DEVELOPING ENHANCED CONVERSATIONAL AGENTS FOR SOCIAL VIRTUAL WORLDS

ACCEPTED MANUSCRIPT

David Griol, Araceli Sanchis, José Manuel Molina, Zoraida Callejas

Software Engineering Department, University of Granada, Spain
{dgriol, zoraida}@ugr.es
Universidad Carlos III de Madrid, Computer Science Dept., Avda. de la Universidad 30, 28911 Leganés, Spain
masm@inf.uc3m.es, molina@ia.uc3m.es

Developing Enhanced Conversational Agents for Social Virtual Worlds

David Griol[a], Araceli Sanchis[a], José Manuel Molina[a], Zoraida Callejas[b]

[a]*Dept. of Computer Science*
Carlos III University of Madrid, Spain.
{*david.griol,araceli.sanchis,josemanuel.molina*}*@uc3m.es*
[b]*Dept. of Languages and Computer Systems,*
University of Granada, Spain.
zoraida@ugr.es

Abstract

In this paper, we present a methodology for the development of embodied conversational agents for social virtual worlds. The agents provide multimodal communication with their users in which speech interaction is included. Our proposal combines different techniques related to Artificial Intelligence, Natural Language Processing, Affective Computing, and User Modeling. A statistical methodology has been developed to model the system conversational behavior, which is learned from an initial corpus and improved with the knowledge acquired from the successive interactions. In addition, the selection of the next system response is adapted considering information stored into user's profiles and also the emotional contents detected in the user's utterances. Our proposal has been evaluated with the successful development of an embodied conversational agent which has been placed in the Second Life social virtual world. The avatar includes the different models and interacts with the users who inhabit the virtual world in order to provide academic information. The experimental results show that the agent's conversational behavior adapts successfully to the specific characteristics of users interacting in such environments.

Keywords: Conversational Interfaces, Speech Interaction, Statistical Dialog Management, User Modeling, Social Networks, Virtual Worlds, Second Life, Affective Computing

1. Introduction

Social Networking is a global consumer phenomenon [1, 2, 3]. The increase in the amount of time people are spending on these sites is changing the way people spend their time online and creating new ways of communication and cooperation that impact how people interact within their normal daily lives. The advance of social networking has entailed a considerable progress in the development of virtual worlds or “metaverses” [4, 5, 6, 7], in which humans, through their avatars [8], “cohabit” with other users[1]. Some interesting statistics about these environments show that, for instance, a total of 57 million accounts have been created in the Second Life virtual world since 2003. In 2018, there is an average number of 350,000 new registrations in this virtual world monthly from about 200 countries.

Traditionally, these computer-simulated graphic environments have had a predefined structure and fixed tasks that the user could carry out. However, social virtual worlds have emerged to emphasize the role of social interaction in these environments, allowing the users to determine their own experiences [9, 10, 11, 12]. The rising of massively multiplayer online real-life games (MMORLGs), also called virtual social worlds, their variety of uses for research and educational goals [13, 14], the emergence of open source virtual worlds (e.g., OpenSimulator), and their combination with virtual reality simulation and compatible and immersion devices (e.g., High Fidelity, Sansar, Sumerian, and VRChat) have extended the possibilities of these environments for not only creating our own avatars, but also create and shape the world they are in, import 3D assets and modify the world.

Intelligent agents can play a relevant role in this social context, as they allow to provide automated services under the same appearance as that of the human-driven avatars and engaging in more natural interactions. On the one hand, having the same appearance and capabilities of the avatars for human users, intensifies the perception of the virtual world, providing gestures, glances, facial expressions and movements necessary for the communication process [15, 16, 17]. On the other hand, more natural interfaces have the potential to boost the social potential of virtual worlds, making them more attractive for institutions, companies and researchers interested in human-machine communication. However, social interaction in virtual worlds is usually carried out using only text by means of chat-type services. In order to enhance communication in these environments, we propose the integration of dialog systems to develop conversational agents with the ability of oral communication and, at the same time, which benefit from the visual

[1] There is a very interesting timeline with references to the main virtual worlds in `http://www.dipity.com/WebHistoryProject/Virtual_Worlds`, and an archive of virtual worlds in `http://www.archive.org/details/virtual_worlds`

modalities provided by virtual worlds.

A conversational agent [18, 19] can be defined as an automatic system capable of emulating a human being in a dialog with another person, in order to complete a specific task. They are able to understand the user and decide what to respond, but, unlike chatbots and other messaging alternatives that have gained popularity and are the de-facto standard in virtual worlds, they must also conduct a multi-turn conversation beyond simple voice commands or question answering [20].

One of the core aspects of developing conversational interfaces for virtual environments is to design flexible and user's adapted dialog strategies [18, 21]. The dialog strategy defines the system conversational behavior in response to user utterances and environmental states that, for example, can be based on observed or inferred events or beliefs. This design is usually carried out in industry by hand-crafting dialog strategies tightly coupled to the application domain in order to optimize the behavior of the dialog system in that context [22]. However, it is a very time-consuming process and has the disadvantage of lack of portability and adaptation to new contexts and application domains [21, 23].

This has motivated the research community to find ways for automating dialog learning and user-adapted interaction using statistical models trained with real conversations [23, 24]. Statistical approaches can model the variability in user behaviors and allow exploring a wider range of strategies. Although the construction and parametrization of the model depends on expert knowledge of the task, the final objective is to develop conversational interfaces that have a more robust behavior, better portability, and are easier to adapt to different user profiles or tasks [25].

Our contribution focuses on four key points. Firstly, since it is very difficult to find studies in the literature that describe the integration of Speech Technologies and Natural Language Processing in virtual worlds, to show that this integration is possible. Secondly, we propose a methodology for learning a statistical model that represents the agents' conversational behavior. This methodology is based on a classification procedure that considers the previous history of the dialog to select the next system response. The new system utterance, which is represented in terms of dialog acts, is selected using the probability distribution provided by a neural network. Additionally, as the methodology is based on statistical methods, it can be employed to facilitate the generation of a dialog model for new tasks, thus it is plausible to use our technique to generate many metabots[2] with different conversational behavior and which are able to maintain a conversation in different

[2] In the virtual worlds context virtual agents are usually addressed to as *metabots*, term coined from the contraction of the terms metaverse and robot.

application domains. Thirdly, we propose to consider information stored in user profiles and the emotional content extracted from the user's utterances for the dialog manager to select user's adapted system responses. Finally, the proposed methodology has been employed to generate the Demic conversational metabot, which provides academic information in the Second Life (http://secondlife.com/) and OpenSimulator (http://opensimulator.org) virtual worlds. A set of measures have been defined to evaluate the performance of the dialog management methodology and the selection of user' adapted responses.

The remaining of the paper is organized as follows. Section 2 briefly describes the related work regarding the interaction in virtual worlds and the design of conversational interfaces. This section also describes the main characteristics of the Second Life and OpenSimulator virtual worlds. Section 3 presents our methodology for developing user's adapted conversational metabots. Section 4 describes the application of our proposal to develop a conversational metabot that provides academic information and Section 5 presents the results of its evaluation. Concluding remarks and directions for future work follow in Section 6.

2. Related work

Virtual words provide a combination of simulation tools, sense of immersion and opportunities for communication and collaboration that have a great potential [15, 7, 26]. In addition, the total number of users of active virtual worlds and virtual reality is forecast to reach 171 million by 2018, and the revenues from this market forecast to increase by over three thousand percent in four years [27, 28, 29]. According to these studies, in 2017 there are more than 500 active virtual worlds (e.g., Second Life, Active Worlds, Multiverse, Kaneva, There, Club Penguin, Dofus, Gaia, etc.), that can be classified into game-orientated or social-orientated. Social-oriented virtual worlds, such as Second Life, mimic real life experiences and augment users' real-life without tasks or objectives that are determined by the platform, and no temporal cycles with beginnings and ends that are typical of many games.

Mikropoulos and Natsis [30] presented a ten-year review on the applications of virtual reality covering more than 50 research studies, and have pointed out that, although virtual worlds support multisensory interaction channels, visual representations predominate. Unfortunately, there are a number of barriers that limit user interaction with computers when interfaces are only visual, as the users must have at least a minimum training for using the devices (mouse and keyboard) and the access is very difficult for people with visual or motor impairments.

In order to address these limitations, an alternative is to use conversational interfaces, which are designed to engage users in a conversation that aims to be as similar as possible as that between humans [18]. Speech offers a greater speed for transmitting information, allows carrying out simultaneous tasks (liberating the user from the need to use his hands and/or eyes, informs about the identity of the speaker and allows disabled users to choose the modality that best fits them to interact with the computer. Also they have demonstrated to provide a more natural interaction than traditional GUI-based interfaces, and have a more affordable learning curve for people without enough technical knowledge [31].

Reeves and Nass [32] demonstrated that individuals' interactions with computers are fundamentally social in nature and correspond to the ways people naturally interact with other people, a principle which is commonly addressed as the "Media Equation". However, in human communication speech is not the only mode for conveying the desired content. Multimodal dialog systems cope with this limitation and Embodied Conversational Agents (ECAs) appeared. ECAs are virtual characters capable of producing and/or responding to verbal and nonverbal communication, usually with the appearance of a human [33, 34, 35]. Due to the great variety of application domains in which they might be employed, ECAs have allowed researchers to reveal a significant amount of behaviors that were taken for granted with traditional spoken dialog systems and which must be taken into account when dealing with a complete simulation of human communication [36].

One of the application domains of ECAs are virtual games and social virtual worlds, in which the Media Equation has a considerable impact [37]. However, some authors such as [38] argue that the debate on the existence of this social effect is suspect in the case of ECAs to a very complicated network of features. For example, bodies are salient indicators of social identity, but there are also many other factors which work together in predicting engagement, task performance and user satisfaction.

If an ECA is engaging, presumably it is more likely that it would be addressed as a person and that users will become more active and interact for longer with it. This situation has been corroborated in the case of pedagogic ECAs, in which engagement lead the students to interact more frequently and increase the time spent within the learning environment, with the result of better learning achievement [39]. Several recent studies have analyzed the myriad of factors that influence sustained interaction over time and user willingness to participate actively and collaborate with other users in the Second Life virtual world [40, 5, 41].

In addition, several authors have attributed emotional responses to events in virtual worlds as one of the most important aspects to increase the believability of these events, make users feel more real experiences, affect decision-making processes, and trigger a positive emotional response [29, 42, 43, 44]. Important factors described in the literature to elicit emotional responses include levels of autonomy (whether users are able to operate without assistance), presentation (whether the virtual environment resembles real-life,) immersion (presence level that users feels in the virtual world) and interactivity (realistic reactive behavior offered by the virtual world). These factors are in general consistent with the real world settings [45], and can even be amplified in the virtual world [46, 47]. Grinber et al. [48] also concluded that social engagement is more important that the realism of the virtual environment to increase the feeling of immersion in the virtual world.

Thus, engagement plays a fundamental role in order to obtain successful and frequent interactions of the users with the ECA. Engagement might be addressed form the perspective of visual realism maximization both of the ECA itself and the environment in which it is placed. This way, ECAs situated in meaningful virtual environments help to recreate situations in which specific conversational behaviors might arise. For example, Hubal et al. [49] studied neurocognitive and emotive predictors of behavioral problems among minority adolescents in high-risk urban settings by making them interact with an embodied conversational agent under controlled situations in predefined scenarios in which they had to show their emotional control and interpersonal communication skills.

Although it has been demonstrated that visual realism is not the only factor for user engagement with such characters [50], creating high resolution, vivid characters remains one of the highest priorities. However, a high realism might receive more negative evaluations than agents demonstrating only moderate realism [38]. This might be because high realism causes higher expectations in the users and thus provokes bad experiences when these expectations are not fulfilled.

Some studies like [38] have shown that these effects can be canceled by considering consistency as the most important indicator of realism. The authors showed that users prefer to interact with agents which show a consistent behavior rather than to highly realistic or human-like agents.

Thus, in this paper we propose a methodology to develop ECAs for social virtual worlds, which we will note as “conversational metabots”, so that they maximize the user engagement. This way, despite them being virtual, there will be social interactions between the users and the metabots in a human-like fashion. In order to do so, we endow the conversational metabots with a consistent conduct by building user models which reflect the users’ conversational behavior. To obtain meaningful results, we have primed the conversational behavior and not the multimodal rendering. Thus, we evaluate our conversational metabot from the conversational perspective although we make use of an optimized physical appearance and a set of basic gestures.

2.1. The Second Life and OpenSimulator virtual worlds

Second Life (SL) is a three dimensional virtual world developed by Linden Lab in 2003 and accessible via the Internet [26, 15, 51, 16, 52, 53]. A free client program called the Second Life Viewer enables its users, called “residents”, to interact with each other through avatars. Residents can explore, meet other residents, socialize, participate in individual and group activities, create and trade items (virtual property) and services from one another. The stated goal is to create a user-defined world of general use in which people can interact, play, do business, and otherwise communicate. SL is currently being used with success as a platform for education and research by many institutions, such as colleges, universities, libraries, health institutions, and government entities [54, 55, 29, 56, 57].

We decided to use Second Life as a testbed for our research for several reasons. Firstly, because it is one of the most widespread popular social virtual worlds available. Although its current popularity no longer reaches the levels it enjoyed in the early years of its existence in the 2000s, it still claims over 57 million accounts created from around the world, 68 million dollars paid to creators, and more than 41,000 residents connected at the same time at the time of writing[3]. Secondly, because it uses a sophisticated physics engine which generates very realistic simulations including collision detection, vehicle dynamics and animation look & feel, thus making the avatars and the environment more credible. Thirdly, because SL's capacity for customization is extensive and encourages user innovation and participation, which increases the naturalness of the interactions that take place in the virtual world. We own an island in Second Life called TESIS, in which different educational activities are performed. Figure 1 shows an image of the TESIS island.

Figure 1: Images of the TESIS island in Second Life

There are different ways in which the residents might communicate with each other. In [52, 26], these interactional affordances are classified into: language-based affordances (text-based chat, Instant Messaging (IM), voice over IP, notecards, action scripts, billboards, road signs, etc.), and avatar-based affordances (avatar appearance, avatar movements and avatar gestures).

[3] There were more than 41,000 residents online in Second Life on 2017-11-11 at 16:32:05 GMT+1 according to http://gridsurvey.com

Open chat, voice over IP and instant messaging are the main communication options [26]. Gestures are animations that can convey a mood or simulate an action. Second Life includes a tool for designing customized gestures, which can also be bought by buying them or trading with other residents [53].

Residents can also hear and view streaming audio and video inside Second. Residents can choose to display video on specific surfaces in the land they own. To do this, they designate the surface's texture as a media surface. If any other surface within that resident's land has the same texture, it will also display the streaming video. Since this can cause confusion, residents should make sure the surface they choose has a unique texture within their land.

Despite these interesting multimedia communication capabilities, speech communication is seldom employed in SL between avatars and metabots. Usually, metabots only provide information to the users, and thus the communication is unidirectional. In the cases in which a dialog takes place between human users and automatic metabots, it occurs through the chat box interface. Thus, although spoken communication is technically plausible in Second Life, it mainly takes place between human users and not between human users and metabots.

OpenSimulator (OpenSim) is an open-source alternative that can be used to simulate virtual environments similar to Second Life. It uses the same standard to communicate with their users and it is compliant with the Second Life viewer as well as a range of other viewers being developed by the open source community. The main features of OpenSimulator include the supporting of 3D virtual spaces of variable size within one single instance, realtime Physics Simulation, multiple clients and protocols, in world scripting using a number of different languages (including LSL/OSSL, C#, JScript and VB.NET), clients that create 3D content in real time, and the provision of unlimited ability to customize virtual world applications through the use of scene plug-in modules. Additionally, it allows to link completely free virtual worlds developed and hosted by different users using technologies such as OsGrid (http://www.osgrid.org/). OpenSim has been used to develop educational virtual worlds and discover usage behaviors, such as the USALSIM educational virtual world [58] developed by the University of Salamanca, and the virtual hospital ward for clinical pharmacy teaching developed by the Umea and Auckland Universities [59].

3. Our methodology for creating conversational metabots

Figure 2 shows the new architecture developed for the integration of conversational metabots both in the Second Life and OpenSim virtual worlds. The conversational agent that governs the metabot is outside the virtual world, using external servers that provide both data access and speech recognition and synthesis

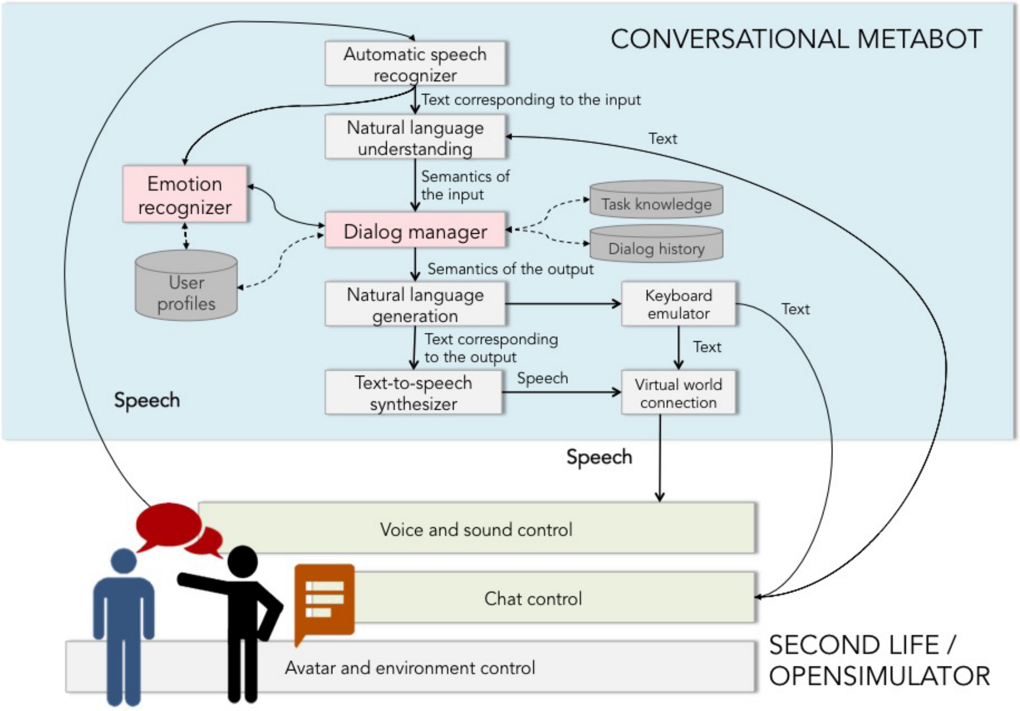


Figure 2: Architecture defined for the development of conversational metabots

functionalities. Using this architecture user's utterances can be easily recognized, the transcription of these utterances can be transcribed in the chat in Second Life, and the result of the user's query can be communicated using both text and speech modalities.

To successfully manage the interaction with the users, conversational agents usually carry out five main tasks: automatic speech recognition (ASR), natural language understanding (NLU), dialog management (DM), natural language generation (NLG) and text-to-speech synthesis (TTS). As Figure 2 shows, these tasks are usually implemented in different modules.

Speech recognition is the process of obtaining a sequence of words (sentence in text format) from a speech signal generated by a speaker [60, 61]. Usually virtual worlds do not have a native ASR system, so we propose to use the speech recognizer in the user's machine (client side). We consider realistic to assume that it is possible to use the client-side ASR, as all main operating system vendors have ASR services available.

Once the conversational agent has recognized what the user uttered, it is necessary to understand what he said. Natural language processing is the process of obtaining the semantic of a text string. It generally involves morphological, lexical, syntactical, semantic, discourse and pragmatical knowledge [62]. The dialog manager decides the next action of the system, for example, provide information to the user after a query to the databases [63]. In addition, it updates the dialog history, provides a context for interpreting the sentences, and coordinates the other modules of the conversational agent.

Following, the action selected by the dialog manager must be translated into a sentence in natural language. Natural language generation is the process of obtaining texts in natural language from a non-linguistic representation [64]. It is usually carried out in five steps: content organization, content distribution in sentences, lexicalization, generation of referential expressions and linguistic realization. Finally, a text-to-speech synthesizer transforms the sentences into synthesized speech [65].

In our architecture, the speech signal provided by the text to speech synthesizer is captured and transmitted to the voice server module in Second Life (SLVoice) using code developed in Visual C#. NET and the SpeechLib library. This module is external to the client program used to display the virtual world and is based on the Vivox technology, which uses the RTP, SIP, OpenAL, TinyXPath, OpenSSL and libcurl protocols to transmit voice data. We also use the lipsynch utility provided by Second Life to synchronize the voice signal with the lip movements of the avatar.

In addition, we have integrated a keyboard emulator that allows the transmission of the text transcription generated by the conversational avatar directly to the chat in Second Life. The system connection with the virtual world is carried out using the libOpenMetaverse library. This .Net library, based on the Client /Server paradigm, allows accessing and creating three-dimensional virtual worlds, and it is used to communicate with servers that control the virtual world of Second Life.

3.1. Adaptive dialog management

As previously described, the *Dialog Manager* selects the next system action according to a dialog strategy. The traditional approach to do this is to handcraft a series of rules which determine such behavior. However, this design method is very time consuming and has the ever-increasing problem of dialog complexity. As an alternative, statistical models can be trained from real dialogs, modeling the variability in user behaviors.

Our dialog manager follows this paradigm and is mainly based on the modelization of the sequences of the user and system dialog acts [66]. We represent dialogs as sequences of pairs (A_i, U_i), where A_i is the system response at time i, and U_i is the semantic representation provided by the natural language understanding module for the user input at time i). This way, a dialog can be represented by:

$$(U_1, A_1), \cdots, (U_i, A_i), \cdots, (U_n, A_n) \tag{1}$$

where A_1 is the greeting turn generated by the system, and U_n is the last user turn. We refer to a pair (A_i, U_i) as S_i, the state of the dialog sequence at time i.

At each time i, the objective of the dialog manager is to select the best system response A_i. This selection takes into account the previous history of the dialog (i.e., sequence of states of the dialog preceding time i):

$$\hat{A}_i = \operatorname*{argmax}_{A_i \in \mathcal{A}} P(A_i | S_1, \cdots, S_{i-1}) \tag{2}$$

where set $\mathcal{A}$ contains all the possible system responses.

As the number of all possible dialog histories is usually very large, we define a data structure, that we call *Dialog Register* (*DR*), to store the information provided by the user throughout the dialog. The possible values for each slot in the *DR* are $\{0, 1, 2\}$, according to the following criteria: (0) The user has not provided a value for the corresponding slot; (1) the user has provided a value for the slot and its confidence score provided by the ASR and NLU modules is higher than a given threshold; (2) the value for the slot has a confidence score that is lower than the given threshold.

During a dialog, the user provides values for the slots defined in this data structure (task-dependent information) and the dialog manager considers these information pieces to select the next system response. In addition, users can provide task-independent information (for instance, *Not-Understood*, *Affirmation* and *Negation* dialog acts). This information sources imply system decisions that are different from simply updating the DR_{i-1}. Hence, for the selection of the best system response A_i, we take into account the *DR* that results from turn 1 to turn $i-1$, and we explicitly consider the last state S_{i-1}:

$$\hat{A}_i = \operatorname*{argmax}_{A_i \in \mathcal{A}} P(A_i | DR_{i-1}, S_{i-1}) \tag{3}$$

There are also more complex application tasks in which the dialog manager must consider not only the information by the user during the dialog, but also the results generated after the queries to the data repositories of the application or the validation of restrictions and privacy policies. Thus, the information generated by the module that controls the application (which we denote as the *Application Manager*, *AM*) must also be taken into account in these tasks for the selection of the best system action.

For this reason, we have decided that for this kind of tasks, two phases are required for the selection of the next system response. In the first phase, the information stored in the DR and the last state S_{i-1} are considered to select the best request to be made to the AM ($\hat{A}_{1_i}$):

$$\hat{A}_{1_i} = \operatorname*{argmax}_{A_{1_i} \in \mathcal{A}_1} P(A_i | DR_{i-1}, S_{i-1}) \tag{4}$$

where $\mathcal{A}_1$ is the set of possible requests to the AM.

In the second phase, the system response ($\hat{A}_{2_i}$) is selected considering $\hat{A}_{1_i}$ and the information provided by the AM (AM_i):

$$\hat{A}_{2_i} = \operatorname*{argmax}_{A_{2_i} \in \mathcal{A}_2} P(A_i | AM_i, A_{1_i}) \tag{5}$$

where $\mathcal{A}_2$ is the set of possible system responses.

We propose to solve the equations 4 and 5 by means of a classification process. The classification function can be defined in several ways. In our previous work on statistical dialog management, we evaluated several definitions of the classification function [66, 23]. The best results were obtained using a multilayer perceptron (MLP) [67].

To deal with context information and personalize the selection of the next system response, we have incorporated user profiles that are also taken into account in the classification process in the previously described phases. This profile consists of user's:

- Id, bot identifier used to log in to the system. The Id is used to personalize the system prompts including the identifier of the user (e.g., the welcome prompt: Good morning José, how can I help you today?);

- Experience, which can be either 0 for novel users (first time the user employs the system) or the number of times the user has interacted with the system. The experience is used to personalize the systems prompts adding more detailed information and incorporating more complete help prompts for novel users;

- Skill level, estimated taking into account the level of expertise, the duration of their previous dialogs and the time that was necessary to access a specific content and the date of the last interaction with the system. A low, medium, high or expert level is assigned using these measures. This information is complementary to the experience, as an experienced user may have difficulties accessing certain contents, and so they can be treated as novel for certain aspects (with more detailed prompts) and experienced for others (with shorter system interventions);

- Most frequent objective of the user and preferred output modality. The most frequent objective is used by the system to suggest the user to consult information related to this objective if the user does not provide a query or an error is detected during the dialog. The output modality can be either the bot's voice or chat transcription or both.

In addition to the information stored in the user's profile, the dialog manager also considers the emotions detected in the user's utterances, according to the results of the proposal for emotion recognition described in Section 3.2.

3.2. Emotion recognition

Emotion recognition is a research topic at the intersection of different areas, including Computational Linguistics, Natural Language Processing, Data Mining, and Information Retrieval [68, 69, 70]. Usually these areas are based on concepts such subjectivity, opinion, or emotion. Emotion plays a key role in human interaction and emotion recognition is currently at the core of the most advanced conversational interfaces [19, 18, 23] to operate in scenarios that are colored with affect and provide personalized services fostering acceptance and trust, such as social virtual worlds.

Recently, we have presented a specific methodology for emotion recognition in conversational interfaces [71]. Our proposal is focused on the recognition of different negative emotions. These bad experiences may have a detrimental effect on the system's usability and acceptance (i.e., discourage users from finishing the interaction with the conversational interface or even employing the system again). Concretely, we center on three negative emotions: doubtfulness, anger and boredom, as well as neutral. To obtain better emotion recognition results for user spoken utterances we use a supervised machine learning approaches and a detailed set of paralinguistic features.

Once these emotional states are detected, the dialog manager tailors the next system answer to the user state by changing the help providing mechanisms, the confirmation strategy and the interaction flexibility. The conciliation strategies adopted are, following the constraints defined in [72], straightforward and well delimited in order not to make the user loose the focus on the task. They are as follows:

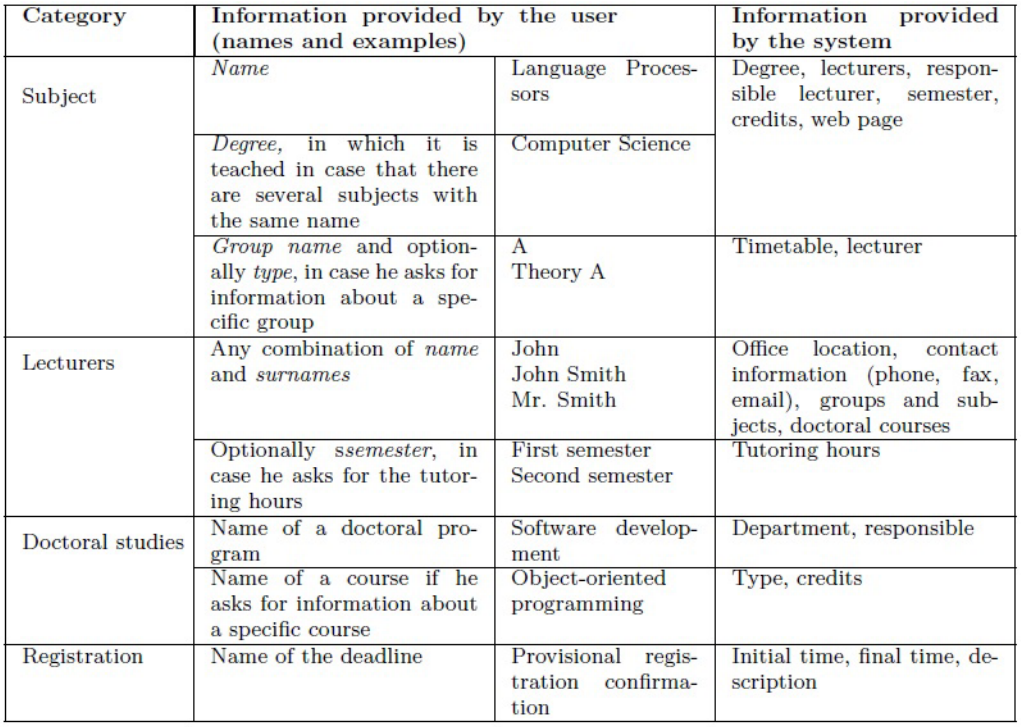
- If the system recognizes the doubtful emotion and the emotions detected in the previous turns of the current dialog are different, the dialog manager decides to restrict the dialog initiative to a system-directed initiative and provides a help message at the end of each system response. The main objective is to describe the possible options for each one of the requirements of the conversational system. The same process is selected if no profile is available for the current user, the profile shows that he/she is a non-expert user, or the first utterances have been also classified as doubtful.

- If anger has been detected, the system apologizes where it has automatically detected that there have been recognition errors in the previous dialog turns. If no communication errors have been detected, the dialog manager informs the user that he/she can require additional help at any time during the dialog, and selects predefined templates to reformulate the messages to the user in a more agreeable way.
- If it is detected that the user could be bored, the strategy selected by the dialog manager is to verify if the user has previously interacted with the system. In this case, the system infer from the user profile the query more times required in the previous dialogs. If this query matches the one detected for the current dialog, the dialogue manager selects templates with more direct messages for the user, uses implicit confirmations, and takes information for granted instead of requiring it to the user (e.g., the dialog manager tries to automatically disambiguate among several subjects if it has been detected that the student has always selected the same academic degree in the previous dialogs).
- In the rest of the cases, the neutral emotion is assumed and the dialog manager selects the next system response taking into account only the slots completed in the dialog register up to the current moment of the dialog and the user profile (previous interactions, preferences, and expertise level).

4. Creation of a conversational metabot for a specific domain

Following the proposal described in the previous section, we have developed a conversational metabot called Demic (see Figure 4) that facilitates two main purposes: provide academic information and carry out test and questionnaires. These functionalities are based on a previously developed dialog system that worked over the telephone [73, 74]. The information that the metabot is able to provide can be classified in four main groups: subjects, professors, doctoral studies and registration, as shown in Table 1. As can be observed, the system must ask the user for different pieces of information before producing a response.

We defined a semantic representation in which one or more concepts represent the intention of the utterance, and a sequence of attribute-value pairs contains the information about the values given by the user. In the case of user turns, we defined four concepts related to the different queries that the user can perform (*Subjects*, *Lecturers*, *Doctoral studies*, *Registration*), three task-independent concepts



Category	Information provided by the user (names and examples)		Information provided by the system
Subject	*Name*	Language Processors	Degree, lecturers, responsible lecturer, semester, credits, web page
	Degree, in which it is teached in case that there are several subjects with the same name	Computer Science	
	Group name and optionally *type*, in case he asks for information about a specific group	A Theory A	Timetable, lecturer
Lecturers	Any combination of *name* and *surnames*	John John Smith Mr. Smith	Office location, contact information (phone, fax, email), groups and subjects, doctoral courses
	Optionally *ssemester*, in case he asks for the tutoring hours	First semester Second semester	Tutoring hours
Doctoral studies	Name of a doctoral program	Software development	Department, responsible
	Name of a course if he asks for information about a specific course	Object-oriented programming	Type, credits
Registration	Name of the deadline	Provisional registration confirmation	Initial time, final time, description

Table 1: Information provided by the conversational metabot

(*Affirmation*, *Negation*, and *Not-Understood*), and eight attributes (*Subject-Name*, *Degree*, *Group-Name*, *Subject-Type*, *Lecturer-Name*, *Program-Name*, *Semester*, and *Deadline*). The *DR* defined for the task consists of 12 slots defined by experts who have identified the four possible user queries and the eight attributes described.

An example of the semantic interpretation of an input sentence is shown below:

User Turn: I would like to know information for the group 88 of the subject Formal Grammars and Automata Theory of the Computer Science Degree.
Semantic Representation:
(*Subjects*)
Subject-Name: Formal Grammars and Automata Theory
Group-Name: 88
Degree: Computer Science

The labeling of the system turns is similar to the labeling defined for the user turns. A total of 30 task-dependent concepts were defined:

- Task-independent concepts (*Affirmation*, *Negation*, *Not-Understood*, *New-Query*, *Opening*, and *Closing*).

- Concepts used to inform the user about the result of a specific query (*Subjects*, *Lecturers*, *Doctoral-Studies*, and *Registration*).

- Concepts defined to require the user the attributes that are necessary for a specific query (*Subject-Name*, *Degree*, *Group-Name*, *Subject-Type*, *Lecturer-Name*, *Program-Name*, *Semester*, and *Deadline*).

- Concepts used for the confirmation of concepts (*Confirmation-Subject*, *Confirmation-Lecturers*, *Confirmation-DoctoralStudies*, *Confirmation-Registration*) and attributes (*Confirmation-SubjectName*, *Confirmation-Degree*, *Confirmation-GroupName*, *Confirmation-SubjectType*, *Confirmation-LecturerName*, *Confirmation-ProgramName*, *Confirmation-Semester*, and *Confirmation-Deadline*).

An initial corpus of 150 dialogs was acquired with the conversational system by means of its interaction with students and professors [73, 74]. Figure 3 shows an example of a dialog extracted from this corpus and translated from Spanish to English. Turns tagged with S are system turns, and with U user turns.

Using the previously described codification for the DR, when a dialog starts (in the opening turn) all the values in the dialog register are initialized to "0". The information provided by the users in each dialog turn is employed to update the previous DR and obtain the current one, as Figure 3 shows.

This figure shows the semantic interpretation and confidence scores (in brackets) for the user's utterances provided by the SLU module. In the first utterance, the user's query about subjects and the name of the degree are recognized with a high confidence score. Thus, a "1" value is added in the corresponding positions of the DR_1. There is not task-independent information (Affirmation, Negation and Not-Understood dialog acts).

In the second utterance, the user provides the name of the subject and the group. In this case, the confidence score assigned to the attribute *Subject-name* is very low. Thus, a "2" value is added in the corresponding position of the DR_2. As the input to the MLP is generated using DR_2, the codification of the labeling of the last system turn (A_1), and the task-independent information provided in the last user turn (none in this case), the dialog manager selects to confirm the name of the subject. This process is repeated to select the next system response after each user turn.

As described in the previous section, the dialog manager requires the second phase to take into account the response generated by the Application Manager

S_1: Hi! I'm Demic. I can provide information about subjects, lecturers, doctoral studies and registrations.
Opening
DR_0: 0000-00000000
U_1: I want to know information about subjects in the Computer Science Degree.
TDI: (*Subjects*) [0.8] *Degree*:Computer Science [0.9]
TII: 000
DR_1: 1000-01000000
S_2: Any subject in particular?
Subject-Name
U_2: I want information for the Group 88 of Formal Languages and Automata Theory.
TDI: *Subject-Name*: Formal Grammars and Automata Theory [0.2], *Group-Name*: 88 [0.8]
TII: 000
DR_2: 1000-21100000
Subject-Name: Formal Grammars and Automata Theory
S_3: Do you want information about Formal Languages and Automata Theory?
Confirmation-SubjectName
U_3: Yes
TDI: ()
TII: 100
DR_3: 1000-11100000
S_4: The schedule of Group 88 is: Wednesday 15:00-17:00, Friday 17:00-19:00. Anything else?
AM: *Schedule-information available*
U_5: No, thanks!
TDI: ()
TII: 010
DR_4: 1000-11100000
S_6: Bye!
Closing

Figure 3: Example of dialog for the academic task

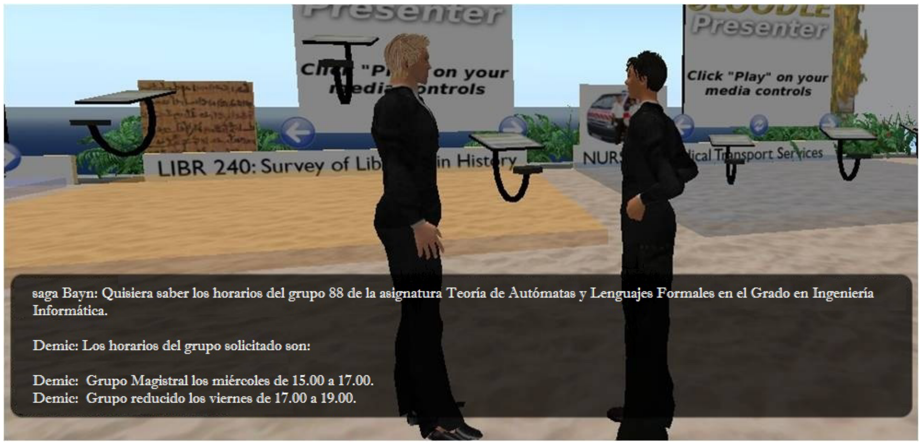
module. The dialog manager would either: 1) provide the response selected by the *AM* in the first phase (e.g., when the dialog manager selects the confirmation of an attribute, to ask the user for additional information, or to finish the dialog) or 2) produce a different answer when the *AM* informs that the information is not available or there is any error.

Demic uses the official Second Life client to carry out its connection to the virtual world. The Java MetaBotLib is used to encapsulate the messages generated by the dialog system and the inputs provided to the system. Figure 4 shows Demic interacting with the avatar of a user.

Speech recognition and synthesis are performed using the Microsoft Speech Application Programming Interface (SAPI) 5.1, integrated into the Microsoft Windows 10 operating system.

4.1. Spoken Language Understanding

As previously described, the Spoken Language Understanding (SLU) component converts natural language sentences (i.e. human language) into a set of data



Presenter
Click "Play" on your media controls
LIBR 240: Survey of Lib in History
NURS
ical Transport Services

saga Bayn: Quisiera saber los horarios del grupo 88 de la asignatura Teoría de Autómatas y Lenguajes Formales en el Grado en Ingeniería Informática.

Demic: Los horarios del grupo solicitado son:

Demic: Grupo Magistral los miércoles de 15.00 a 17.00.
Demic: Grupo reducido los viernes de 17.00 a 19.00.

Figure 4: Image of Demic (on the right) in Second Life

that can be understood by the system. Two main types of statistical approaches have been proposed in recent years to address the SLU task: generative and discriminative models [18, 75, 76]. Generative models calculate the joint probability of concepts and semantic constituents. They are robust to over-fitting and they are less affected by errors and noise. However, they cannot easily integrate complex structures. Discriminative models learn a classification function based on conditional probabilities of concepts given words. These models can easily integrate very complex features that can capture arbitrarily long-distance dependencies. On the other hand, they usually over-fit training data.

The most representative generative models for language understanding are based on the Hidden Vector State model (HVS), Stochastic Finite State Transducers (SFST), and Dynamic Bayesian Networks (DBN).

The HVS model extends the discrete Markov model encoding the context of each state as a vector. As detailed in [77], all the parameters of the model are denoted by λ and each state at time t is denoted by a vector of semantic concept labels.

$$c_t = c_{1t}, c_{2t}, \cdots, c_{D_t} \tag{6}$$

where c_{1t} is the preterminal concept and c_{D_t} is the root concept.

The joint likelihood function is defined as:

$$L(\lambda) = logP(W, C, N|\lambda) \tag{7}$$

where W is the word sequence, C is the concept vector sequence, and N is the sequence of stack pop operations.

The auxiliary function Q is defined to apply the Expectation-Maximization (EM) technique to maximize the expectation of $L(\lambda)$ given the observed data and current estimates:

$$Q(\lambda|\lambda^k) = E \cdot log(P(W, C, N|\lambda^k)) \cdot \sum_{C,N} P(C, N|W, \lambda) log(P(W, C, N|\lambda^k)) \qquad (8)$$

The term $P(W, C, N)$ can be decomposed as follows:

$$P(W, C, N) = \prod_{t=1}^{T} P(n_t|W_1^{t-1}, C_1^{t-1}) \cdot P(c_t[1]|W_1^{t-1}, C_1^{t-1}, n_t) \cdot P(w_t|W_1^{t-1}, C_1^{t-1}) \quad (9)$$

SFSTs model the SLU task as a translation process from words to concepts, using Finite State Machines (FSM) to implement the stochastic language models [78]. An FSM is defined for each elementary concept. Each transducer takes words as input and outputs the concept tag conveyed by the accepted sentence. All these transducers are grouped together into a single transducer, called λ_{W2C}, which is the union of all of them. A stochastic conceptual language model is computed as the joint probability $P(W, C)$:

$$P(W, C) = \prod_{i=1}^{k} P(w_i c_i | h_i) \qquad (10)$$

where $h_i = w_{i-1}c_{i-1} \cdots w_1 c_1$ is usually approximated by $h_i = w_{i-1}c_{i-1}, w_{i-2}c_{i-2}$ as a 3-gram model; $C = c_1, c_2, \cdots, c_k$ is the sequence of concepts; and $W = w_1, w_2, \cdots, w_k$ is the sequence of words.

The concept of decoding is reformulated DBNs for SLU to combine the concept sequence with the value sequence as follows:

$$\hat{c}_1^N, \hat{v}\,_1^N = argmax_{{c1}^N, {v1}^N} p(c_1^N, v_1^N | w_1^T) \cdot argmax_{{c1}^N, {v1}^N} p(w_1^T, c_1^N | v_1^N) p(v_1^N | c_1^N) P({c1}^N) \qquad (11)$$

where we have hypothesized the terms ${c1}^N$ by means of:

$$\hat{c}_1^N = argmax_{{c1}^N} \cdot \sum_{{v1}^N} argmax_{{c1}^N, {v1}^N} p(w_1^T, c_1^N | v_1^N) p(v_1^N | c_1^N) P({c1}^N) \qquad (12)$$

The most representative discriminative models for SLU are based on support vector machines (SVMs) [78] and conditional random fields (CRFs) [79].

SVMs are machine-learning algorithms included into the class of linear classifiers. These models learn a hyperplane

$$H(\overrightarrow{x}) = \overrightarrow{w}\overrightarrow{x} + b = 0 \tag{13}$$

that divides training examples with maximum margin, where the learned parameters are given as follows:

- $\overrightarrow{x}$ is the feature vector representation of a classifying object o;
- $\overrightarrow{w} \in R$
- $b \in R$ (Vapnik 1998).

The hyperplane can be represented in the following dual form applying the Lagrangian optimization theory:

$$\sum_{i=1\cdots l} y_i \alpha_i \overrightarrow{x}_l \overrightarrow{x} + b = 0 \tag{14}$$

where $\overrightarrow{x}_l$ are the training examples, y_i is the label associated with $\overrightarrow{x}_l$ (+1 or -1), and α_i are the Lagrange multipliers.

CRFs are log-linear models that train conditional probabilities considering features of the input sequence. Conditional dependence is captured using feature functions and a factor for probability normalization. The conditional probabilities of the concept sequences $c_1^N = c_1, \cdots, c_N$ given the word sequences $w_1^N = w_1, \cdots, w_N$ are calculated by means of:

$$p(c_1^N | w_1^N) = \frac{1}{Z} \prod_{n=1}^{N} exp(\sum_{m=1}^{M} \lambda_m \cdot h_m(c_{n-1}, c_n, w_{n-2}^{n+2})) \tag{15}$$

where λ_m is the vector of parameters to be trained, $h_m(c_{n-1}, c_n, w_{n-2}^{n+2})$ are the feature functions used to capture dependencies between input features (words and other features that can be associated with words in a window around the current word to be labeled) and the output concept [80].

4.2. Neural networks for dialog management

To apply a MLP to select the next system response in the dialog management process, the input layer holds a codification of the input pairs specified in equations 4 and 5. The representation defined for the different terms is as follows:

- Dialog Register (DR_{i-1}): As previously stated, the DR includes N task-dependent user dialog acts, which can take the values $\{0, 1, 2\}$ and then be modeled using a variable with three bits.

$$\vec{x}_i = (x_{j_1}, x_{j_2}, x_{j_3}) \in \{0,1\}^3 \; j = 2, \cdots, N+1 \tag{16}$$

$$\vec{x}_{DR} = (x_{1_1}, x_{1_2}, x_{1_3}, \cdots, x_{1_N}) \in \{0,1\}^N \tag{17}$$

- Last system response (A_{i-1}): This information is modeled using a variable, which has as many bits as possible system responses (C).

$$\vec{x}_A = (x_{1_1}, x_{1_2}, x_{1_3}, \cdots, x_{1_C}) \in \{0,1\}^C \tag{18}$$

  where C is the number of possible system responses (i.e., system actions) as it has been previously described.

- Task-independent information (*Affirmation*, *Negation*, and *Not-Understood* dialog acts): These three dialog acts have been coded with the same codification used for the task-dependent information in the DR; that is, each one of these three dialog acts can take the values $\{0, 1, 2\}$. This information is modeled using three variables with three bits.

$$\vec{x}_{TII} = (x_{j_1}, x_{j_2}, x_{j_3}) \in \{0,1\}^3 \tag{19}$$

- Output of the Application Manager (AM_i): This output is modeled using a variable, which has as many bits as possible responses defined for the AM (M).

$$\vec{x}_{AM} = (x_{1_1}, x_{1_2}, x_{1_3}, \cdots, x_{1_M}) \in \{0,1\}^M \tag{20}$$

To train and evaluate the neural networks, we used the *April* toolkit [81]. We firstly tested the influence of the topology of the MLP, by training different MLPs of increasing number of weights using the standard backpropagation algorithm (with a sigmoid activation function and a learning rate equal to 0.2), and selecting the best topology according to the mean square error (MSE) of the validation data. The minimum MSE value was achieved using an MLP of one hidden layer of 32 units. We followed our experimentation with MLPs of this topology, training MLPs with several algorithms. The best result on the validation data was obtained using the MLP trained with the standard backpropagation algorithm and a value of LR equal to 0.3.

5. Evaluation of our proposal

In this section we describe the process and measures defined for the evaluation of the conversational metabot described in the previous section. The evaluation of our proposal has been completed by means of two main processes. Firstly, several experts developed an initial, handcrafted dialog strategy for the conversational system, resembling the behavior of experienced human operators who provide the same information. After implementing this strategy, the initial corpus of 150 dialogs described in the previous section was collected from spontaneous telephone calls to the initial dialog system.

Secondly, we employed a statistical dialog simulation technique previously developed [82] to generate 50,000 successful dialogs for the task. The goal was to explore the space of possible dialog states to learn a new dialog model for the Demic metabot. To do so, a set of different scenarios was created for the user simulator, with the same goals as those of the real users in the initial corpus for each task. Figure 5 shows an example of a simulated dialog corresponding to a scenario with two objectives (to know timetables of a given lecturer and subject, and information about the registration in its specific degree). The confidence scores generated by an error simulator are shown between brackets. Sentences in natural language, that are equivalent to the labeled system and user turns, have been incorporated to clarify the explanation. The results of the comparative assessment of the initial dialog manager and the one trained using the proposed statistical dialog management technique with the successful simulated dialogs and the dialog corpus collected from the real users will be discussed in Section 5.2.

Finally, we evaluated the behavior of the conversational metabot with recruited users using the same set of scenarios designed for the user simulation. A total of 250 dialogs were recorded from the interactions of 55 students with the Demic metabot. Additionally, these users were asked to fill in a questionnaire with their opinion about several aspects of the interaction. The results of the objective evaluation using the corpus and the subjective evaluation using the information provided in the questionnaires will be discussed in Section 5.3.

5.1. Evaluation of the spoken language understanding approaches

The accuracy of the SLU module is evaluated measuring the percentage of sentences that are 'correctly understood'. That is, the percentage of sentences whose semantic representation is equal to the reference annotated in the initial corpus (f_c):

$$f_c = 100 * \frac{number\_of\_sentences\_correctly\_annotated}{total\_number\_of\_sentences} \qquad (21)$$

S1: Opening
Hi! I'm Demic. I can provide information about subjects, lecturers, doctoral studies and registrations.
U1: Lecturers [0.1]
I want to know information about lecturers.
S2: (Confirmation:Lecturers)
Do you want to know information about lecturers?
U2: Affirmation [0.9]
Yes
S3: (Question:Lecturer_Name)
Tell me the name of the lecturer.
U3: Lecturer_Name: John Smith [0.1]
John Smith
S4: (Confirmation:Lecturer_Name)
Do you want to know the information about John Smith?
U4: Affirmation [0.9]
Yes
S5: (Answer:Lecturers) (New-Query)
{*Lecturer information*} *Anything else?*
U5: (Question:Registration) [0.9]
Degree: Computer Science[0.9]
The registration information in Computer Science
S6: (Answer:Registration) (New-Query)
{*Registration information*} *Anything else?*
U6: Negation
S7: (Closing:Nil:Nil)
Thank you!

Figure 5: An example of a dialog acquired by means of the simulation technique

We have evaluated the approaches described in Section 4.1 using the initial 150 dialogs. Table 2 shows the results of the evaluation. As it can be observed, the best results were obtained using Support Vector Machines. Thus, this methodology was used for the practical implementation of the SLU module for the Demic metabot.

Methodology	f_c
Hidden Vector State	78.33%
Stochastic Finite State Transducers	85.41%
Dynamic Bayesian Networks	88.11%
Support Vector Machines	94.13%
Conditional Random Fields	92.64%

Table 2: Results of the evaluation of the different statistical approaches for SLU

5.2. Evaluation of the user's adapted dialog management methodology

A 5-fold cross-validation process has been used to evaluate the initial dialog manager developed for the task (initial dialog model) and the one trained with the simulated and real dialogs (enhanced dialog model) incorporating the user-adaptive strategy. The corpus for each dialog manager was randomly split into five folds, each containing 20% of the corpus. The experiments were carried out in five trials, each using as a test set a different fold whereas the remaining folds were used as the training set. A validation subset (20%) was extracted from each training set. We carried out a detailed study of the dialogs obtained with both dialog managers using the set of quantitative evaluation measures proposed in [83, 84]. We then used two-tailed t tests to compare the means across the different types of scenarios and users as described in [83]. The significance of the results was computed using the SPSS software with a significance level of 95%.

We propose three measures to evaluate the adaptive dialog manager compared to the initial dialog manager. The first measure, which we call *%unseen*, makes reference to the percentage of unseen situations, i.e., the dialog situations that are present in the test partition but are not present in the corpus used for learning the dialog model. The following measures are calculated by comparing the response automatically generated by the dialog manager for each input in the test partition with regard to the reference response annotated in the evaluation corpus. This way, the evaluation is carried out turn by turn. These three measures are:

- *%real*: the percentage of responses provided by the dialog manager that are contained in the set of possible responses annotated in the training corpus for the same dialog situation;
- *%coherent*: the percentage of responses provided by the dialog manager that are coherent with the current state of the dialog although they are not contained in the set described for the previous measure.
- *%error*: the percentage of responses provided by the dialog manager that would cause the failure of the dialog;

The measure *%real* is automatically calculated. On the other hand, the measures *%coherent* and *%error* are manually evaluated by an expert in the task. The expert evaluates whether the response provided by the dialog manager allows the correct continuation of the dialog for the current situation or whether the answer causes the failure of the dialog (e.g., the conversational metabot suddenly ends the interaction with the user, the user would not be able to answer to the requirement of the metabot, etc.).

Finally, the number of responses generated by the MLP that can cause the failure of the system is only a 4.67% percentage. A response that is coherent with the current state of the dialog is generated in 95.33% of cases. These last two results also demonstrate the correct operation of the classification methodology.

A new dialog model was learned each time a new set of simulated dialogs was generated. Table 3 shows the results of the evaluation of the initial dialog model and the dialog model obtained after the successful simulated dialogs were incorporated to the training corpus.

	%unseen	%real	%coherent	%error
Initial dialog model	11.18%	93.41%	95.33%	4.67%
Enhanced dialog model	6.25%	81.13%	98.65%	2.21%

Table 3: Results of the evaluation of the initial dialog model and the one obtained after the dialog simulation

The results of the *%real* and *%coherent* measures for the initial dialog model show the satisfactory operation of the developed dialog model. The codification developed to represent the state of the dialog and the good operation of the MLP classifiers make it possible for the response selected by the dialog manager to agree with one of the reference responses for the same dialog situations (*%real*) by a percentage of 93.41%.

It can be observed that, once the successful simulated dialogs were incorporated, the number of unseen situations was reduced by 4.93%, as expected with the addition of the simulated dialogs. Figure 6 and Figure 7 respectively show how the number of unseen situations (%unseen) and erroneous system responses (%error) decreased when the training corpus was enriched by adding the simulated dialogs, which is the expected behavior. These measures continued to decrease until 20,000 dialogs were simulated.

Figure 8 shows the evolution of %real and %coherent measures. The evolution of the %real and %coherent measures shows how the dialog manager can move away from an initial strategy by increasing the number of system responses that are coherent with the current situation in the dialog. Thus, the variability of the dialog model is increased by detecting new dialog situations that are not present in the initial dialog model and new valid responses for the situations that were already contained in the initial corpus. The results also show the reduction in the %error measure (from 4.67% to 2.21%).

Regarding the quality of the dialogs obtained using the initial and the enhanced dialog models, we computed the following high-level groups of dialog features: the average number of turns per dialog, the percentage of different dialogs without considering the attribute values, and the distribution of user and system dialog acts. On the system side, we have measured the confirmation of concepts and

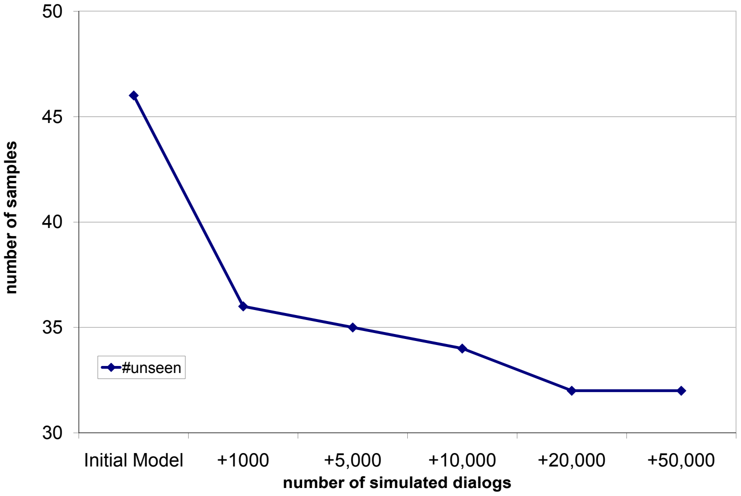


Figure 6: Evolution of the number of unseen situations (#unseen) with regard to the incorporation of new simulated dialogs in the dialog model

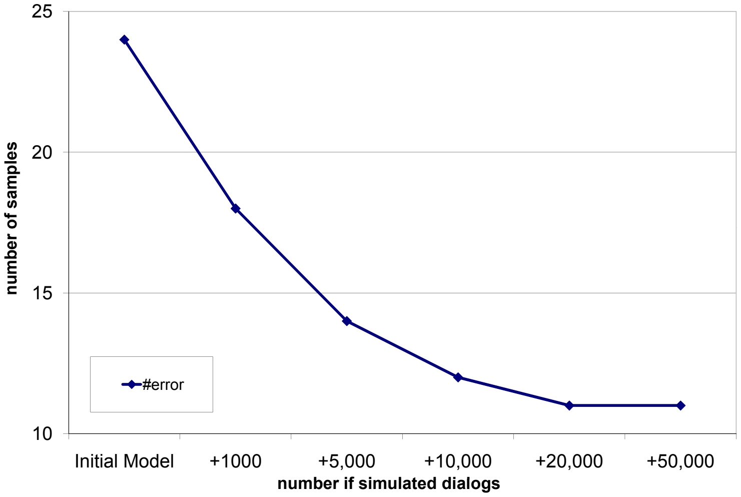


Figure 7: Evolution of the number of erroneous system answers (#error) with regard to the incorporation of new simulated dialogs in the dialog model

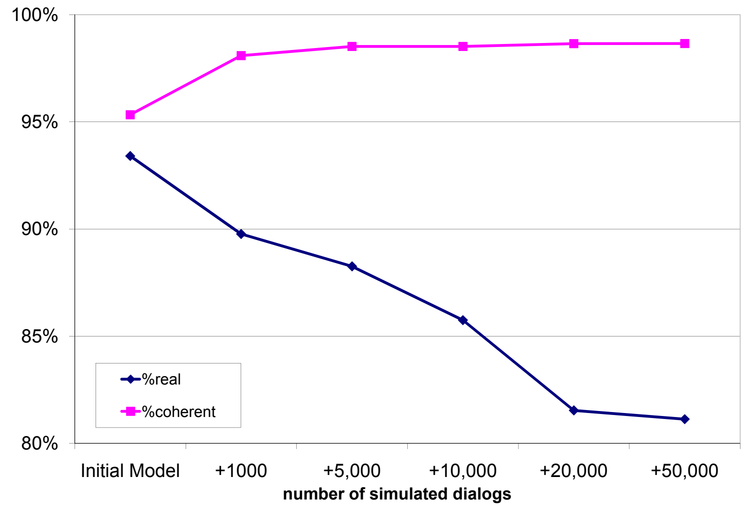


Figure 8: Evolution of the %real and %coherent measures with regard to the incorporation of new simulated dialogs in the dialog model

attributes, questions to require information, and system answers generated after a database query. We have not take into account the opening and closing system turns. On the user side, we have measured the percentage of turns in which the user carries out a request to the system, provide information, confirms a concept or attribute, Yes/No answers, and other answers not included in the previous categories. Table 4 shows the results of the comparison of the high-level dialog features.

It can be seen that there are significant differences between the dialogs acquired with both dialog managers. It can be observed that there is a reduction in the average number of turns when the enhanced model is used. The results also show a higher variability in the dialogs generated with this dialog manager as there is a higher percentage of different dialogs. These results show that improving the dialog strategy made it possible to reduce the number of necessary system actions to attain the dialog goals for the different tasks. There was an increment in the number of system turns providing information to the user. The number of confirmation turns is reduced with the enhanced model, which explains the higher proportion of user responses to request and provide information.

	Initial system	**Enhanced system**
Average number of user turns per dialog	12.9±2.3	9.8±1.6
Percentage of different dialogs	62.90%	77.42%
System responses		
Confirmations	17.41%	12.23%
Questions to require information	18.79%	16.57%
Responses generated after a database query	63.80%	71.20%
User responses		
Request to the system	32.74%	35.43%
Provide information	21.72%	25.98%
Confirmations	11.81%	7.34%
Yes/No answers	33.73%	31.25%

Table 4: Results of the high-level dialog features defined for the comparison of the two corpora acquired

5.3. Evaluation with recruited users

A set of 250 dialogs has been also acquired with the metabot by means of its interaction in the virtual world with 55 recruited students of the Computer Science Degree at the Carlos III University of Madrid. The acquisition process resulted in a spontaneous Spanish speech dialog corpus with a duration of 350 minutes. We have completed an objective and subjective assessments of the conversational metabot. For the objective evaluation, we considered the following statistical measures:

1. Dialog success rate (Success Rate). This is the percentage of successfully completed dialogs in which the metabot provides the correct information to each one of the required questions.
2. Average number of turns per dialog (nT).
3. Confirmation rate (Confirmation Rate). It was computed as the ratio between the number of explicit confirmations turns (nCT) and the number of turns in the dialog (nCT/nT).
4. Average number of corrected errors per dialog (nCE). This is the average of errors detected and corrected by the dialog manager of the conversational metabot. We have considered only those errors that modify the values of the attributes and that could cause the failure of the dialog.
5. Average number of uncorrected errors per dialog (nNCE). This is the average of errors not corrected by the dialog manager. Again, only errors that modify the values of the attributes are considered.

6. Error correction rate (ECR). The percentage of corrected errors, computed as nCE/ (nCE + nNCE).

The results presented in Table 5 for the described 200 dialogs show that the developed conversational metabot could interact correctly with the users in most cases, achieving a success rate of 94%. The dialog success depends on whether the Demic metabot provides the correct data for every query required by the user. The analysis of the main problems detected in the acquired dialogs shows that, in some cases, the conversational metabot did not detect that the user wanted to finish the dialog. A second problem was related to the introduction of data with a high confidence value due to errors generated by the automatic speech recognizer that were not detected. However, the evaluation confirms a good operation of the approach since the information is correctly provided by the metabot in the majority of cases, as it is also shown in the value of the error correction rate.

Success Rate	94%
nT	11.6
Confirmation Rate	28%
ECR	93%
nCE	0.89
nNCE	0.06

Table 5: Results of the objective evaluation of the conversational metabot

In addition, we have completed an evaluation of the conversational metabot based on questionnaire to assess the students' subjective opinion about the metabot performance. The questionnaire had 10 questions and the answers were placed in the 5-points Likert scale: i) Q1: State on a scale from 1 to 5 your previous knowledge about new technologies for information access; ii) Q2: State on a scale from 1 to 5 your previous experience with virtual worlds like Second Life; iii) Q3: How well did the metabot understand you?; iv) Q4: How well did you understand the messages generated by the metabot?; v) Q5: Was it easy for you to get the requested information?; vi) Q6: Was the interaction rate adequate?; vii) Q7: Was it easy for you to correct the metabot errors?; viii) Q8: Were you sure about what to say to the system at every moment?; ix) Q9: Do you believe the system behaved similarly as a human would do?; x) Q10: In general terms, are you satisfied with the metabot performance?. Table 6 shows the average, minimal and maximum values for the subjective evaluation.

From the results of the evaluation, it can be observed that students positively evaluated the facility of obtaining the data necessary to complete the exercises

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Mean Score	4.6	3.1	4.0	4.5	4.2	3.8	3.3	3.2	3.8	4.4
Maximum Value	5	4	5	5	5	4	4	4	4	5
Minimal Value	4	1	3	3	3	2	2	2	3	3
Std. Deviation	0.3	1.7	0.6	0.5	0.3	0.5	1.1	1.0	1.3	0.6

Table 6: Results of the subjective evaluation of the conversational metabot

and found the interaction rate suitable. The suggestions that they mentioned for the improvement of the system include the correction of system errors and a better clarification of the set of actions expected by the metabot at each time. Another interesting consideration concerns the correlation between the student background and the rest of scores. We verified that the questionnaire results are not influenced by the sample characteristics: user impressions are positive also when students did not have a previous experience with virtual worlds. The students were very satisfied with the experience, not only because it facilitated learning but also because it was amusing for them.

6. Conclusions

The development of social networks and virtual worlds brings a wide set of opportunities and new communication channels that cannot be fully unveiled with traditional interfaces. In this paper, we have proposed a methodology to develop embodied conversational agents that are able to interact as conversational metabots in virtual worlds. A practical implementation of an automatic avatar that provides academic information has been integrated in Second Life to evaluate our proposals.

Social virtual worlds, such as Second Life and OpenSimulator, provide an enormous range of possibilities for evaluating new ways of communication given that users inside this world can explore, meet other residents, socialize, participate in individual and group activities, and create and trade virtual property and services with one another, or travel throughout the world. Our research has been focused on showing that the integration of Speech Technologies and Natural Language Processing in virtual worlds is possible so that it emulates human face-to-face conversation.

We have proposed an architecture to develop conversational metabots. The metabots' conversational behavior is based on a statistical dialog model that can be trained using a corpus of dialogs, and improved by means of dialog and user simulation techniques. The system response of the conversational metabot is selected by means of two classification processes according to the probability distributions provided by neural networks that consider the complete history of the dialog. This statistical methodology can be employed to generate metabots with different conversational behavior and which are able to maintain a conversation in different application domains. In addition, the integration of user profiles and the emotional content extracted from the user's utterances allow the conversational interface to select the next system response considering these valuable information sources, thus improving the dialog strategy to select user's adapted system responses

Our proposal has been employed to generate the Demic conversational metabot, which provides academic information in the Second Life and OpenSimulator virtual worlds. The behavior of the bot has been trained over a set of real and simulated dialogs and can modify its dialog strategy by detecting new correct answers that were not defined in the initial dialogs. Demic has been evaluated according to its performance and the satisfaction of the users after interacting with it. The results show high success rates, reduced average number of dialog turns and improved confirmation and error correction rates. Also, users reported higher satisfaction when compared to a non-adaptive version of the bot and found it easier to obtain the information they were seeking.

For future work we intend to study the differences between the conversational models generated in this paper and the ones that could be obtained when the user interacts directly without using an avatar. This way, we plan to study the similarities and differences in the behavior of the user when influenced by the image of their avatar compared to their usual conversational behavior. Additionally, we want to evaluate the effect of considering the user profile as proposed, compared to a baseline system that does not incorporate such information. Finally, the combination of virtual worlds and virtual reality systems has been proposed as one of the most important challenges during the next years to facilitate a user's full immersion experience. This challenge implies improving also the integration of non-verbal communication features (e.g., improved recognition and generation of gestures) in addition to the recognition of emotions.

Acknowledgements

Work partially supported by Projects TRA2015-63708-R and TRA2016-78886-C3-1-R.

References

[1] P. Aparicio-Martínez, A. Perea-Moreno, M. Martínez-Jiménez, I. S.-V. Varo, M. Vaquero-Abellán, Social networks' unnoticed influence on body image in

Spanish university students, Telematics and Informatics 34 (8) (2017) 1685–692.

[2] J. Penni, The future of online social networks (OSN): A measurement analysis using social media tools and application, Telematics and Informatics 34 (5) (2017) 498–517.

[3] Nielsen, Social media report on social studies: A look at the social lanscape, The Nielsen Company, 2016.

[4] M. Gross, Exploring virtual worlds, Current Biology 27 (11) (2017) 399–402.

[5] R. Hooi, H. Cho, Virtual world continuance intention, Telematics and Informatics 34 (8) (2017) 1454–1464.

[6] K. S. Hale, K. M. Stanney, Handbook of Virtual Environments. Design, Implementation, and Applications, Taylor and Francis, 2014.

[7] S. D'Agustino, Immersive Environments, Augmented Realities, and Virtual Worlds: Assessing Future Trends in Education, IGI Global, 2013.

[8] H. Lin, H. Wang, Avatar creation in virtual worlds: Behaviors and motivations, Computers in Human Behavior 34 (2014) 213–218.

[9] S. Turkle, Reclaiming conversation. The power of talk in a digital age, Penguin Press, 2015.

[10] F. Li, T. C. Du, The effectiveness of word of mouth in offline and online social networks, Expert Systems with Applications 88 (2017) 338–351.

[11] D. Boyd, N. Ellison, Social Network Sites, Definition, History and Scholarship, Journal of Computer Mediated Communication 13 (1) (2007) 210–230.

[12] R. Dunbar, V. Arnaboldi, M. Conti, A. Passarella, The structure of online social networks mirrors those in the offline world, Social Network 43 (2015) 39–47.

[13] B. Nelson, B. Erlandson, Design for learning in Virtual Worlds: Interdisciplinary approaches to educational technology, Routledge, 2014.

[14] Bainbridge, The scientific research potential of virtual worlds, Science 27 (2007) 472–476.

[15] T. Boellstorff, Coming of Age in Second Life. An Anthropologist Explores the Virtually Human, Princeton University Press, 2015.

[16] E. LaPensee, J. Lewis, Nonverbal Communication in Virtual Worlds: Understanding and Designing Expressive Characters, ETC Press, 2014, Ch. Time-Traveller: first nations nonverbal communication in Second Life, pp. 105–119.

[17] D. Kirschner, J. Williams, Nonverbal Communication in Virtual Worlds: Understanding and Designing Expressive Characters, ETC Press, 2014, Ch. A microsociological perspective on non-verbal communicative strategies in MMORPGS, pp. 307–322.

[18] M. F. McTear, Z. Callejas, D. Griol, The Conversational Interface: Talking to Smart Devices, Springer, 2016.

[19] R. Pieraccini, L. Rabiner, The Voice in the Machine: Building Computers that Understand Speech, The MIT Press, 2012.

[20] M. F. McTear, Future and Emerging Trends in Language Technology. Machine Learning and Big Data: Second International Workshop, FETLT 2016, Springer International Publishing, 2017, Ch. The Rise of the Conversational Interface: A New Kid on the Block?, pp. 38–49.

[21] S. Young, M. Gasic, B. Thomson, J. Williams, POMDP-based Statistical Spoken Dialogue Systems: a Review, Proceedings of IEEE 101 (5) (2013) 1160–1179.

[22] T. Paek, R. Pieraccini, Automating spoken dialogue management design using machine learning: An industry perspective, Speech Communication 50 (8–9) (2008) 716–729.

[23] D. Griol, Z. Callejas, R. López-Cózar, G. Riccardi, A domain-independent statistical methodology for dialog management in spoken dialog systems, Computer Speech & Language 28 (3) (2014) 743–768.

[24] S. Young, J. Schatzmann, K. Weilhammer, H. Ye, The Hidden Information State Approach to Dialogue Management, in: Proc. of 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 4, Honolulu, Haway (USA), 2007, pp. 149–152.

[25] S. Young, The Statistical Approach to the Design of Spoken Dialogue Systems, Tech. rep., CUED/F-INFENG/TR.433, Cambridge University Engineering Department, Cambridge (UK) (2002).

[26] M. Berger, A. H. Jucker, M. A. Locher, Interaction and space in the virtual world of second life, Journal of Pragmatics 101 (2016) 83–100.

[27] Statista, Virtual Reality (VR) - Statistics & Facts, Statista. Available in: https://www.statista.com/topics/2532/virtual-reality-vr/, 2017.

[28] A. Cox, Virtual World Consumer Behavior, in: Proc. of ACM SIGMIS Conference on Computers and People Research, Alexandria, Virginia, USA, 2016, pp. 1–2.

[29] S. J. Barnes, J. Mattsson, N. Hartley, Assessing the value of real-life brands in virtual worlds, Technological Forecasting and Social Change 92 (2015) 12–24.

[30] T. Mikropoulos, A. Natsis, Educational virtual environments: A ten-year review of empirical research (1999-2009), Computers and Education 56 (3) (2011) 769–780.

[31] M. F. McTear, Spoken Dialogue Technology: Towards the Conversational User Interface, Springer, 2004.

[32] B. Reeves, C. Nass, Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places, Cambridge University Press, Cambridge, UK, 1996.

[33] J. Cassell, Embodied conversational agents. representation and intelligence in user interfaces, AI Magazine 2 (4) (2001) 67–84.

[34] J. Martínez-Miranda, Embodied conversational agents for the detection and prevention of suicidal behaviour: Current applications and open challenge, Journal of Medical Systems 41 (9) (2017) 135.

[35] T. Bickmore, J. Cassell, Advances in Natural Multimodal Dialogue Systems, Springer, 2005, Ch. Social Dialongue with Embodied Conversational Agents, pp. 23–54.

[36] O. J. Romero, R. Zhao, J. Cassell, Cognitive-Inspired Conversational-Strategy Reasoner for Socially-Aware Agents, in: Proc. of the Int. Joint Conference on Artificial Intelligence (IJCAI'17), Melbourne, Australia, 2017, pp. 3807–3813.

[37] N. Yee, J. Bailenson, K. Rickertsen, A meta-analysis of the impact of the inclusion and realism of human-like faces on user experiences in interfaces, in: Proc. of the SIGCHI Conference on Human factors in computing systems, San Jose, CA, USA, 2007, pp. 1–10.

[38] V. Groom, C. Nass, T. Chen, A. Nielsen, J. K.Scarborough, E. Robles, Evaluating the effects of behavioral realism in embodied agents, International Journal on Human-Computer Studies (2009) 842–849.

[39] A. Gulz, Benefits of virtual characters in computer based learning environments: Claims and evidence, International Journal of Artificial Intelligence in Education 14 (3) (2004) 313–334.

[40] B. S. Hasler, P. Tuchman, D. Friedman, Virtual research assistants: Replacing human interviewers by automated avatars in virtual worlds, Computers in Human Behavior 29 (4) (2013) 1608–1616.

[41] A. Cruz, H. Paredes, B. Fonseca, L. Morgado, P. Martins, Can presence improve collaboration in 3D Virtual Worlds, Procedia Technology 13 (2014) 47–55.

[42] I. Krasonikolakis, A. Vrechopoulos, A. Pouloudi, Store selection criteria and sales prediction in virtual worlds, Information & Management 51 (6) (2014) 641–652.

[43] S. Gregory, Virtual Worlds for Online Learning : Cases & Applications, Nova Science, 2015.

[44] N. Magnenat-Thalmann, H.-S. Kim, A. Egges, S. Garchery, Believability and interaction in virtual worlds, in: Proc. of 11th International Multimedia Modelling Conference (MMM'05), Melbourne, Australia, 2005, pp. 2–9.

[45] Y. G. Zhang, Y. M. Dang, S. A. Brown, H. Chen, Investigating the impacts of avatar gender, avatar age, and region theme on avatar physical activity in the virtual world, Computers in Human Behavior 68 (2017) 378–387.

[46] K. Gabriels, C. J. D. Backer, Virtual gossip: How gossip regulates moral life in virtual worlds, Computers in Human Behavior 63 (2016) 683–693.

[47] T. Partala, Psychological needs and virtual worlds: Case second life, International Journal of Human-Computer Studies 69 (12) (2011) 787–800.

[48] A. M. Grinberg, J. S. Careaga, M. R. Mehl, M.-F. O'Connor, Social engagement and user immersion in a socially based virtual world, Computers in Human Behavior 36 (2014) 479–486.

[49] R. C. Hubal, D. H. Fishbein, M. S. Sheppard, M. J. Paschall, D. L. Eldreth, C. T. Hyde, How do varied populations interact with embodied conversational

agents? Findings from inner-city adolescents and prisoners, Computers in Human Behavior (2008) 1104–1138.

[50] H. van Vugt, E. Konijn, J. Hoorn, I. Keur, A. Eliens, Realism is not all! User engagement with task-related interface characters, Interacting with Computers 19 (2) (2007) 267–280.

[51] A. Abdullah, Language and Virtual Identity in Second Life: An Ethnographic Sociolinguistic Study, Lambert Academic Publishing, 2016.

[52] M. Locher, A. Jucker, M. Berger, Negotiation of space in Second Life newbie interaction, Discourse Context Media 9 (2015) 34–45.

[53] J. Martin, Nonverbal Communication in Virtual Worlds: Understanding and Designing Expressive Characters, ETC Press, 2014, Ch. A few choice animations: nonverbal communication through production and consumption in Second Life, pp. 291–305.

[54] J. Green, A. Wyllie, D. Jackson, Virtual worlds: A new frontier for nurse education?, Collegian 21 (2) (2014) 135–141.

[55] A. M. Lomanowska, M. J. Guitton, My avatar is pregnant! representation of pregnancy, birth, and maternity in a virtual world, Computers in Human Behavior 31 (2014) 322–331.

[56] Y. Jung, S. Pawlowski, The meaning of virtual entrepreneurship in social virtual worlds, Telematics and Informatics 32 (1) (2015) 193–203.

[57] N. Ahern, D. Wink, Virtual learning environments: Second life, Nurse Educator 15 (6) (2010) 225–227.

[58] J. Cruz-Benito, R. Therón, F. J. García-Penalvo, E. Pizarro-Lucas, Discovering usage behaviors and engagement in an educational virtual world, Computers in Human Behavior 47 (2015) 18–25.

[59] M. Gustafsson, C. Englund, G. Gallego, The description and evaluation of virtual worlds in clinical pharmacy education in Northern Sweden, Currents in Pharmacy Teaching and Learning 9 (5) (2017) 887–892.

[60] R. López-Cózar, Z. Callejas, ASR post-correction for spoken dialogue systems based on semantic, syntactic, lexical and contextual information, Computer Speech and Language 50 (8-9) (2008) 745–766.

[61] L. Rabiner, B. Juang, C. Lee, An overview of automatic speech recognition, in: K. A. Publishers (Ed.), Automatic Speech and speaker Recognition: Advanced Topic, 1996, pp. 1–30.

[62] R. López-Cózar, Z. Callejas, D. Griol, ASR post-correction for spoken dialogue systems based on semantic, syntactic, lexical and contextual information, Knowledge-Based Systems 23 (5) (2010) 471–485.

[63] C. Lee, S. Jung, K. Kim, D. Lee, G. Lee, Recent approaches to dialog management for spoken dialog systems, Journal of Computing Science and Engineering 4 (1) (2010) 1–22.

[64] M. Walker, A. Stent, F. Mairesse, R. Prasad, Individual and domain adaptation in sentence planning for dialogue, Journal of Artificial Intelligence Research 30 (2007) 413–456.

[65] R. Hoffmann, Speech Synthesis: An Introduction for Engineers (Signals and Communication Technology), Springer, 2012.

[66] D. Griol, L. Hurtado, E. Segarra, E. Sanchis, A Statistical Approach to Spoken Dialog Systems Design and Evaluation, Speech Communication 50 (8–9) (2008) 666–682.

[67] D. E. Rumelhart, G. E. Hinton, R. J. Williams, PDP: Computational models of cognition and perception, I, MIT Press, 1986, Ch. Learning internal representations by error propagation, pp. 319–362.

[68] K. Ravi, V. Ravi, A survey on opinion mining and sentiment analysis: Tasks, approaches and applications, Knowledge-Based Systems 89 (2015) 14–46.

[69] J. A. Balazs, J. D. Velásquez, Opinion mining and information fusion: A survey, Information Fusion 27 (2016) 95–110.

[70] C. Clavel, Z. Callejas, Sentiment analysis: From opinion mining to human-agent interaction, IEEE Transactions on Affective Computing 7 (1) (2016) 74–93.

[71] D. Griol, J. Molina, Z. Callejas, Combining speech-based and linguistic classifiers to recognize emotion in user spoken utterances, Neurocomputing In press.

[72] F. Burkhardt, M. van Ballegooy, K. Engelbrecht, T. Polzehl, J. Stegmann, Emotion detection in dialog systems - Usecases, strategies and challenges,

in: Proc. of International Conference on Affective Computing and Intelligent Interaction (ACII'09), 2009, pp. 1–6.

[73] Z. Callejas, R. López-Cózar, Implementing modular dialogue systems: a case study, in: Proc. of Applied Spoken Language Interaction in Distributed Environments (ASIDE'05), Aalborg, Denmark, 2005, pp. 1–4.

[74] Z. Callejas, R. López-Cózar, Relations between de-facto criteria in the evaluation of a spoken dialogue system, Speech Communication 50 (8–9) (2008) 646–665.

[75] M. Dinarelli, Spoken language understanding: from spoken utterances to semantic structures, Ph.D. thesis, University of Trento, Trento (Italy) (2010).

[76] F. Mairesse, M. Gasic, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, S. Young, Spoken language understanding from unaligned data using discriminative classification models, in: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09), Taipei, Taiwan, 2009, pp. 4749–4752.

[77] Y. He, S. Young, Spoken language understanding using the Hidden Vector State model, Speech Communication 48 (3–4) (2006) 262–275.

[78] C. Raymond, G. Riccardi, Generative and discriminative algorithms for spoken language understanding, in: Proc. of the 8th Annual Conference of the International Speech Communication Association (Interspeech 2007), Antwerp, Belgium, 2007, pp. 1605–1608.

[79] J. Lafferty, A. McCallum, F. Pereira, onditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proc. of the 18th International Conference on Machine Learning (ICML'01), Williamstown, MA, USA, 2001, pp. 282–289.

[80] K. Macherey, O. Bender, H. Ney, Applications of statistical machine translation approaches to spoken language understanding, IEEE Transactions on Speech Audio Processing 17 (4) (2009) 803–818.

[81] S. Espana, F. Zamora, M. Castro, J. Gorbe, Efficient bp algorithms for general feedforward neural networks, Lecture Notes in Computer Science 4527 (2007) 327–336.

[82] D. Griol, J. Carbo, J. M. Molina, A statistical simulation technique to develop and evaluate conversational agents, Applied Artificial Intelligence 26 (4) (2013) 355–371.

[83] H. Ai, A. Raux, D. Bohus, M. Eskenazi, D. Litman, Comparing Spoken Dialog Corpora Collected with Recruited Subjects versus Real Users, in: Proc. of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium, 2007, pp. 124–131.

[84] J. Schatzmann, K. Georgila, S. Young, Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems, in: Proc. of the 6th SIGdial Workshop on Discourse and Dialogue, Lisbon (Portugal), 2005, pp. 45–54.