

This document is published in:

Abraham, A. et al. (eds.) (2011). *International Symposium on Distributed Computing and Artificial Intelligence*. (Advances in Intelligent and Soft Computing, 91), Springer, 69-76.

DOI: http://dx.doi.org/10.1007/978-3-642-19934-9_9

© 2011 Springer-Verlag Berlin Heidelberg

Agent Simulation to Develop Interactive and User-Centered Conversational Agents

David Griol, Javier Carbó, and José M. Molina

Group of Applied Artificial Intelligence (GIAA), Computer Science Department, Carlos III University of Madrid

e-mail: {david.griol,javier.carbo,josemanuel.molina}@uc3m.es

Abstract. In this paper, we present a technique for developing user simulators which are able to interact and evaluate conversational agents. Our technique is based on a statistical model that is automatically learned from a dialog corpus. This model is used by the user simulator to provide the following answer taking into account the complete history of the interaction. The main objective of our proposal is not only to evaluate the conversational agent, but also to improve this agent by employing the simulated dialogs to learn a better dialog model. We have applied this technique to design and evaluate a conversational agent which provides academic information in a multi-agent system. The results of the evaluation show that the conversational agent reduces the time needed to fulfill to complete the the dialogs, thereby allowing the conversational agent to tackle new situations and generate new coherent answers for the situations already present in an initial model.

1 Introduction

As we move towards a world where all the information is in the digital domain, it becomes necessary to provide straightforward ways of retrieving it. To achieve this goal it is necessary to provide an effective, easy, save and transparent interaction between the user and the system. Thus, it is important to identify which modality or combination of modalities would be optimal to present the information and interact with the user. To do so, in the last years there has been an increasing interest in simulating human-to-human communication, including the so-called conversational agents in multi-agents system [8]. There is a high variety of applications in

Funded by projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485), and DPS2008-07029-C02-02.

which conversational agents can be used, one of the most wide-spread of which is information retrieval. One of the most recent applications of these agents is for the development of e-learning and tutoring systems [7].

Multi-agent systems are designed as a collection of interacting autonomous agents, each having their own capacities and goals that are situated to a common environment. This way, the development of multi-agent systems offers the capability of simulating autonomous agents and the interaction between them. In the literature, there are several corpus-based approaches for developing user simulators, learning optimal dialog strategies, and evaluating conversational agents [10, 9, 6, 3]. The construction of user models based on statistical methods has provided interesting and well-founded results in recent years and is currently a growing research area. A probabilistic model to emulate the user agent can be trained from a corpus of human-computer dialogs to simulate user answers. Therefore, it can be used to learn a dialog strategy by means of its interaction with the conversational agent. A summary of user simulation techniques for reinforcement learning of the dialog strategy can be found in [9].

In this paper, we present a technique to develop a user agent simulator to automatically interact with a conversational agent and generate the dialogs required to learn an enhanced dialog model for a conversational agent. Our user simulation technique is based on a classification process in which a neural network is employed to take into account the previous dialog history to select the next user answer. We have applied this technique to develop a conversational agent which provides academic information in Spanish. The results of the evaluation of the conversational agent show that the conversational agent reduces the time needed to fulfill the different tasks, thereby allowing the conversational agent to tackle new situations and generate better answers for the situations already present in an initial model.

2 Design of an Academic Conversational Agent

The design of our conversational agent is based on the requirements defined for a dialog system developed to provide spoken access to academic information about the Department of Languages and Computer Systems in the University of Granada [1]. To successfully manage the interaction with the users, the conversational agent carry out six main tasks: automatic speech recognition (ASR), natural language understanding (NLU), dialog management (DM), database access and storage (DB), natural language generation (NLG), and text-to-speech synthesis (TTS).

The dialog manager of the the conversational agent has been developed using VoiceXML documents that are dynamically created using PHP. This way, it can adapt the system responses to the context of the conversation and the dialog state, which improves the naturalness of the interaction. For example, the help messages provided by the conversational agent take into account the topic that the user and the agent are addressing at a particular moment. The context is used as well to decide the confirmation strategy to use. In addition, we have implemented a statistical module

to automatically select the next system response (i.e., a VoiceXML file) by using a model which is learned from a dialog corpus for the task [2].

The information that the conversational agent provides has been classified in four main groups: subjects, professors, doctoral studies and registration. The information that the agent provides for each of these categories is shown in Table 1. As can be observed, the conversational agent must have gathered some data by asking the user about the name of the subjects, the professors, etc. The way in which the user is queried for this information follows in most cases a system-directed initiative.

Table 1 Information provided by the academic conversational agent

Category	Information provided by the user (names and examples)	Information provided by the system
Subject	<i>Name</i>	Compilers
	<i>Degree</i> , in which it is taught in case that there are several subjects with the same name	Computer Science
	<i>Group name</i> and optionally <i>type</i> , in case he asks for information about a specific group	A Theory A
Lecturers	Any combination of <i>name</i> and <i>surnames</i>	John John Smith Mr. Smith
	Optionally <i>semester</i> , in case he asks for the tutoring hours	First semester Second semester
Doctoral studies	Name of a doctoral program	Software development
	Name of a course if he asks for information about a specific course	Object-Oriented Programming
Registration	Name of the deadline	Provisional registration confirmation

As in many other conversational agents, the semantic representation that we have chosen for the task is based on the concept of frame, 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 to the system (*Subject*, *Lecturers*, *Doctoral studies*, *Registration*), three task-independent concepts (*Affirmation*, *Negation*, and *Not-Understood*), and eight

attributes (*Subject-Name, Degree, Group-Name, Subject-Type, Lecturer-Name, Program-Name, Semester, and Deadline*). The labeling of the system turns is similar to the labeling defined for the user turns. A total of 30 task-dependent concepts was 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 (*Subject, 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*)

A set of 100 dialogs was acquired with the conversational agent by means of its interaction with real users using an initial version of the system [1]. The acquisition process resulted in a spontaneous Spanish speech dialog corpus with 60 different speakers. The total number of user turns was 422. The recorded material has a duration of 150 minutes. Figure 1 shows an example of a dialog extracted from this corpus. Turns with S refer to system turns, and turns with U refer to user turns.

<p>S1: Welcome to the university information system. You can consult information about subjects, lecturers, doctoral studies and registrations.</p> <p>U1: I want to know information about subjects in Computer Science.</p> <p>S2: Which subject do you want to consult?</p> <p>U2: Language Processors I.</p> <p>S3: Here you can consult the information of the subject. Anything else?</p> <p>U3: Can you inform me about the schedules of Group D?</p> <p>S4: I give the information for the Group D of Language Processors I: Monday 9:00-11:00, Wednesday 11:00-13:00. Anything else?</p> <p>U4: Can you tell me the name of the lecturer?</p> <p>S5: I give the information for the Group D of Language Processors I. The name of the lecturer is David Griol. Anything else?</p> <p>U5: No, thank you.</p> <p>S6: Thank you for using the system.</p>
--

Fig. 1 Example of a dialog acquired with the interaction of the conversational agent with real users

3 Our Proposal to Develop a User Agent Simulator

Our proposed architecture to provide context-aware services by means of conversational agents is described in [5]. It consists of five different types of agents that

cooperate to provide an adapted service. *User agents* are configured into mobile devices or PDAs. *Provider Agents* supply the different services in the system and are bound to *Conversational Agents* that provide the specific services. A *Facilitator Agent* links the different positions to the providers and services defined in the system. A *Positioning Agent* communicates with the ARUBA positioning system to extract and transmit positioning information to other agents in the system. Finally, a *Log Analyzer Agent* generates user profiles that are used by Conversational Agents to adapt their behaviour taking into account the preferences detected in the users' previous dialogs.

The user simulator replaces the user agent in our architecture. This agent simulates the user intention level, that is, the simulator provides concepts and attributes that represent the intention of the user utterance. Therefore, the user simulator carries out the functions of the ASR and NLU modules, i.e., it generates frames in the same format defined for the output of the NLU module.

The methodology that we have developed for user simulation extends our work for developing a statistical methodology for dialog management [4]. The user answers are generated taking into account the information provided by the simulator throughout the history of the dialog, the last system turn, and the objective(s) pre-defined for the dialog.

In order to control the interaction, our user simulator uses the representation the dialogs as a sequence of pairs (A_i, U_i) , where A_i is the output of the dialog system (the system answer) at time i , expressed in terms of dialog acts; and U_i is the semantic representation of the user turn (the result of the understanding process of the user input) at time i , expressed in terms of frames. This way, each dialog is represented by $(A_1, U_1), \dots, (A_i, U_i), \dots, (A_n, U_n)$, where A_1 is the greeting turn of the system (the first turn of the dialog), 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 .

In this framework, we consider that, at time i , the objective of the dialog manager is to find an appropriate user answer U_i . This selection is a local process for each time i and takes into account the sequence of dialog states that precede time i , the system answer at time i , and the objective of the dialog \mathcal{O} . If the most probable user answer U_i is selected at each time i , the selection is made using the maximization:

$$\hat{U}_i = \underset{U_i \in \mathcal{U}}{\operatorname{argmax}} P(U_i | S_1, \dots, S_{i-1}, A_i, \mathcal{O})$$

where set \mathcal{U} contains all the possible user answers.

As the number of possible sequences of states is very large, we establish a partition in this space (i.e., in the history of the dialog preceding time i). This data structure, that we call *User Register (UR)*, contains the information provided by the user throughout the previous history of the dialog. After applying the above considerations and establishing the equivalence relations in the histories of the dialogs, the selection of the best U_i is given by:

$$\hat{U}_i = \underset{U_i \in \mathcal{U}}{\operatorname{argmax}} P(U_i | UR_{i-1}, A_i, \mathcal{O})$$

We propose the use of a multilayer perceptron (MLP) to make the assignment of a user turn. The input layer receives the current situation of the dialog, which is represented by the term $(UR_{i-1}, A_i, \mathcal{O})$ in the previous equation. The values of the output layer can be viewed as the a posteriori probability of selecting the different user answers defined for the simulator given the current situation of the dialog. The choice of the most probable user answer of this probability distribution leads to the previous equation. In this case, the user simulator will always generate the same answer for the same situation of the dialog. Since we want to provide the user simulator with a richer variability of behaviors, we base our choice on the probability distribution supplied by the MLP on all the feasible user answers.

A real corpus includes information about the errors that were introduced by the ASR and the NLU modules during the acquisition. This information also includes confidence measures, which are used by the conversational agent to evaluate the reliability of the concepts and attributes generated by the NLU module. This way, an error simulator agent has been designed to perform error generation. This agent modifies the frames generated by the user simulator once the UR is updated. In addition, the error simulator adds a confidence score to each concept and attribute in the frames.

4 Results of the Evaluation

A dialog corpus of 3000 successful dialogs was acquired using the proposed user simulation technique following the same set of scenarios defined for the acquisition with real users. A maximum number of 14 user turns per dialog was defined for the acquisition. A user request for closing the dialog is selected once the system has provided the information defined in the objective(s) of the dialog.

We have considered three dimensions in order to evaluate the initial conversational agent and its evolution once the simulated dialogs are incorporated to learn a new dialog model: high-level features (dialog and turn lengths), dialog style (speech-act frequency and proportion of goal-directed actions), and dialog efficiency (goal completion rates and times). Table 2 shows the comparison of the different high-level measures defined for the evaluation. As it can be seen, after the incorporation of the simulated dialogs there is a reduction in the average number of turns required to fulfill the complete set of objectives defined in the scenarios. This reduction can also be observed in the number of turns of the longest, shortest and most seen dialogs. Figure 2 shows the previously described evolution of the average duration in terms of total dialog turns (*duration*). It also shows the reduction in the number of

Table 2 High-level dialog features defined for the comparison of the conversational agents

	Initial Convers. Agent	Final Convers. Agent
Average number of user turns per dialog	4.99	3.75
Number of turns of the most seen dialog	2	2
Number of turns of the shortest dialog	2	2
Number of turns of the longest dialog	14	12

responses provided by the conversational agent which cause a failure of the dialog ($\#error$) and the number of unseen situations for which there is not a system response in the dialog model ($\#unseen$). As it can be seen from these results, the final conversational agent not only reduces the time required to fulfill each one of the objectives of the dialog, but also it reduces the possibility of selecting an erroneous response.

Finally, Table 3 shows the frequency of the most dominant user and system dialog acts in the initial and final conversational agents. From its comparison, it can be observed that there are significant differences in the dialog acts distribution. With regard to user actions, it can be observed that users need to employ less confirmation turns in the final agent, which explains the higher proportion for the rest of user actions using the final conversational agent. It also explains the lower proportion of yes/no actions in the final agent, which are mainly used to confirm that the system's query has been correctly provided. With regard to the system actions, it can be observed a reduction in the number of system confirmations for data items.

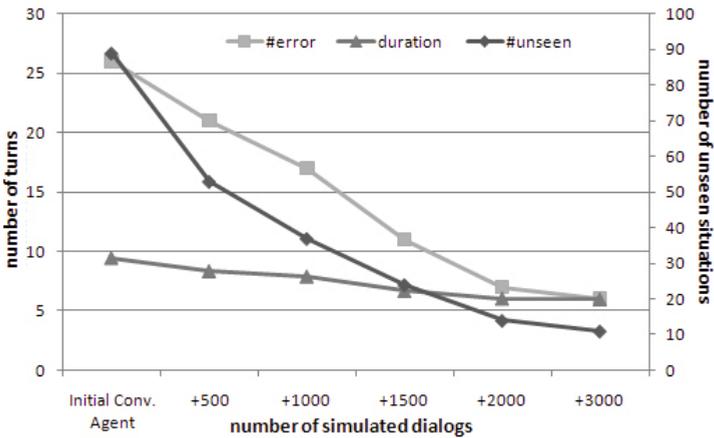


Fig. 2 Evolution of the number of unseen situations, number of errors, and average number of turns

Table 3 Percentages of different types of user [up] and system [down] dialog acts

	Initial Convers. Agent	Final Convers. Agent
Request to the system	31.74%	35.43%
Provide information	20.72%	24.98%
Confirmation	10.81%	7.34%
Yes/No answers	31.47%	28.77%
Other answers	3.26%	3.48%

	Initial Convers. Agent	Final Convers. Agent
Confirmation of concepts and attributes	13.51%	10.23%
Questions to require information	18.44%	19.57%
Answers generated after a database query	68.05%	70.20%

This explains a higher proportion of turns to inform and provide data items for the final agent. Both results show that the final conversational agent carries out a better selection of the system responses.

5 Conclusions

In this paper, we have described a technique for simulating user agents and evaluate conversational agents. Our technique is based on a statistical model which takes the complete history of the interaction into account to decide the next user answer. This decision is modeled by a classification process in which a neural network is used. In addition, the simulated dialogs are used to automatically reinforce the dialog model of the conversational agent. We have described the application of this technique to develop an enhanced academic conversational agent. The results of the evaluation of this agent show that the proposed user simulation methodology can be used not only to evaluate Conversational agents but also to explore new enhanced dialog strategies. As a future work, we are adapting the proposed user simulation technique for its application in more difficult domains in our multi-agent architecture.

References

1. Callejas, Z., López-Cózar, R.: Relations between de-facto criteria in the evaluation of a spoken dialogue system. *Speech Communication* 50(8-9), 646–665 (2008)
2. Griol, D., Callejas, Z., López-Cózar, R.: Statistical Dialog Management Methodologies for Real Applications. In: *Proc. of the 11th Sigdial Meeting*, pp. 124–131 (2010)
3. Griol, D., Hurtado, L., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. *Speech Communication* 50(8-9), 666–682 (2008)
4. Griol, D., Riccardi, G., Sanchis, E.: Learning the Structure of Human-Computer and Human-Human Dialogs. In: *Proc. of the 10th Interspeech Conference*, pp. 2775–2778 (2009)
5. Griol, D., Sánchez-Pi, N., Carbó, J., Molina, J.M.: An Architecture to Provide Context-Aware Services by Means of Conversational Agents. In: de Leon F. de Carvalho, A.P., Rodríguez-González, S., De Paz Santana, J.F., Rodríguez, J.M.C. (eds.) *Distributed Computing and Artificial Intelligence*. AISC, vol. 79, pp. 275–282. Springer, Heidelberg (2010)
6. Lemon, O., Liu, X.: Dialogue Policy Learning for Combinations of Noise and User Simulation: Transfer Results. In: *Proc. of the 8th Sigdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, pp. 55–58 (2007)
7. Litman, D.J., Silliman, S.: ITSPOKE: An Intelligent Tutoring Spoken Dialogue System. In: *Proc. of the 4th HLT/NAACL Conference*, pp. 233–236 (2004)
8. McTear, M.F.: *Spoken Dialogue Technology: Towards the Conversational User Interface*. Springer, Heidelberg (2004)
9. Schatzmann, J., Weillhammer, K., Stuttle, M., Young, S.: A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Engineering Review* 21(2), 97–126 (2006)
10. Scheffler, K., Young, S.: Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In: *Proc. of HLT 2002 Conference*, pp. 12–18 (2001)