16th international conference on World Wide Web*. 11–20.

Mark Assad, David J. Carmichael, Judy Kay, and Bob Kummerfeld. 2007. PersonisAD: Distributed, active, scrutable model framework for context-aware services. In *Proceedings of the International Conference on Pervasive Computing*. Springer, 55–72.

Fedor Bakalov, Birgitta König-Ries, Andreas Nauerz, and Martin Welsch. 2010. IntrospectiveViews: An interface for scrutinizing semantic user models. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*. Springer, 219–230.

Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 265–274.

Pierpaolo Basile and Nicole Novielli. 2014. UNIBA at EVALITA 2014-SENTIPOLC Task: Predicting tweet sentiment polarity combining micro-blogging, lexicon and semantic features. *Proceedings of the International Workshop on Evaluation of Natural Language and Speech Tools for Italian (EVALITA'14)*. 58–63.

Antoine Bordes, Y.-Lan Boureau, and Jason Weston. 2016. Learning End-to-end Goal-oriented Dialog. Retrieved from http://arxiv.org/abs/1605.07683.

John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7.

Susan Bull and Judy Kay. 2010. Open learner models. In *Advances in Intelligent Tutoring Systems*. Springer, 301–322.

Thorsten Büring and Harald Reiterer. 2005. Zuiscat: Querying and visualizing information spaces on personal digital assistants. In *Proceedings of the 7th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 129–136.

Federica Cena, Silvia Likavec, and Amon Rapp. 2018. Real world user model: Evolution of user modeling triggered by advances in wearable and ubiquitous computing. *Info. Syst. Front.* 21, 5 (2018), 1–26.

Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems*. 39–46.

Marco De Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. 2015. Semantics-aware content-based recommender systems. In *Recommender Systems Handbook*. Springer, 119–159.

Tineke De Vriendt, Inge Huybrechts, Charlene Ottevaere, Inge Van Trimpont, and Stefaan De Henauw. 2009. Validity of self-reported weight and height of adolescents, its impact on classification into BMI-categories and the association with weighing behaviour. *Int. J. Environ. Res. Public Health* 6, 10 (2009), 2696–2711.

Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, et al. 2013. Recent advances in deep learning for speech research at Microsoft. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 8604–8608.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke Van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Info. Process. Manage.* 51, 2 (2015), 32–49.

Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. 2015. Evaluating prerequisite qualities for learning end-to-end dialog systems. Retrieved from https://arXiv:1511.06931.

Yuxiao Dong, Nitesh V. Chawla, Jie Tang, Yang Yang, and Yang Yang. 2017. User modeling on demographic attributes in big mobile social networks. *ACM Trans. Info. Syst.* 35, 4, Article 35 (July 2017), 33 pages. https://doi.org/10.1145/3057278

Khalid El-Arini, Ulrich Paquet, Ralf Herbrich, Jurgen Van Gael, and Blaise Agüera y Arcas. 2012. Transparent user models for personalization. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 678–686.

Gerhard Friedrich and Markus Zanker. 2011. A taxonomy for generating explanations in recommender systems. *AI Mag.* 32, 3 (2011), 90–98.

Debjyoti Ghosh, Pin Sym Foong, Shan Zhang, and Shengdong Zhao. 2018. Assessing the utility of the system usability scale for evaluating voice-based user interfaces. In *Proceedings of the 6th International Symposium of Chinese (CHI'18)*. 11–15.

Giorgio Gianforme, Sergio Miranda, Francesco Orciuoli, and Stefano Paolozzi. 2009. From classic user modeling to scrutable user modeling. In *Proceedings of the Conference on Ontology for e-Technologies (OET'09)*. 23–32.

Lewis R Goldberg. 1993. The structure of phenotypic personality traits. *Amer. Psychol.* 48, 1 (1993), 26.

Julio Guerra-Hollstein, Jordan Barria-Pineda, Christian D. Schunn, Susan Bull, and Peter Brusilovsky. 2017. Fine-grained open learner models: Complexity versus support. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. 41–49.

Marc Hassenzahl, Michael Burmester, and Franz Koller. 2003. AttrakDiff: A questionnaire to measure perceived hedonic and pragmatic quality. In *Mensch & Computer*, Vol. 57. 187–196.

Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta von Wilamowitz-Moellendorff. 2005. Gumo—The general user model ontology. In *Proceedings of the International Conference on User Modeling*. Springer, 428–432.

Robert Hogan. 1969. Development of an empathy scale. *J. Consult. Clin. Psychol.* 33, 3 (1969), 307.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Trans. Info. Syst.* 38, 3, Article 21 (Apr. 2020), 32 pages. https://doi.org/10.1145/3383123

Andrea Iovine, Fedelucio Narducci, and Marco de Gemmis. 2019. A dataset of real dialogues for conversational recommender systems. In *Proceedings of the 6th Italian Conference on Computational Linguistics*. Retrieved from http://ceur-ws.org/Vol-2481/paper37.pdf.

Andrea Iovine, Fedelucio Narducci, and Giovanni Semeraro. 2020. Conversational recommender systems and natural language: A study through the converse framework. *Decision Supp. Syst.* 131, 3 (2020), 113250. https://doi.org/10.1016/j.dss.2020.113250

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2020. A survey on conversational recommender systems. Retrieved from https://arXiv:2004.00646.

Yucheng Jin, Wanling Cai, Li Chen, Nyi Nyi Htun, and Katrien Verbert. 2019. MusicBot: Evaluating critiquing-based music recommenders with conversational interaction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 951–960.

Judy Kay, Bob Kummerfeld, and Piers Lauder. 2002. Personis: A server for user models. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*. Springer, 203–212.

Judy Kay and Andrew Lum. 2005. Exploiting readily available web data for scrutable student models. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED'05)*. 338–345.

Alfred Kobsa. 2001. Generic user modeling systems. *User Model. User-adapt. Interact.* 11, 1-2 (2001), 49–63.

Demetris Kyriacou. 2008. A scrutable user modelling infrastructure for enabling life-long user modelling. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer, 421–425.

Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. 2008. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*. 208–211.

Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and evaluation of a user experience questionnaire. In *Proceedings of the Symposium of the Austrian HCI and Usability Engineering Group*. Springer, 63–76.

Danh Le-Phuoc, Anh Le-Tuan, Gregor Schiele, and Manfred Hauswirth. 2014. Querying heterogeneous personal information on the go. In *Proceedings of the International Semantic Web Conference*. Springer, 454–469.

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS'18)*. 17.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. Retrieved from https://arXiv:1703.01008.

Bing Liu and Ian Lane. 2017. An End-to-End Trainable Neural Network Model with Belief Tracking for Task-oriented Dialog. In *Proceedings of Interspeech'17*. 2506–2510. https://doi.org/10.21437/Interspeech.2017-1326 arXiv: 1708.05956.

Pasquale Lops, Marco de Gemmis, Giovanni Semeraro, Cataldo Musto, Fedelucio Narducci, and Massimo Bux. 2009. A semantic content-based recommender system integrating folksonomies for personalized access. In *Studies in Computational Intelligence*, vol. 229 (2009), 27.

Tariq Mahmood and Francesco Ricci. 2009. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*. 73–82.

Judith Masthoff, Floriana Grasso, and Jaap Ham. 2014. Preface to the special issue on personalization and behavior change. *User Model. User-Adapt. Interact.* 24, 5 (2014), 345–350.

Cataldo Musto, Fedelucio Narducci, Marco De Gemmis, Pasquale Lops, and Giovanni Semeraro. 2009. STaR: A social tag recommender system. In *Proceedings of the International Conference on ECML PKDD Discovery Challenge*. 215–227.

Cataldo Musto, Fedelucio Narducci, Marco Polignano, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2020a. Towards queryable user profiles: Introducing conversational agents in a platform for holistic user modeling. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP'20)*, Tsvi Kuflik, Ilaria Torre, Robin Burke, and Cristina Gena (Eds.). ACM, 213–218. https://doi.org/10.1145/3386392.3399298

Cataldo Musto, Marco Polignano, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2020b. Myrror: A platform for holistic user modeling. *User Model. User-Adapt. Interact.* 30, 3 (2020), 477–511.

Cataldo Musto, Christoph Trattner, Alain Starke, and Giovanni Semeraro. 2020c. Towards a knowledge-aware food recommender system exploiting holistic user models. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 333–337.

Fedelucio Narducci, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2019. An investigation on the user interaction modes of conversational recommender systems for the music domain. *User Model. User-Adapt. Interact.* 30 (2019), 1–34. https://doi.org/10.1007/s11257-019-09250-7

Fedelucio Narducci, Cataldo Musto, Giovanni Semeraro, Pasquale Lops, and Marco De Gemmis. 2013. Leveraging encyclopedic knowledge for transparent and serendipitous user profiles. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*. Springer, 350–352.

Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. 2019. A model of social explanations for a conversational movie recommendation system. In *Proceedings of the 7th International Conference on Human-Agent Interaction*. 135–143.

Till Plumbaum, Songxuan Wu, Ernesto William De Luca, and Sahin Albayrak. 2011. User modeling for the social semantic web. In *Proceedings of the Second Workshop on Semantic Personalized Information Management: Retrieval and Recommendation (SPIM'11), CEUR-WS*, volume 781. 78–89.

Marco Polignano, Pierpaolo Basile, Gaetano Rossiello, Marco de Gemmis, and Giovanni Semeraro. 2017. Learning inclination to empathy from social media footprints. In *Proceedings of the 25th Conference on User Modeling, Adaptation, and Personalization*. ACM, 383–384.

Marco Polignano, Basile Pierpaolo, Marco De Gemmis, and Giovanni Semeraro. 2018. An Emotion-driven Approach for Aspect-based Opinion Mining. In *Proceedings of the 9th Italian Information Retrieval Workshop*, Nicola Tonellotto, Luca Becchetti, and Marko Tkalčič (Eds.), Vol. 2140. Retrieved from http://ceur-ws.org/Vol-2140/.

Pearl Pu, Li Chen, and Rong Hu. 2011. A user-centric evaluation framework for recommender systems. In *Proceedings of the 5th ACM Conference on Recommender Systems*. 157–164.

Dimitrios Rafailidis. 2018. The technological gap between virtual assistants and recommendation systems. Retrieved from https://arXiv:1901.00431.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Quoc Viet Hung Nguyen, Zi Huang, and Xiangliang Zhang. 2020. CRSAL: Conversational recommender systems with adversarial learning. *ACM Trans. Info. Syst.* 38, 4 (2020), 1–40. https://doi.org/10.1145/3394592

Elaine Rich. 1979. *Cogn. Sci.* 3, 4 (1979), 329–354.

Alexander I. Rudnicky, Eric H. Thayer, Paul C. Constantinides, Chris Tchou, R. Shern, Kevin A. Lenzo, W. Xu, and Alice Oh. 1999. Creating natural dialogs in the carnegie mellon communicator system. In *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH'99)*. Retrieved from http://www.isca-speech.org/archive/eurospeech_1999/e99_1531.html.

Ruhi Sarikaya. 2015. The technology powering personal digital assistants. In *Proceedings of the 16th Annual Conference of the International Speech Communication Association*.

Ruhi Sarikaya, Paul A. Crook, Alex Marin, Minwoo Jeong, Jean-Philippe Robichaud, Asli Celikyilmaz, Young-Bum Kim, Alexandre Rochette, Omar Zia Khan, Xiaohu Liu, et al. 2016. An overview of end-to-end language understanding and

dialog management for personal digital assistants. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT'16)*. IEEE, 391–397.

Bahar Sateli, Felicitas Löffler, Birgitta König-Ries, and René Witte. 2016. Semantic user profiles: Learning scholars' competences by analyzing their publications. In *Proceedings of the International Workshop on Semantic, Analytics, Visualization*. Springer, 113–130.

Melanie Swan. 2013. The quantified self: Fundamental disruption in big data science and biological discovery. *Big Data* 1, 2 (2013), 85–99.

Christoph Trattner, Dominik Moesslang, and David Elsweiler. 2018. On the predictability of the popularity of online recipes. *EPJ Data Sci.* 7, 1 (2018).

Hoai Phuoc Truong, Prasanna Parthasarathi, and Joelle Pineau. 2017. Maca: A modular architecture for conversational agents. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. 93–102.

Chun-Hua Tsai and Peter Brusilovsky. 2017. Providing control and transparency in a social recommender system for academic conferences. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. 313–317.

Daisuke Tsumita and Tomohiro Takagi. 2019. Dialogue based recommender system that flexibly mixes utterances and recommendations. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'19)*. IEEE, 51–58.

Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide*, 1st ed. Springer International Publishing, Cham.

Zhuoran Wang, Hongliang Chen, Guanchun Wang, Hao Tian, Hua Wu, and Haifeng Wang. 2014. Policy learning for domain selection in an extensible multi-domain spoken dialogue system. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 57–67.

Rainer Wasinger, James Wallbank, Luiz Pizzato, Judy Kay, Bob Kummerfeld, Matthias Böhmer, and Antonio Krüger. 2013. Scrutable user models and personalised item recommendation in mobile lifestyle applications. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*. Springer, 77–88.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. Retrieved from https://arXiv:1604.04562.

Chirayu Wongchokprasitti and Peter Brusilovsky. 2007. Newsme: A case study for adaptive news systems with open user model. In *Proceedings of the 3rd International Conference on Autonomic and Autonomous Systems (ICAS'07)*. IEEE, 69–69.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. Retrieved from http://arxiv.org/abs/1606.02560.

Received May 2020; revised December 2020; accepted January 2021