This document is published in:

Demazeau, Y. et al. (eds.) (2011) *Advances on Practical Applications of Agents and Multiagent Systems: 9th International Conference on Practical Applications of Agents and Multiagent Systems*, (Advances in Intelligent and Soft Computing, 88), Springer, 255-264.
DOI: http://dx.doi.org/10.1007/978-3-642-19875-5_33

An Agent-Based Dialog Simulation Technique to Develop and Evaluate Conversational Agents

David Griol, Nayat Sánchez-Pi, Javier Carbó, and José M. Molina

Group of Applied Artificial Intelligence (GIAA), Computer Science Department,
Carlos III University of Madrid e-mail:
{david.griol,nayat.sanchez,javier.carbo,josemanuel.molina}@uc3m.es

Abstract. In this paper, we present an agent-based dialog simulation technique for learning new dialog strategies and evaluate conversational agents. Using this technique the effort necessary to acquire data required to train the dialog model and then explore new dialog strategies is considerably reduced. A set of measures has also been defined to evaluate the dialog strategy that is automatically learned and compare different dialog corpora. We have applied this technique to explore the space of possible dialog strategies and evaluate the dialogs acquired for a conversational agent that collects monitored data from patients suffering from diabetes.

1 Introduction

Conversational agents have became a strong alternative to provide computers with intelligent and natural communicative capabilities. A conversational agent is a software that accepts natural language as input and generates natural language as output, engaging in a conversation with the user. 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).

The application of statistical approaches to the design of this kind of agents, specially regarding the dialog management process, has attracted increasing interest during the last decade [8]. Statistical models can be trained from real dialogs, modeling the variability in user behaviors. The final objective is to develop

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.

conversational agents that have a more robust behavior and are easier to adapt to different user profiles or tasks.

The success of these approaches depends on the quality of the data used to develop the dialog model. Considerable effort is necessary to acquire and label a corpus with the data necessary to train a good model. A technique that has currently attracted an increasing interest is based on the automatic generation of dialogs between the dialog manager and an additional module, called the user simulator, which represents user interactions with the conversational agent [7, 4]. This way, a very important application of the simulated dialogs is to support the automatic learning of optimal dialog strategies.

In this paper, we present an agent-based dialog simulation technique to automatically generate the data required to learn a new dialog model for a conversational agent. We have applied our technique to explore dialog strategies for the DI@L-log conversational agent, designed to collect monitored data from patients suffering from diabetes. In addition, a set of specific measures has been defined to evaluate the main characteristics of the acquired data and the new dialog strategy that can be learned from them. The results of the comparison of these measures for an initial corpus and a corpus acquired using the dialog simulation technique show how the quality of the dialog model is improved once the simulated dialogs are incorporated.

The remainder of the paper is organized as follows. Section 2 describes the proposed agent-based dialog generation technique and the measures used to evaluate the quality of dialogs acquired with different dialog strategies. Section 3 describes the DI@L-log conversational agent and the acquisition of a initial corpus for this task. Section 4 shows the results of the comparison of the measures for the two corpora acquired for the DI@L-log task. Finally, some conclusions and future work lines are described in Section 5.

2 Our Agent-Based Dialog Simulation Technique

Our proposed architecture to provide context-aware services by means of conversational agents is described in [3]. 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.

In this paper we focus on the simulation of the user and conversational agents to acquire a dialog corpus. In our dialog generation technique, both agents use a random selection of one of the possible responses defined for the semantics of the task (expressed in terms of user and system dialog acts). At the beginning of the simulation, the set of system responses is defined as equiprobable. When a successful dialog is simulated, the probabilities of the answers selected by the the conversational agent simulator during that dialog are incremented before beginning a new simulation.

An error simulation agent has been implemented to include semantic errors in the generation of dialogs. This module modifies the dialog acts by the user agent simulator once it has selected the information to be provided to the user. In addition, the error simulation module adds a confidence score to each concept and attribute in the semantic representation obtained from the user turn.

For the study presented in this paper, we have improved this agent using a model for introducing errors based on the method presented in [6]. The generation of confidence scores is carried out separately from the model employed for error generation. This model is represented as a communication channel by means of a generative probabilistic model $P(c, a_u|\tilde{a}_u)$, where a_u is the true incoming user dialog act $\tilde{a}_u$ is the recognized hypothesis, and c is the confidence score associated with this hypothesis.

The probability $P(\tilde{a}_u|a_u)$ is obtained by Maximum-Likelihood using the initial labeled corpus acquired with real users and considers the recognized sequence of words w_u and the actual sequence uttered by the user $\tilde{w}_u$. This probability is decomposed into a component that generates a word-level utterance from a given user dialog act, a model that simulates ASR confusions (learned from the reference transcriptions and the ASR outputs), and a component that models the semantic decoding process.

$$P(\tilde{a}_u|a_u) = \sum_{\tilde{w}_u} P(a_u|\tilde{w}_u) \sum_{w_u} P(\tilde{w}_u|w_u) P(w_u|a_u)$$

Confidence score generation is carried out by approximating $P(c|\tilde{a}_u, a_u)$ assuming that there are two distributions for c. These two distributions are handcrafted, generating confidence scores for correct and incorrect hypotheses by sampling from the distributions found in the training data corresponding to our initial corpus.

$$P(c|a_w, \tilde{a}_u) = \begin{cases} P_{corr}(c) & if \quad \tilde{a}_u = a_u \\ P_{incorr}(c) & if \quad \tilde{a}_u \neq a_u \end{cases}$$

The conversational agent simulator considers that the dialog is unsuccessful when one of the following conditions takes place: i) the dialog exceeds a maximum number of system turns slightly higher than the average number of turns of the dialogs acquired with real users; ii) the answer selected by the dialog manager in the conversational agent simulator corresponds to a query not made by the user simulator; iii) a query to the database generates an error because the user agent simulator has not provided the mandatory data needed to carry out the query; iv) the answer generator generates an error when the answer selected by the conversational agent simulator involves the use of a data item not provided by the user agent simulator.

A user request for closing the dialog is selected once the conversational agent simulator has provided the information defined in its objective(s). The dialogs that fulfill this condition before the maximum number of turns are considered successful.

Figure 1 shows the complete architecture for the proposed dialog simulation technique.

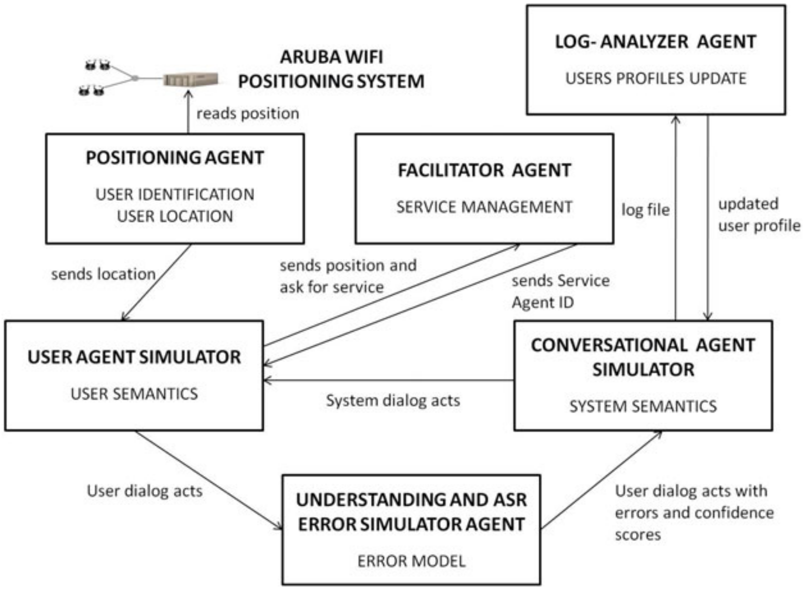

Fig. 1 Graphical scheme of the proposed agent-based dialog simulation technique

For the evaluation of the quality of the dialogs provided by a conversational agent, we have defined a set of quantitative evaluation measures based on prior work in the dialog literature [5, 1]. This set of proposed measures can be divided into two types:

- High-level dialog features: These features evaluate how long the dialogs last, how much information is transmitted in individual turns, and how active the dialog participants are.
- Dialog style/cooperativeness measures: These measures analyze the frequency of different speech acts and study what proportion of actions is goal-directed, what part is taken up by dialog formalities, etc.

Six high-level dialog features have been defined for the evaluation of the dialogs: the average number of turns per dialog, the percentage of different dialogs without considering the attribute values, the number of repetitions of the most seen dialog, the number of turns of the most seen dialog, the number of turns of the shortest dialog, and the number of turns of the longest dialog. Using these measures, we tried to evaluate the success of the simulated dialogs as well as its efficiency and variability with regard to the different services.

For dialog style features, we defined and counted a set of system/user dialog acts. On the system side, we have measured the confirmation of concepts and attributes, questions to require information, and system answers generated after a database query. 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.

3 Case Application: The DI@L-log Conversational Agent

DI@L-log is a spoken conversational agent which acts as a voice logbook to collect home monitored data from patients suffering from Type-2 diabetes [2]. The data collected by the system are the patient's weight, blood pressure (systolic and diastolic values) and sugar levels. The system validates and analyses the data, providing some immediate feedback to the patients regarding their current progress as well as communicating the results to doctors at the hospital who are able to review the patient's progress graphically and deal with any alerts generated by the system.

To apply our dialog simulation technique to this task, the definition of the semantics was carried out considering the information that is required to monitor the patients and inform them about their condition. Taking into account the requirements of the task, an initial strategy was designed for the DM. Figure 2 shows an example of a dialog acquired using this strategy.

S1: Welcome to the DI@L-log System. Please say or enter your four digit pin.
U1: 1356.
S2: Are you ready to give your measurements?
U2: Yes.
S3: Let's begin with your weight. How many stones?
U3: 12.
S4: I have understood 12 stones. Is it correct?
U4: Yes.
S5: What is your blood sugar?
U5: 11.
S6: I have understood 11. Is it correct?
U6: Yes.
S7: And finally, for your blood pressure you will be asked for the top number and then the lower number. What is the top number of your blood pressure?
U7: 160.
S8: I have understood 160. Is it correct?
U8: Yes.
S9: And what is the lower number?
U9: 86.
S10: I have understood 86. Is it correct?
U10: Yes.
S11: Your readings have been successfully updated. According to them, your weight is higher than last week, be very careful. Thanks for contacting DI@L-log. Goodbye.

Fig. 2 An example of a dialog for the DI@L-log task

As can be observed, three different phases are present in every dialog. Firstly, there is an identification phase in which the system asks the user about his login and password and then waits until the user says that he is ready to provide the control data (S1 and S2 system turns). Secondly, the system analyzes which data is required for the current user, taking into account that the weight and sugar values are mandatory and the blood control is only carried out for specific patients (S3 to S10 system turns). In this phase, the system requires the user to provide this data. Every item is confirmed after the user has provided its value. The user can only provide one item at a time. In the last phase, the system consults the information that the patient has provided during the current dialog and compares it with the data that is present in a database that contains the values that he provided in previous dialogs. By means of this comparison, the system is able to inform the user about his condition and provide him with instructions that take this into account (S11 system turn).

A corpus of 100 dialogs was acquired using this strategy. In order to learn statistical models, the dialogs of the corpus were labeled in terms of dialog acts. In the case of user turns, the dialog acts correspond to the classical frame representation of the meaning of the utterance. For the DI@L-log task, we defined three task-independent concepts (*Affirmation*, *Negation*, and *Not-Understood*) and four attributes (*Weight*, *Sugar*, *Systolic-Pressure*, and *Diastolic-Pressure*).

The labeling of the system turns is similar to the labeling defined for the user dialog acts. A total of twelve task-dependent concepts was defined, corresponding to the set of concepts used by the system to acquire each of the user variables (*Weight*, *Sugar*, *Systolic-Pressure*, and *Diastolic-Pressure*), concepts used to confirm the values provided by the user (*Confirmation-Weight*, *Confirmation-Sugar*, *Confirmation-Systolic*, and *Confirmation-Diastolic*), concepts used to inform the patient about his condition (*Inform*), and three task-independent concepts (*Not-Understood*, *Opening*, and *Closing*).

4 Evaluation Results

By employing the agent-based architecture proposed in Section 2 for dialog simulation, a set of 100,000 automatically labeled dialogs was acquired for the DI@-log task. Table 1 summarizes the statistics of the acquisition of this simulated corpus. A set of 11 different scenarios was defined to specify the objectives of the simulation, taking into account if the pressure values are necessary and different possibilities for the generation of errors and confidence measures. Given that the first and third phases of the dialog are mandatory and they have always the same structure, only the second phase in which the system collects the different values to monitor patients was taken into account for the simulation.

Table 1 Statistics of the corpus acquisition for the DI@L-log system

Simulated dialogs	100,000
Successful dialogs	27,521
Different dialogs	1,573

Figure 3 shows an example of a dialog from the acquired corpus. The objective defined for the dialog was to collect the weight, sugar and pressure values. The values defined in the scenario are 12, 11, 160, and 80 respectively. Confidence scores generated by the error simulator agent are shown between brackets. A sentence in natural language, that is equivalent to the labeled system turn selected by the conversational agent simulator, has been incorporated to clarify the explanation.

————————

LOGIN PHASE

————————

S1: (*Weight*)
Tell me your weight in stones?
U1: *Weight*: 12 [0.1]
S2: (*Confirmation-Weight*)
I have understood 12 stones. Is it correct?
U2: (*Acceptance*) [0.9]
S3: (*Sugar*)
What is your blood sugar?
U3: *Sugar*: 11 [0.9]
S4: (*Systolic*)
Tell me your blood systolic pressure.
U4: (*Systolic*): 150 [0.1]
S5: (*Confirmation-Systolic*)
I have understood 150. Is it correct?
U5: (*Rejection*) [0.9]
S6: (*Systolic*)
Tell me your systolic pressure.
U6: *Systolic*: 160 [0.9]
S7: (*Diastolic*)
And what is the lower number?
U7: *Diastolic*: 80 [0.9]

————————

DATA ANALYSIS - FINAL ANSWER

————————

Fig. 3 A dialog extracted from the simulated corpus of the DI@L-log task

In this dialog, the conversational agent simulator begins asking the user about his weight. As a low confidence measure is introduced for the value provided by the user agent simulator in U1, the conversational agent simulator decides to confirm this value in S2. Then, this agent asks for the sugar value. The user agent simulator provides this value in U3 and a high confidence measure is assigned.

The conversational agent simulator asks for the systolic pressure in S4. An error is introduced in the value provided by the error simulator agent for this parameter (it changes 160 to 150) and a low confidence measure is assigned to this value. Then, the conversational agent simulator asks the user agent simulator to confirm this value. The user agent simulator rejects this value in U5 and the conversational agent simulator decides to ask for it again. Finally, the conversational agent simulator asks for the diastolic pressure. This value is correctly introduced by the user agent simulator and the error simulator agent also assigns a high confidence level. Then, the conversational agent simulator obtains the data required from the patient, next the third phase of the dialog carries out the analysis of the condition of the patient and finally it informs him.

4.1 High-Level Dialog Features

The first group of experiments covers the following statistical properties to evaluate the quality of the dialogs obtained using different dialog strategies: i) Dialog length, measured as the number of turns per task; number of turns of the shortest dialog; number of turns of the longest dialog; and number of turns of the most seen dialog; ii) Different dialogs in each corpus, measured as the percentage of different dialogs (different labeling and/or order of dialog acts) and the number of repetitions of the most observed dialog; iii) Turn length, measured as the number of actions per turn; iv) Participant activity, measured as the ratio between system and user actions per dialog. Table 2 shows the comparison of the different high-level measures for the initial corpus and the corpus acquired incorporating the successfully simulated dialogs.

Table 2 Results of the high-level dialog features defined for the comparison of the dialogs for the initial and final strategy

	Initial Strategy	**Final Strategy**
Average number of turns per dialog	12.9±2.3	7.4±1.6
Percentage of different dialogs	62.9%	78.3%
Repetitions of the most seen dialog	18	3
User turns of the most seen dialog	9	7
User turns of the shortest dialog	7	5
User turns of the longest dialog	13	9

The first improvement that can be observed is the reduction in the number of turns. This reduction can also be observed in the number of turns of the longest, shortest and most seen dialogs. These results show that improving the dialog strategy makes it possible to reduce the number of necessary system actions. The greater variability of the resulting dialogs can be observed in the higher percentage of different dialogs and less repetitions of the most seen dialog obtained with the final dialog strategy.

We have observed that there is also a slight increment in the mean values of the turn length for the dialogs acquired with the final strategy. These dialogs are statistically longer, as they showed 1.6 actions per user turn instead of the 1.3 actions observed in the initial dialogs. This is also due to the better selection of the system actions Regarding the dialog participant activity, Figure 5 shows the ratio of user versus system actions. Dialogs in the final corpus have a higher proportion of system actions because the systems needs to make a smaller number of confirmations.

4.2 Dialog Style and Cooperativeness

The experiments described in this section cover the following statistical properties: frequency of different user and system actions (dialog acts), and proportion of goal-directed actions (request and provide information) versus grounding actions (confirmations). We consider as well the remaining possible actions. The histograms in Figure 4 show the frequency of the most dominant user and system dialog acts in

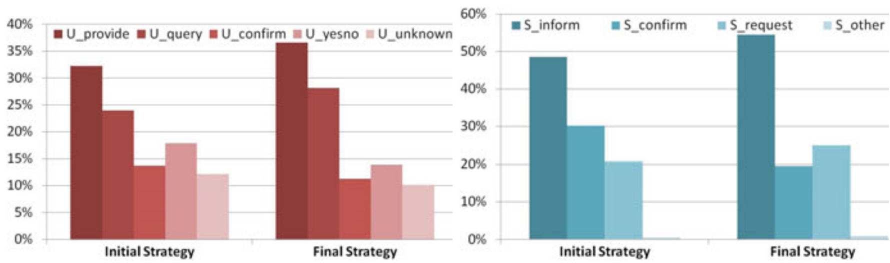

Fig. 4 Histogram of user dialog acts (left) and system dialog acts (right)

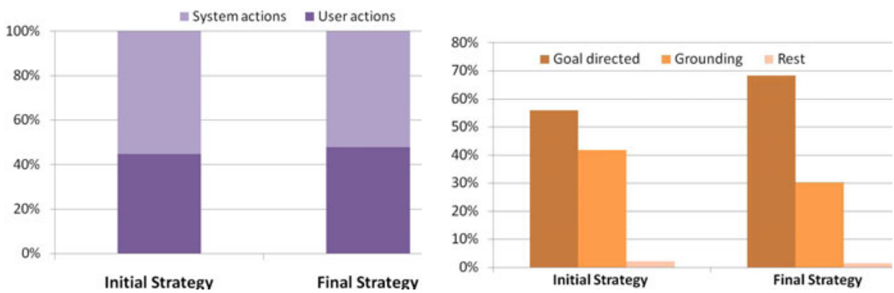

Fig. 5 Ratio of user versus system actions (left) and proportions of dialog spent on-goal directed actions, ground actions and the rest of possible actions (right)

the initial and final strategy. In both cases, significant differences in the dialog acts distribution can be observed.

With regard to user actions, it can be observed that users need to employ less confirmation turns in the final strategy, which explains the higher proportion for the rest of user actions in this strategy. It also explains the lower proportion of yes/no actions in the final strategy, which are mainly used to confirm that the system's service has been correctly provided. With regard to the system actions, it can be observed a reduction in the number of system requests for data items. This explains a higher proportion of turns to inform and confirm data items in the dialogs of the final strategy. Finally, we grouped user and system actions into categories in order to compare turns to request and provide information (goal directed actions) versus turns to confirm data items and make other actions (grounding actions), as shown in Figure 5. This study also shows the better quality of the dialogs in the final strategy, given that the proportion of goal-directed actions is higher in these dialogs.

5 Conclusions

In this paper, we have described a technique for exploring dialog strategies in conversational agents. Our technique is based on an automatic dialog simulation technique to generate the data that is required to re-train a dialog model. The results of applying our technique to the DI@L-log system, which follows a very strict initial interaction flow, show that the proposed methodology can be used to automatically explore new enhanced strategies. Carrying out these tasks with a non-automatic approach would require a very high cost that sometimes is not affordable. As a future work, we are adapting a previously developed statistical dialog management technique to learn a dialog manager for this task and evaluate the complete agent-based architecture with real users.

References

1. Ai, H., Raux, A., Bohus, D., Eskenazi, M., Litman, D.: Comparing Spoken Dialog Corpora Collected with Recruited Subjects versus Real Users. In: Proc. of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium, pp. 124–131 (2007)
2. Black, L., McTear, M.F., Black, N.D., Harper, R., Lemon, M.: Appraisal of a conversational artefact and its utility in remote patient monitoring. In: Proc. of the 18th IEEE Symposium CBMS 2005, Dublin, Ireland, pp. 506–508 (2005)
3. Griol, D., Sánchez-Pi, N., Carbó, J., Molina, J.: An Architecture to Provide Context-Aware Services by means of Conversational Agents. Advances in Intelligent and Soft Computing 79, 275–282 (2010)
4. Paek, T., Horvitz, E.: Conversation as action under uncertainty. In: Proc. of the 16th Conference on Uncertainty in Artificial Intelligence, San Francisco (USA), pp. 455–464 (2000)
5. Schatzmann, J., Georgila, K., Young, S.: Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems. In: Proc. of the 6th SIGdial Workshop on Discourse and Dialogue, Lisbon, Portugal, pp. 45–54 (2005)
6. Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., Young, S.: Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In: Proc. of Human Language Technologies HLT/NAACL 2007 Conference, Rochester, USA, pp. 149–152 (2007)
7. Schatzmann, J., Weilhammer, 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)
8. Young, S.: The Statistical Approach to the Design of Spoken Dialogue Systems. Tech. rep., CUED/F-INFENG/TR 433, Cambridge University Engineering Department, Cambridge, UK (2002)