

The DI@L-log System: Integration of Speech Technologies in Healthcare Applications

David Griol, José Manuel Molina

Universidad Carlos III de Madrid
Computer Science Department
Group of Applied Artificial Intelligence (GIAA)
28911 - Leganés (Spain)
dgriol@inf.uc3m.es, molina@ia.uc3m.es

Resumen In this paper, we describe a spoken dialog system developed to collect monitored data from patients suffering from diabetes. The dialog model of this system has been developed by means of a statistical methodology for automatically exploring the dialog space and learning new enhanced dialog strategies from a dialog corpus. A dialog simulation technique has been applied to acquire data required to train the dialog model and then explore the new dialog strategies. A set of measures has also been defined to evaluate the dialog strategy. The results of the evaluation show how the dialog manager deviates from the initially predefined strategy, allowing the dialog manager to tackle new situations and generate new coherent answers for the situations already present in the initial corpus. The proposed technique can be used not only to develop new dialog managers but also to explore new enhanced strategies.

Keywords: Dialog Systems, Healthcare Applications, Dialog Strategy, Dialog Simulation, Dialog Management

1. Introduction

A dialog system can be defined as a software that accepts natural language as input and generates natural language as output, engaging in a conversation with the user. In a dialog system of this kind, several modules cooperate to perform the interaction with the user: the Automatic Speech Recognizer (ASR), the Language Understanding Module (NLU), the Dialog Manager (DM), the Natural Language Generation module (NLG), and the Synthesizer (TTS). Each one of them has its own characteristics and the selection of the most convenient model varies depending on certain factors: the goal of each module, the possibility of manually defining the behavior of the module, or the capability of automatically obtaining models from training samples.

The application of statistical approaches to dialog management has attracted increasing interest during the last decade [10]. Statistical models can be trained from real dialogs, modeling the variability in user behaviors. The final objective is to develop dialog systems that have a more robust behavior and are easier to adapt to different user profiles or tasks. The most extended methodology for machine-learning of dialog strategies consists of modeling human-computer interaction as an optimization problem

using Partially Observable MDPs (POMDPs) and reinforcement methods [11]. However, they are limited to small-scale problems, since the state space would be huge and exact POMDP optimization would be intractable. Other interesting approaches for statistical dialog management are based on Hidden Markov Models (HMMs) [2] or Bayesian networks [6].

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 dialog system [9].

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 user model 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 dialog manager and reduce the effort required to acquire the corpus. In the literature, there are several corpus-based approaches for developing user simulators, learning optimal management strategies, and evaluating the dialog system [7, 3].

In this paper, we present a technique for learning optimal dialog strategies in dialog systems. Our technique is based on the use of a statistical dialog manager that is learned using a dialog corpus for the specific task. A dialog simulation technique is used to automatically generate the data required to learn a new dialog model. We have applied our technique to explore dialog strategies for a dialog system designed to collect monitored data from patients suffering from diabetes. In addition, a set of specific measures has been defined to evaluate the new strategy once new simulated data is used to re-train the dialog manager. The results of the evaluation of a dialog manager developed for this system show how the variability of the dialog model is increased by detecting new dialog situations that are not present in an initial model and new valid answers for the situations that were already present.

2. The DI@L-log System

DI@L-log is a spoken dialog system which acts as a voice logbook to collect home monitored data from patients suffering from Type-2 diabetes [1]. The data collected by the system using VoiceXML files¹ 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 concerning abnormal developments. Figure 1 shows the architecture described for the development of the different functionalities of the DI@L-log system.

To apply our methodology to this task, the definition of the semantics was carried out considering the information that is required to monitor the patients and inform them

¹<http://www.w3.org/TR/voicexml20/>

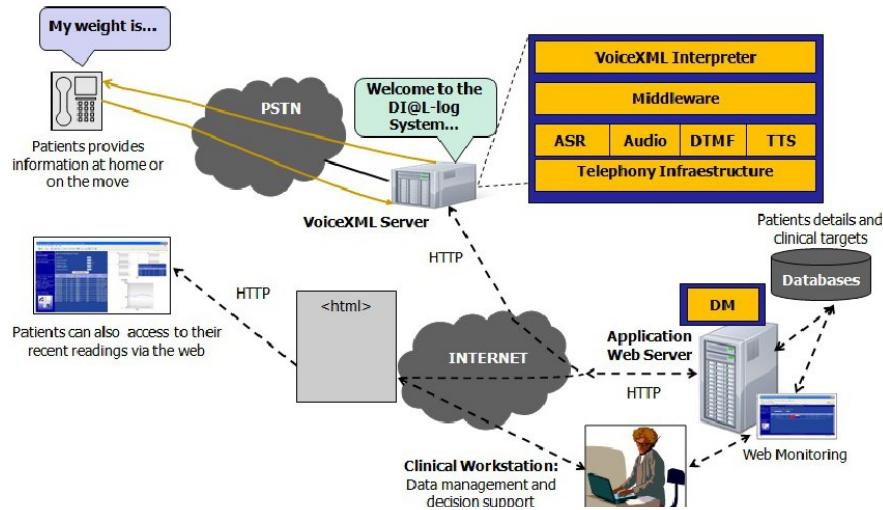


Figure 1. Architecture of the DI@L-log System

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.

Figure 2. An example of a dialog for the DI@L-logtask

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 turns. A total of 12 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*).

3. Our Statistical Dialog Management Technique

In most dialog systems, the DM takes its decisions based only on the information provided by the user in the previous turns and its own model. This is the case with most slot-filling dialog systems, like the DI@L-log system. The methodology that we propose for the selection of the next system answer in this kind of task is as follows [4].

We consider that, at time i , the objective of the DM is to find the best system answer A_i . This selection is a local process for each time i and takes into account the previous history of the dialog, that is to say, the sequence of states of the dialog (i.e. pairs *system-turn*, *user-turn*) preceding time i :

$$\hat{A}_i = \arg \max_{A_i \in \mathcal{A}} P(A_i | S_1, \dots, S_{i-1})$$

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

As the number of all possible sequences of states is very large, we define a data structure in order to establish a partition in the space of sequences of states (i.e., in the history of the dialog preceding time i). This data structure, that we call Dialog Register

(DR), contains the information provided by the user throughout the previous history of the dialog. Using this data structure, the selection of the best A_i is given by:

$$\hat{A}_i = \arg \max_{A_i \in \mathcal{A}} P(A_i | DR_{i-1}, S_{i-1})$$

The selection of the system answer is carried out through a classification process, for which a multilayer perceptron (MLP) is used. The input layer receives the codification of the pair (DR_{i-1}, S_{i-1}) . The output generated by the MLP can be seen as the probability of selecting each of the different system answers defined for a specific task. The DR defined for the DI@L-log task is the sequence of four fields related to the information that the system requires from the user (*Weight*, *Sugar*, *Systolic-Pressure*, and *Diastolic-Pressure*).

4. Our Dialog Simulation Technique

Our approach for acquiring a dialog corpus is based on the interaction of a user simulator and a DM simulator [5]. Both modules use a random selection of one of the possible answers defined for the semantics of the task (user and system dialog acts). At the beginning of the simulation, the set of system answers is defined as equiprobable. When a successful dialog is simulated, the probabilities of the answers selected by the dialog manager during that dialog are incremented before beginning a new simulation.

An error simulator module has been designed to perform error generation. The error simulator modifies the frames generated by the user simulator once it selects the information to be provided. In addition, the error simulator adds a confidence score to each concept and attribute in the frames. Experimentally, we have detected 2.3 errors per dialog in the initial corpus of 100 dialogs acquired for the task. This value can be modified to adapt the error simulator module to the operation of any ASR and NLU modules.

The model employed for introducing errors and confidence scores is inspired in the one presented in [8]. Both processes are carried out separately following the noisy communication channel metaphor 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.

On the one hand, the probability $P(\tilde{a}_u | a_u)$ is obtained by Maximum-Likelihood using the initial labeled corpus acquired with real users. To compute it, we consider 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 the word-level utterance that corresponds to 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.

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

On the other hand, the generation of confidence scores is carried out by approximating $P(c | \tilde{a}_u, a_u)$ assuming that there are two distributions for c . These two distributions are

defined manually generating confidence scores for correct and incorrect hypotheses. These definitions are based on a sampling over the distributions found in the training data corresponding to our initial corpus.

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

The DM simulator considers that the dialog is unsuccessful when one of the following conditions take place: i) The dialog exceeds the maximum number of system turns; ii) the answer selected by the DM corresponds with a query not required by the user simulator; iii) the database query module provides an error warning because the user simulator has not provided the mandatory information needed to carry out the query; iv) the answer generator provides an error warning when the selected answer involves the use of a data item not provided by the user simulator. A user request for closing the dialog is selected once the system has provided the information defined in its objective(s). The dialogs that fulfill this condition before the maximum number of turns are considered successful.

4.1. Measures defined for the Evaluation

We propose three measures to evaluate the evolution of the dialog strategy once the simulated dialogs are used to reestimate it. These measures are calculated by comparing the answer automatically generated by the DM for each input in the test partition with regard to the reference answer annotated in the evaluation corpus. This way, the evaluation is carried out turn by turn. These three measures are: i) *strategy*: the percentage of answers provided by the DM that exactly follow the initial strategy defined for the task; ii) *correct*: the percentage of answers provided by the DM that are coherent with the current state of the dialog although they do not follow the original strategy; iii) *error*: the percentage of answers provided by the DM that would cause the failure of the dialog.

The measure *strategy* is automatically calculated, evaluating whether the answer generated by the DM follows the set of rules defined for the initial strategy. On the other hand, the measures *correct* and *error* are manually evaluated by an expert in the task. The expert evaluates whether the answer provided by the DM allows the correct continuation of the dialog for the current situation or whether the answer causes the failure of the dialog. (e.g., the DM suddenly ends the interaction with the user, a query to the database is generated without the required information, etc).

5. Evaluation Results

Firstly, we evaluated the behavior of the original DM that was learned using the initial corpus of 100 dialogs acquired using the strict strategy described in section 2. A 5-fold cross-validation process was used to carry out the evaluation of this manager. The corpus was randomly split into five subsets of 253 samples (20 % of the corpus). Our experiment consisted of five trials. Each trial used a different subset taken from the five subsets as the test set, and the remaining 80 % of the corpus was used as the training set. A validation subset (20 %) was extracted from each training set. Table 5 shows the results of the evaluation.

	<i>strategy</i>	<i>correct</i>	<i>error</i>
System answer	96.11 %	97.45 %	2.55 %

Cuadro 1. Results of the evaluation of the initial DM learned for the DI@L-log task

The results of the *strategy* and *correct* measures show that the satisfactory operation of the developed DM due to the structure of the dialogs is almost the same in the complete set of dialogs of the initial corpus. The codification developed to represent the state of the dialog and the good operation of the MLP classifier make it possible for the answer generated by the manager to agree with one of the valid answers of the defined strategy (*strategy*) by a percentage of 96.11 %. Finally, the number of answers generated by the MLP that can cause the failure of the system is only a 2.55 % percentage. An answer that is coherent with the current state of the dialog is generated in 97.45 % of cases. These last two results also demonstrate the correct operation of the classification methodology.

Then, by employing the methodology proposed in this paper for dialog simulation, we acquired 100,000 dialogs which have been useful for the manager to deviate from the initial strategy. Table 2 summarizes the statistics of the acquisition of the simulated corpus for the DI@L-log task. 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 always mandatory and 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.

Simulated dialogs	100,000
Successful dialogs	27,521
Different dialogs	1,573
Number of user turns per dialog	7.02

Cuadro 2. Statistics of the corpus acquisition for the DI@L-log system

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 are shown between brackets. A sentence in natural language, that is equivalent to the labeled system turn selected by the DM, has been incorporated to clarify the explanation.

In this dialog, the system begins asking the user about his weight. As a low confidence measure is introduced for the value provided by the user simulator in U1, the system decides to confirm this value in S2. Then, the system asks for the sugar value. The user simulator provides this value in U3 and a high confidence measure is assigned. Therefore, this value does not need to be confirmed.

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

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

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

Finally, we evaluated the evolution of the DM when the successful simulated dialogs were incorporated into the training corpus. A new DM model was learned each time a new set of simulated dialogs was generated. For this evaluation, we used a test partition that was extracted from the simulated corpus (20 % of the samples). Table 5 shows the results of the evaluation of the DM model after the successful dialogs were incorporated into the training corpus.

	<i>strategy</i>	<i>correct</i>	<i>error</i>
System answer	13.64 %	98.84 %	1.16 %

Cuadro 3. Results of the evaluation of the DI@L-log DM obtained after the dialog simulation

Figure 4 shows the evolution of *strategy*. It can be observed how the original strategy was modified since the measure decreases to 13.64 %, thereby allowing the DM to tackle new situations and generate new coherent answers for the situations already present in the initial corpus. Due to the new learning process, the DM can now ask for the required information using different orders, confirm these information items taking into account the confidence scores, reduce the number of system turns for the different kinds of dialogs, automatically detect different valid paths to achieve each of the required

objectives, etc. The values obtained for the coherent and error measures also indicate the correct performance of the enhanced DM.

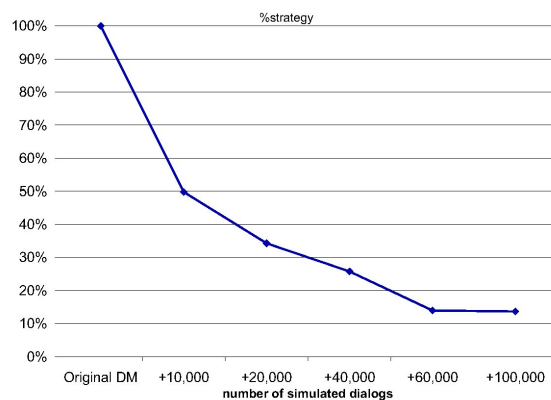


Figura 4. Evolution of the *strategy* measure with regard to the incorporation of new simulated dialogs in the DI@L-log task

6. Conclusions

In this paper, we have described a technique for exploring dialog strategies in dialog systems. Our technique is based on two main elements: a statistical dialog methodology for dialog management and an automatic dialog simulation technique to generate the data that is required to re-train the dialog model. The results of applying our technique to the DI@L-log system, which follows a very strict interaction flow, show that the proposed methodology can be used not only to develop new dialog managers but also to explore new enhanced strategies. Carrying out these tasks with a non-statistical approach would require a very high cost that sometimes is not affordable. As a future work, we are adapting the proposed dialog management for its application in more difficult domains, in which additional information sources and more complex operations are required.

Acknowledgements

Research funded by projects CICYT TIN 2008-06742-C02-02/TSI, CICYT TEC 2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485), and DPS 2008-07029-C02-02.

References

1. Black, L.A., 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 CBMS'05. pp. 506–508 (2005)

2. Cuayáhuitl, H., Renals, S., Lemon, O., Shimodaira, H.: Human-Computer Dialogue Simulation Using Hidden Markov Models. In: Proc. of ASRU'05. pp. 290–295 (2005)
3. Georgila, K., Henderson, J., Lemon, O.: User Simulation for Spoken Dialogue Systems: Learning and Evaluation. In: Proc. of Interspeech/ICSLP. pp. 1065–1068 (2006)
4. Griol, D., Hurtado, L., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialogue Systems Design and Evaluation. *Speech Communication* 50(8–9), 666–682 (2008)
5. Griol, D., Hurtado, L.F., Sanchis, E., Segarra, E.: Acquiring and Evaluating a Dialogue Corpus through a Dialogue Simulation Technique. In: Proc. of SIGdial'07. pp. 39–42 (2007)
6. Paek, T., Horvitz, E.: Conversation as action under uncertainty. In: Proc. of the 16th Conference on Uncertainty in Artificial Intelligence. pp. 455–464 (2000)
7. Pietquin, O., Dutoit, T.: A probabilistic framework for dialogue simulation and optimal strategy learning. *IEEE Transactions on Speech and Audio Processing* 14, 589–599 (2005)
8. Schatzmann, J., Thomson, B., Young, S.: Error Simulation for Training Statistical Dialogue Systems. In: Proc. of ASRU'07. pp. 526–531 (2007)
9. 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)
10. Young, S.: The Statistical Approach to the Design of Spoken Dialogue Systems. Tech. rep., CUED/F-INFENG/TR.433, Cambridge University Engineering Department (2002)
11. Young, S., Schatzmann, J., Weilhammer, K., Ye, H.: The Hidden Information State Approach to Dialogue Management. In: Proc. of ICASSP'07. pp. 149–152 (2007)