Acquiring and Evaluating a Dialog Corpus through a Dialog Simulation Technique

David Griol, Lluís F. Hurtado, Emilio Sanchis, Encarna Segarra
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València, E-46022 València, Spain
{dgriol,lhurtado,esanchis,esegarra}@dsic.upv.es

Abstract

In this paper, we present an approach for automatically acquiring a dialog corpus by means of the interaction of a dialog manager and a user simulator. A random selection of the answers has been used for the operation of both modules, defining stop conditions for automatically deciding if the dialog is successful or not. Therefore, an initial corpus is not necessary to develop these two modules. In this work, we use a statistical dialog manager to evaluate the behavior of the corpus acquired using this approach. This dialog manager has been learned from the simulated corpus and has been evaluated using a previous corpus acquired for the task with real users.

1 Introduction

Learning statistical approaches to model the different modules that compose a dialog system has reached a growing interest during the last decade (Young, 2002). Although, in the literature, there are models for dialog managers that are manually designed, over the last few years, approaches using statistical models to represent the behavior of the dialog manager have also been developed (Williams and Young, 2007), (Lemon et al., 2006), (Torres et al., 2003).

In this field, we have recently developed an approach to manage the dialog using a statistical model that is learned from a data corpus. This work has been applied within the domain of a Spanish project call DIHANA (Benedí et al., 2006). The task that we considered is the telephone access to information about train timetables and prices in Spanish. A set of 900 dialogs was acquired in the DIHANA project using the Wizard of Oz technique. A set of 300 different scenarios was used to carry out the acquisition. Two main types of scenarios were defined. Type S1 defined only one objective for the dialog. Type S2 defined two objectives for the dialog. This corpus was labeled in terms of dialog acts to train the dialog model. The results of this work can be found in (Hurtado et al., 2006).

The success of statistical approaches depends on the quality of the data used to develop the dialog model. A great effort is necessary to acquire and label a corpus with the data necessary to train a good model. One solution for this problem consists of the development of a module that simulates the user answers. A summary of user simulation techniques for reinforcement learning of the dialog strategy can be found in (Schatzmann et al., 2006).

In this paper, we present an approach to acquire a labeled dialog corpus from the interaction of a user simulator and a dialog manager. In this approach, a random selection of the system and user answers is used. The only parameters that are needed for the acquisition are the definition of the semantics of the task (that is, the set of possible user and system answers), and a set of conditions to automatically discard unsuccessful dialogs. We have acquired a corpus for the DIHANA task using this approach. This corpus has been used for training our statistical dialog manager. Then, the Wizard of Oz corpus of the DIHANA project has been used to evaluate the be-

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 39–42,
Antwerp, September 2007. ©2007 Association for Computational Linguistics

havior of this dialog manager with real users.

2 Our approach for automatically acquiring a dialog corpus

As stated in the introduction, our approach for acquiring a dialog corpus is based on the interaction of a user simulator and a dialog manager. Both modules use a random selection of one of the possible answers defined for the semantic of the task (user and system dialog acts).

The user simulation 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. The semantics selected for the dialog manager is represented through the 51 possible system answers defined for the task. The selection of the possible user answers is carried out using the semantics defined for the user in the NLU module.

An error simulator module has been designed to perform error generation and the addition of confidence measures in accordance with an analysis of the DIHANA corpus. This information modifies the frames generated by the user simulator and also incorporates confidence measures for the different concepts and attributes. Experimentally, we have detected 2.7 errors per dialog. This value can be modified to adapt the error simulator module to the operation of any ASR and NLU modules.

2.1 The corpus acquisition

A maximum number of system turns per dialog was defined for acquiring a corpus. The dialog manager considers that the dialog is unsuccessful and decides to abort it when the following conditions take place:

- The dialog exceeds the maximum number of system turns.
- The answer selected by the dialog manager corresponds with a query not required by the user simulator.
- The database query module provides an error warning because the user simulator has not provided the mandatory information needed to carry out the query.
- The answer generator provides a error warning when the selected answer involves the use of a data not contained in the DR, that is, not provided by the user simulator.

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

A total of two million simulated dialogs were necessary in order to acquire a dialog corpus following the same characteristics of the DIHANA corpus and using the same set of scenarios developed for its acquisition. The computer used for the acquisition was able to simulate 1,000 dialogs every 40 minutes.

Table 1 summarizes the statistics of the acquisition for the two types of scenarios that were defined.

	Type S1	**Type S2**
Simulated dialogs	10^6	10^6
Successful dialogs	15,383	1,010
Different dialogs	14,921	998
Number of user turns per dialog	4.9	6.2

Table 1: Statistics of the new corpus acquisition

Table 2 shows an example of a dialog corresponding to a Type S1 scenario (to know timetables from Valencia to Barcelona for May the 30th).

S1: (Opening:Nil:Nil)
Welcome to the railway information system. How can I help you?
U1: Departure-Hour [0.9]
Destination: Barcelona [0.9]
Departure-Date: 05-31-2007 [0.1]
Arrival-Hour: 08.00-23.00 [0.9]
S2: (Confirmation:Date:Date)
Do you want to leave on May the 31th?
U2: Negation[0.9]
Departure-Date: 05-30-2007 [0.9]
S3: (Answer:Departure-Hour:Departure-Hour)
(New-Query:Nil:Nil)
{*Timetables information*} Anything else?
U3: Negation
S4: (Closing:Nil:Nil)
Thank you for using the system.

Table 2: An example of a Type S1 dialog extracted from the acquired corpus

The dialog begins with a greeting turn generated by the system (S1). In the first turn, the simulated user provides the *Destination* and the *Departure-Date*. In addition, it facilitates the *Arrival-Hour* (set as optional data for the scenario). The error simulator introduces in this first turn an error value in the *Departure-Date* slot (it changes day 30 by 31) and assigns confidence scores to the different slots. In this case, a low confidence is assigned to this erroneous value.

In the second system turn, a confirmation for the *Departure-Date* is selected. Considering the information defined in the objective of the scenario, the user simulator selects a *Negation* dialog act and provides the correct value for the *Departure-Date* according to the objective (U2). In this turn, the error simulator assigns a high confidence value to the information provided by the user. In the following system turn (S3), the dialog manager selects to make a query about timetables to the database. As the necessary information is available, the database query module carries out the query and the dialog manager provides the information defined as objective for the dialog. Having this information, the user simulator selects a request for closing the dialog in the following turn (U3).

3 Dialog management in the DIHANA project

We have developed a Dialog Manager (DM) based on the statistical modelization of the sequences of dialog acts (user and system dialog acts). A detailed explanation of the dialog model can be found in (Hurtado et al., 2006). We represent a dialog as a sequence of pairs (*system-turn, user-turn*):

$$(A_1, U_1), \cdots, (A_i, U_i), \cdots, (A_n, U_n)$$

where A_1 is the greeting turn of 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.

The objective of the dialog manager at time i is to generate the best system answer. This selection, that is a local process, takes into account the previous history of the dialog, that is to say, the sequence of states of the dialog preceding time i:

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

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

As the number of all possible sequences of states is very large, we defined a data structure in order to establish a partition in the space of sequences of states. This data structure, that we call Dialog Register (DR), contains the concepts and attributes provided by the user throughout the previous history of the dialog. Using the DR, the selection of the best system answer is made using this maximization:

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

The last state (S_{i-1}) is considered for the selection of the system answer due to the fact that a user turn can provide kinds of information that are not contained in the DR, but are important to decide the next system answer. This is the case of the task-independent information.

The selection of the system answer is carried out by means of a classification process, in which a multilayer perceptron (MLP) is used. The input layer holds the codification of the pair (DR_{i-1}, S_{i-1}) and the output of the MLP can be seen as the probability of selecting each one of the 51 different system answers defined for the DIHANA task. For the DIHANA task, the DR is a sequence of 15 fields, where each concept or attribute has a field associated to it.

4 Evaluation

A statistical dialog manager was learned using the corpus acquired with the dialog simulator technique (M1 manager). The DIHANA corpus was used as test set to evaluate the behavior of this dialog manager with a real users corpus.

We also learned another dialog manager using the DIHANA corpus as training set (M2 manager). A 5-fold cross-validation process was used to carry out the evaluation of this manager. Therefore, all the DIHANA corpus is used for testing both M1 and M2 dialog managers.

We defined three measures to evaluate the performance of both dialog managers:

1. The percentage of answers that follows the strategy defined for the acquisition of the DIHANA corpus ($\%strategy$).

2. The percentage of answers that are coherent with the current state of the dialog, but that not necessary follow this strategy ($\%correct$).

3. The percentage of answers that are considered erroneous according to the current state of the dialog ($\%error$).

Table 3 shows the results obtained for the different measures after the evaluation.

	M1 manager	M2 manager
$\%strategy$	54.57%	97.34%
$\%correct$	88.83%	99.33%
$\%error$	11.17%	0.67%

Table 3: DM evaluation results

It can be observed that the M1 manager provides a 88.83% of answers that are coherent with the current state of the dialog. Using the DIHANA corpus in order to learn the dialog model (M2 manager), the 97.34% of the answers provided by this dialog manager follows the strategy defined for the WOz. With regard to the M1 manager, only the 54.57% follows this strategy. Therefore, we can see that the M1 dialog manager separates from the strategy defined for the WOz as expected. Regarding to the $\%error$ measure, the M1 dialog manager provides a 11.17% percentage of answers that are not compatible with the state of the dialog.

5 Conclusions

In this paper, we have presented an approach to automatically acquire a dialog corpus by means of the interaction of a user simulator and a dialog manager. For the development of both modules, we defined the semantics of the possible answers for the system and the user in a specific task. A random selection of these answers and a set of stop conditions were used in order to acquire a dialog corpus, deciding automatically if the dialog has to be considered successful.

The corpus that has been obtained by means of this approach has been used to learn a dialog manager, using a statistical dialog model. We have used a previous corpus acquired with real users to evaluate this dialog manager. The results of the evaluation show that the learned dialog model could be used as an initial dialog manager, generated without many effort and with very high performance. This initial dialog manager could be improved with a posteriori interaction with real users.

As future work, we want to use this approach to acquire a dialog corpus within the framework of a new project called EDECAN. The main objective of the ongoing EDECAN project is to develop a dialog system for booking sports facilities in our university. Using this approach, we want to acquire a corpus that makes possible the learning of a dialog manager for the domain of the EDECAN project. This dialog manager will be used in a supervised acquisition of a dialog corpus with real users.

6 Acknowledgements

Work partially supported by the Spanish MEC and FEDER under contract TIN2005-08660-C04-02.

References

J.M. Benedí, E. Lleida, A. Varona, M.J. Castro, I. Galiano, R. Justo, I. López, and A. Miguel. 2006. Design and acquisition of a telephone spontaneous speech dialogue corpus in Spanish: DIHANA. In *Proc. of LREC'06*, Genove, Italy.

L.F. Hurtado, D. Griol, E. Segarra, and E. Sanchis. 2006. A Stochastic Approach for Dialog Management based on Neural Networks. In *Procs. of InterSpeech'06*, Pittsburgh, USA.

O. Lemon, K. Georgila, and J. Henderson. 2006. Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: the talk towninfo evaluation. In *Proc. of IEEE-ACL Workshop on Spoken Language Technology (SLT 2006)*, Aruba.

J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. In *Knowledge Engineering Review*, volume 21(2), pages 97–126.

F. Torres, E. Sanchis, and E. Segarra. 2003. Development of a stochastic dialog manager driven by semantics. In *Proc. of EuroSpeech'03*, pages 605–608.

J. Williams and S. Young. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. In *Computer Speech and Language 21(2)*, pages 393–422.

S. Young. 2002. The Statistical Approach to the Design of Spoken Dialogue Systems. Technical report, Cambridge University Engineering Department.