Acquisition of synthetic dialog corpora using a task-independent dialog system

Francisco Torres Goterris

Departamento de Sistemas Informáticos y Computación,
Universitat Politècnica de València / Universidad Politécnica de Valencia, Spain
ftgoterr@dsic.upv.es

Abstract

In this paper, we present the acquisition of synthetic dialog corpora through a dialog system that integrates a stochastic dialog manager and a rule-oriented user simulator. These modules are task-independent, and can be adapted to different semantic-restricted domains. Our stochastic dialog manager can interact with real or simulated users, storing automatically the acquired dialogs. In addition, the simulation mode allows us to acquire series of dialogs, verifying automatically their successful endings. These dialogs are used to adapt the stochastic dialog models and, therefore, to enhance the system in new acquisitions. This methodology has been applied to develop two dialog systems in different domains: a train services information system, and a sport booking system.

Index Terms: stochastic dialog management, user simulation, task independence, synthetic acquisition

1. Introduction

In the development of spoken dialog systems, statistical techniques have provided good results, as described in [1], [2], and [3]. However, there are some drawbacks using these techniques, as the high cost of the acquisition of the corpora and the evaluation interacting with real users. In order to overcome them, the user simulation techniques have been considered, as described in [4], [5], and [6]. In addition, significant work has been made to design dialog systems that can be easily adapted to different domains, i.e., to design task-independent dialog systems, as described in [7], and [8].

The adaptation to new tasks is one of the aims in the EDECAN project [9]. In this research frame, a dialog system has been developed to attend different semantic-restricted tasks: the BASURDE and DIHANA tasks [10], which access a train information system; and the EDECAN-SPORT task, which provides access to a sport courts booking system. In this paper, we call them BASURDE and EDECAN tasks.

In this platform, the dialog manager [11] is based on a stochastic dialog model, which is a bigram model (BM) of dialog acts, and includes a historic register (HR), which stores all the data provided in previous turns. This dialog manager follows a hybrid strategy, half stochastic (due to the use of BM), and half heuristic (due to the query of HR). It must be remarked that it can attend both tasks, just reading their corresponding bigram models, and other configuration files.

In addition, a user simulator [12] has been developed. This module allows us to acquire synthetic dialogs, learn dialog models, and evaluate the dialog system. The user simulator selects states of the same BM, and applies some heuristic rules that implement a collaborative dialog strategy. These rules are task-independent, and they serve to generate consistent dialogs, which are useful for learning dialog models, in both tasks.

During a synthetic acquisition, on the one hand, the dialog manager automatically verifies the success of the dialogs and can modify the BM, readjusting the probabilities of the transitions. On the other hand, the user simulator just provides an appropriate flow of user turns to easily generate consistent dialogs. The validity of this simulation technique has been demonstrated by testing and enhancing the dialog manager.

2. Tasks and models description

The definition of the semantics of the tasks is based on the concepts of dialog act and frame. The dialog act is the semantic unit for describing the dialog turns, and dialog acts (DAs) are used in the definition of the dialog models. The frame is the unit that structures the concepts and attributes supplied in dialog turns. Thus, our two tasks have been semantically characterized by identifying the concepts and attributes involved in dialogs with real users. Each user turn consists of utterances in which some intentions are transmitted (i.e., involved concepts) and some items of information are supplied (i.e., attributes and their values).

In the case of the BASURDE task, there was a labeled corpus from which the BM of DAs was extracted. However, at the moment of designing our system, there was not a labeled corpus of the EDECAN task. Therefore, we had to design by hand an initial BM, defining the labels of the states and setting all the transitions with the same probability. Then, applying the simulation technique, this BM was modified appropriately.

The BASURDE dialog corpus was labeled applying the concept of dialog act and a hierarchy of three levels. In this hierarchy, the first level (L1) identifies the generic dialog act; the second level (L2), the semantic of the task; and the third level (L3), the instantiated attributes. Once the dialog turns are labeled, each dialog consists of a sequence of DAs. Thus, the dialog models are structured by sequences of DAs. In the EDECAN task, although the acquired corpus had not been labeled, we have used the same methodology for defining their DAs, which could be used for its labeling.

Table 1 shows the concepts and attributes defined for coding the user turns, and Table 2 shows the labels defined for coding the DAs. There are task-independent labels and concepts, which are common in both tasks, and there are other labels, concepts and attributes that are specific to each task.

Task-independent concepts	
ACCEPTANCE, REJECTION, NOT-UNDERSTOOD	
Task-dependent concepts	**BASURDE task**
DEPARTURE-HOUR, ARRIVAL-HOUR, PRICE, TRAIN-TYPE, SERVICES	
Task-dependent concepts	**EDECAN task**
AVAILABILITY, BOOKING, BOOKED, CANCELLATION	
Attributes	**BASURDE task**
ORIGIN, DESTINATION, DEPARTURE-DATE, ARRIVAL-DATE, DEPARTURE-HOUR, ARRIVAL-HOUR, PRICE, TRAIN-TYPE, SERVICES, NUMBER-OF-TRAINS, ORDER-NUMBER	
Attributes	**EDECAN task**
SPORT, DATE, HOUR, COURT-TYPE, NUMBER-OF-COURTS, COURT-ID	

Table 1. *Concepts and attributes*

First level labeling	
OPENING, CLOSING, WAITING, NEW-QUERY, QUESTION, CONFIRMATION, ANSWER, CHOICE, ACCEPTANCE, REJECTION, NOT-UNDERSTOOD, UNDEFINED	
Second and third levels labeling	**BASURDE task**
ORIGIN, DESTINATION, DEPARTURE-DATE, ARRIVAL-DATE, DEPARTURE-HOUR, ARRIVAL-HOUR, PRICE, TRAIN-TYPE, SERVICES, NUMBER-OF-TRAINS, ORDER-NUMBER, NIL	
Second and third levels labeling	**EDECAN task**
AVAILABILITY, BOOKING, BOOKED, CANCELLATION, SPORT, DATE, HOUR, COURT-TYPE, NUMBER-OF-COURTS, COURT-ID, NIL	

Table 2. *Labels of the dialog acts*

Task-dependent concepts are the goals of the user queries. In the BASURDE task, they are DEPARTURE-HOUR, and ARRIVAL-HOUR (involved in queries about timetables), PRICE (queries about prices), TRAIN-TYPE, and SERVICES (queries about services). In the EDECAN task, they are AVAILABILITY (queries about availability of courts), BOOKING (bookings of courts), BOOKED (queries about booked courts), and CANCELLATION (cancellations of the bookings). In both tasks, the attributes are the items that the users must or can provide to specify their goals, and the system must or can supply in order to answer the queries. The attributes are specific of each task and their names are self-explanatory of their meaning.

In order to acquire dialogs, it was also necessary to define a set of task scenarios. We have defined 15 scenarios for each task, with different levels of complexity. The first and the last of them have been coded as it is shown in Table 3.

Scenario-0	**BASURDE task**
<DEPARTURE-HOUR> <PRICE> ORIGIN DESTINATION DEPARTURE-DATE [TRAIN-TYPE] [DEPARTURE-HOUR]	
Scenario-14	**BASURDE task**
<DEPARTURE-HOUR> <ARRIVAL-HOUR> <PRICE> <TRAIN-TYPE> ORIGIN DESTINATION DEPARTURE-DATE [DEPARTURE-HOUR] [ARRIVAL-HOUR]	
Scenario-0	**EDECAN task**
<AVAILABILITY> SPORT [COURT-TYPE] [DATE] [HOUR]	
Scenario-14	**EDECAN task**
<BOOKED> <CANCELLATION> [SPORT] [DATE] [HOUR] [<AVAILABILITY>] <BOOKING> SPORT [COURT-TYPE] DATE HOUR	

Table 3. *Codification of some scenarios*

In the case of the BASURDE task, Scenario-0 consists of a query about departure timetables and prices on a journey, in which the user must specify origin, destination, and departure date, and s/he can provide the train-type or the departure time slot. Scenario-14 is a complex query, with four user goals (arrival and departure timetables, prices and train-types), three mandatory attributes, and two optional attributes.

In the case of the EDECAN task, Scenario-0 consists of a query about availability on a certain sport, allowing the user to specify the date, the hour, and the court-type. Scenario-14 can be decomposed into three phases: (1) the user has to obtain his/her booked courts; (2) s/he has to cancel some court of the previous list, and s/he can specify the court providing the sport, the date, or the hour; and (3) s/he has to book some court providing the sport, the date, and the hour, and s/he can supply the court-type, or can make an availability query.

Thus, dialogs of complex scenarios are composed by sequences of sub-dialogs, which can share data among them. In the EDECAN task, it occurs between the BOOKED and CANCELLATION sub-dialogs, and between the AVAILABILITY and BOOKING sub-dialogs. In the BASURDE task, it occurs when there are several user goals related to the same journey.

Regarding the dialog models, we have to consider the DAs labeling of each task. Examples of labeling of segments of dialogs, one for each task, are shown in Figures 1 and 2. As it can be observed, each turn is labeled with one or more DAs.

U0: *I may want to know the train timetables from Zaragoza to Ciudad-Real on next Friday evening, or on Saturday morning.*
(U:QUESTION:DEPARTURE-HOUR:DESTINATION,DEPARTURE-DATE, DEPARTURE-HOUR,ORIGIN)
S1: *I'm sorry, but I haven't understood you. Do you want to travel from Zaragoza to Ciudad-Real on Friday the 29th of October?*
(S:NOT-UNDERSTOOD:NIL:NIL) (S:CONFIRMATION:DESTINATION, DEPARTURE-DATE,ORIGIN:DESTINATION,DEPARTURE-DATE,ORIGIN)

Figure 1. *Segment of a BASURDE dialog*

U0: *Could I book a tennis-court on next Friday?*
(U:QUESTION:BOOKING:DATE,SPORT)
S1: *Do you want to play on Tuesday?*
(S:CONFIRMATION:DATE:DATE)
U1: *No. I want to play on Friday.*
(U:REJECTION:DATE:NIL) (U:ANSWER:DATE:DATE)

Figure 2. *Segment of an EDECAN dialog*

In Figure 1, the descriptor of the U0 turn identifies a state in which the user asks for (L1: QUESTION) timetables (L2: DEPARTURE-HOUR), providing the values of four attributes (those included in L3). In the S1 turn, the system replies with two DAs: a misunderstanding (L1: NOT-UNDERSTOOD), and a confirmation of attributes (L1: CONFIRMATION), giving their values (L2 and L3: DESTINATION, DEPARTURE-DATE, ORIGIN).

In Figure 2, the user asks for (L1: QUESTION) bookings (L2: BOOKING), giving the values of two attributes (L3: DATE, SPORT). In the S1 turn, the system needs to confirm (L1: CONFIRMATION) the date, providing its value (L2 and L3: DATE). Then, the user carries out two DAs in the same turn: s/he rejects the date (L1: REJECTION), and provides other value of this attribute (L1: ANSWER).

Using these label sets, we have defined the descriptors of the dialog states. However, in the case of the BASURDE task, the reduced size of the corpus (215 dialogs) and the great number of different DAs leads to a poor estimation of these states. We found a solution by dismissing the L3 labels in defining the states. In such a case, the number of states is reduced (155 identifiers), and each one is better estimated.

The dialog states of the BASURDE model are defined by one or more descriptors that match the (US-ID:L1-ID:L2-ID) pattern, where US-ID specifies whether a user or system turn, L1-ID is one of the L1 labels, and L2-ID are one or more of the L2 labels. For instance, the U0 turn of Figure 1 is assigned to the (U:QUESTION:DEPARTURE-HOUR) dialog state. Therefore, the same state describes all the turns characterized by certain L1 and L2 labels, without considering the instantiated attributes.

In the EDECAN task, we have defined two dialog models: L2-BM and L3-BM, excluding or including the L3 labeling in the states description. For instance, the (U:QUESTION:BOOKING) descriptor identifies a L2-BM state in which the user asks for booking a court, no mattering the provided attributes; and the (U:QUESTION:BOOKING:DATE) descriptor identifies a L3-BM state of booking question specifying the date. The L2-BM has 66 states, and the L3-BM has 494 states. In both models, initially, all the transitions had the same probability.

3. Task-independent dialog platform

Our platform integrates the user and system dialog managers, the user and system language generators, and the database manager. It can also integrate understanding modules.

In a BASURDE synthetic acquisition, the understanding module receives the sentences generated by the user simulator (user dialog manager, UDM), and extracts its meaning, providing a set of user frames. Currently, none understanding module has been designed for the EDECAN task. Thus, in dialogs of this task, the UDM frames are supplied to the system dialog manager (SDM). In both tasks, there is the possibility of introducing error simulation in the user frames.

The database manager attends the queries of the SDM. The user/system language generators translate the user/system frames into sentences in natural language (currently, in Spanish and English). Both language generators work using a set of templates and rules for instantiating the templates.

In each dialog turn, the SDM reads the user frames, decides the system dialog strategy, and builds the system frames. The SDM determines its action selecting a new state in its BM, by taking into account the last user turn, the probabilities of the transitions in the BM, and the consistence of these transitions given the content of its system HR. The UDM reads the system frames, decides its action (according to its BM, its user HR, and the rules that establish a collaborative strategy for satisfying the scenarios), and builds the user frames. The SDM and UDM algorithms are task-independent. All the task information has been encapsulated into the models, the scenarios, and other configuration files. Thus, the data-structures are initialized reading these files, and the methods have been appropriately parameterized.

We have developed a JAVA dialog platform [13], according to this design of the system. By means of this platform, we can acquire dialogs for both tasks, selecting real or simulated users. In the interactive mode, any human user can give the user frames through a graphical interface, and s/he can read the system answers, carrying out whole dialogs. In the simulation mode, the dialog is completely done by the platform. This application allows us to simulate dialogs turn by turn, or whole dialogs, or series of any number of dialogs, and to specify which scenarios are simulated. In addition, the user frames can be modified by including errors in the attributes whose values are critical to the success of the dialog. Moreover, there are the training and test modes, which are used for learning and evaluating the BM.

Using this platform, we have carried out several training sets for the EDECAN task, starting from the BM described in Section 2. Different trainings have been made by enabling or disabling the error simulation (each training set contains 4,000 dialogs for each scenario, i.e., a total of 60,000 dialogs). Several test sets have been made to evaluate the learnt models (15,000 dialogs per test set). In addition, several test sets for the BASURDE task have been carried out. The platform successfully works with both tasks (achieving success rates of 0.90, in the simulation mode). Although the success rates would be lower interacting with real users, the performance seems acceptable to use it for a real acquisition.

It must be remarked that the SDM applies a hybrid dialog strategy. However, the EDECAN training starts from an initial BM, applying a heuristic strategy. To measure the quality of the learnt model, the initial BM and the learnt BM have been tested disabling the heuristic rules. In such a situation, the initial BM does not work (its success rate is 0.05), whereas the success rate is 0.43 using the learnt BM. This result is coherent with a similar experiment done for BASURDE, and confirms the utility of this technique for training the models.

4. Dialog corpora description

Our platform allows us to quickly acquire a great number of synthetic dialogs. A sample of these acquisitions can be found in [13]. Figure 3 shows a simplified segment of an acquired dialog (because of the platform stores more information of each turn). Three turns are shown: a user turn, asking for courts availability; a system turn, confirming the sport and the date; and a new user turn, in which these items are accepted. After reading the availability frame, the system transits in both models, updates its SHR, and builds a confirmation frame, according to the (S:CONFIRMATION:SPORT:DATE,SPORT) state. The SLG translates this frame into a sentence. Once the system frame is read by the UDM, this module selects transitions in its BM, and generates an acceptance frame, given that the data to confirm are right.

U0: (AVAILABILITY) 1.00
SPORT: tennis 0.32
DATE: 12-08-2010 0.77

*** **System L2-BM** transits to (U:Question:Availability)
*** **System L3-BM** transits to (U:Question:Availability:Date,Sport)
*** **System HR**:
[AVAILABILITY] confidence = 1.00 value = ???
[SPORT] confidence = 0.32 value = tennis
[DATE] confidence = 0.77 value = 12-08-2010
*** **System L2-BM** transits to (S:Confirmation:Sport)
*** **System L3-BM** transits to (S:Confirmation:Sport:Date,Sport)
S1: (CONFIRMATION) 1.00
SPORT: tennis 0.45
CURRENT-DATE: 12-08-2010 0.77
S1: On August the twelfth, do you want to play tennis?

*** **User L2-BM** transits to (S:Confirmation:Sport)
*** **User L2-BM** transits to (U:Acceptance:Sport)
U1: (ACCEPTANCE) 1.00

Figure 3. *Segment of an EDECAN synthetic dialog*

Statistics about the synthetic acquisitions, one for each task, are shown in Tables 4 and 5. The first columns on the right side (labeled *All*) show the results for all the dialogs (10,500 dialogs of each task). The other columns (labeled by number of scenario) show the results for each scenario.

In the BASURDE acquisition, an average success rate of 0.88 has been achieved introducing 1.02 errors per dialog. The average length of the dialogs is 6.07 turns. In the BASURDE corpus acquired with real users, the average length was 6.79 turns. Therefore, our platform develops dialogs of similar length. The system does an average of 2.25 answers, 2.94 confirmations, 0.46 questions, and 0.37 both question and confirmation turns, per dialog. The user simulator does an average of 2.35 questions, 0.59 answers (in which providing items of information), 1.28 acceptances, 0.68 both acceptance and answer turns (in which confirming some items, and providing other items), and 1.15 both rejection and answer turns (in which rejecting some items, and providing the right values), per dialog. There are clear correlations between the user questions and the system answers, and also between the user acceptances & rejections and the system confirmations. The transitions to other states (like the not-understood turns) are scarce (0.02 turns per dialog) due to the collaborative strategy of the user simulator. In an acquisition with real users, these states are more frequent, and the mentioned correlations between system and user turns are not as strong as here.

In the EDECAN acquisition, an average success rate of 0.92 has been achieved introducing 1.50 errors per dialog. The average length of the dialogs is 7.12 turns. The system does an average of 1.72 answers (providing courts according to the specified goals), 1.18 choices (confirming the booking or the cancellation of selected courts), 1.54 confirmations, and 2.65 questions, per dialog. The user simulator does an average of 1.72 questions, 3.84 answers (providing items of information, or selecting courts from supplied lists), 1.01 acceptances, and 0.52 both rejection and answer turns, per dialog. Again, the transitions to other states are infrequent. In the detailed information by scenarios, it can be observed more differences in the EDECAN dialogs than in the BASURDE dialogs because of the former includes more complex scenarios.

BASURDE Scenario	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**All**
Success Rate	**0.92**	**0.88**	**0.91**	**0.90**	**0.90**	**0.88**	**0.81**	**0.89**	**0.87**	**0.93**	**0.82**	**0.88**	**0.86**	**0.89**	**0.84**	**0.88**
Number of Turns	**5.80**	**5.75**	**5.83**	**6.98**	**6.86**	**6.91**	**5.14**	**6.98**	**5.70**	**4.16**	**5.09**	**6.30**	**5.69**	**6.95**	**6.91**	**6.07**
System Turns																
Answer	1.92	1.84	1.90	2.89	2.43	2.88	1.82	2.89	1.80	1.00	1.83	2.87	1.81	2.89	3.05	**2.25**
Confirmation	3.01	2.83	3.05	3.18	3.78	3.06	2.48	3.14	2.83	2.53	2.37	2.77	2.77	3.14	3.18	**2.94**
Question	0.51	0.62	0.55	0.51	0.20	0.55	0.42	0.54	0.62	0.35	0.44	0.32	0.66	0.54	0.11	**0.46**
Quest. & Confirm.	0.33	0.43	0.31	0.38	0.29	0.39	0.40	0.38	0.43	0.27	0.42	0.32	0.42	0.36	0.49	**0.37**
Others	0.00	0.01	0.00	0.00	0.14	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.06	**0.02**
User Turns																
Answer	0.64	0.74	0.64	0.64	0.29	0.67	0.56	0.67	0.74	0.41	0.54	0.39	0.78	0.66	0.41	**0.59**
Acceptance	0.95	1.17	1.01	1.35	2.25	1.26	1.27	1.33	1.17	0.87	1.30	1.38	1.17	1.35	1.40	**1.28**
Accept. & Answer	0.69	0.65	0.65	0.64	0.84	0.65	0.71	0.65	0.64	0.70	0.70	0.75	0.61	0.62	0.65	**0.68**
Reject. & Answer	1.39	1.33	1.43	1.43	0.47	1.41	0.77	1.41	1.32	0.60	0.70	0.87	1.28	1.40	1.40	**1.15**
Question	2.11	1.84	2.08	2.90	2.86	2.89	1.82	2.91	1.80	1.56	1.83	2.88	1.81	2.90	3.03	**2.35**
Others	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	**0.01**

Table 4. *Statistics of a synthetic acquisition in the BASURDE task*

EDECAN Scenario	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**All**
Success Rate	**0.90**	**0.91**	**0.87**	**0.91**	**0.91**	**0.93**	**0.91**	**0.96**	**0.95**	**0.90**	**0.91**	**0.93**	**0.96**	**0.89**	**0.89**	**0.92**
Number of Turns	**3.31**	**4.51**	**10.3**	**3.27**	**5.87**	**10.4**	**4.29**	**4.54**	**8.20**	**7.14**	**9.10**	**12.5**	**7.02**	**8.07**	**7.94**	**7.12**
System Turns																
Answer	1.00	1.00	1.98	1.00	1.00	1.98	1.00	2.00	2.00	1.99	1.99	2.99	2.00	1.99	1.98	**1.72**
Confirmation	1.06	1.18	1.94	1.05	0.96	1.56	1.10	1.21	1.65	1.94	1.67	2.04	1.87	1.91	1.92	**1.54**
Question	1.25	1.33	4.47	1.21	2.90	4.93	1.19	1.32	3.54	2.20	3.49	5.51	2.14	2.17	2.03	**2.65**
Choice	0.00	0.99	1.96	0.00	0.99	1.96	0.99	0.00	0.98	0.99	1.96	1.98	0.99	1.97	1.96	**1.18**
Others	0.00	0.00	0.02	0.00	0.01	0.01	0.01	0.00	0.01	0.01	0.02	0.01	0.00	0.02	0.02	**0.01**
User Turns																
Answer	1.25	2.33	6.45	1.21	3.91	6.90	2.18	1.32	4.53	3.20	5.47	7.50	3.14	4.15	4.01	**3.84**
Acceptance	0.58	0.69	1.35	0.65	0.65	1.16	0.65	0.67	1.13	1.27	1.15	1.50	1.21	1.26	1.32	**1.01**
Reject. & Answer	0.48	0.49	0.59	0.40	0.30	0.40	0.44	0.54	0.51	0.67	0.52	0.54	0.65	0.65	0.60	**0.52**
Question	1.00	1.00	1.98	1.00	1.00	1.98	1.00	2.00	2.00	1.99	1.99	2.99	2.00	1.99	1.98	**1.72**
Others	0.00	0.00	0.01	0.00	0.00	0.02	0.01	0.00	0.01	0.00	0.02	0.00	0.00	0.01	0.02	**0.01**

Table 5. *Statistics of a synthetic acquisition in the EDECAN task*

5. Conclusions

In this paper, the acquisition of synthetic dialog corpora by means of a task-independent dialog platform has been discussed. This dialog platform allows us to carry out real and simulated dialogs, to acquire synthetic corpora, to learn dialog models, and to evaluate the system using these models.

We have integrated different techniques: stochastic dialog management, rule-oriented user simulation, template-based language generation, and task parameterization. Thus, we have faced the known problems in the corpora acquisition and the systems evaluation. Following the proposed methodology, we have developed a dialog platform that facilitates the evaluation for different tasks, whether initial dialog corpora exist or not.

The results are enough satisfactory as to consider using the platform in an acquisition with real users. Future work will be oriented to acquire real user dialog corpora for the considered tasks, and to extend its use to other semantic domains.

6. Acknowledgements

This work has been partially funded by CICYT under project TIN2008-06856-C05-02/TIN, Spain.

7. References

[1] Levin, E., Pieraccini, R., Eckert, W., "A stochastic model of human-machine interaction for learning dialog strategies", IEEE Trans. on Speech and Audio Processing, 8(1):11–23, 2000.

[2] Young, S., "The statistical approach to the design of spoken dialogue systems", Cambridge University, Tech. Report, 2002.

[3] Griol, D., et al., "A statistical approach to spoken dialog systems design and evaluation", Speech Communication, 50, 666–682, 2008.

[4] Eckert, W., Levin, E., Pieraccini, R., "User modeling for spoken dialogue system evaluation", Proc. of ASRU – IEEE Workshop, Santa Barbara, USA, 1997.

[5] López-Cózar, R., et al., "Assessment of dialogue systems by means of a new simulation technique", Speech Communication 40, 387–407, 2003.

[6] Jung, S., et al., "Data-driven user simulation for automated evaluation of spoken dialogue systems", Computer Speech & Language 23, 479–509, 2009.

[7] Lemon, O., et al., "Multi-tasking and collaborative activities in dialogue systems", Proc. of SIGdial Workshop, Philadelphia, USA, 113–124, 2002.

[8] Bohus, D., Rudnicky, A.I., "The RavenClaw dialog management framework: Architecture and systems", Computer Speech & Language, 23, 332–361, 2008.

[9] Lleida, E., et al., "EDECÁN: sistema de diálogo multidominio con adaptación a contexto acústico y de aplicación", Jornadas en Tecnología del Habla (JTH), Zaragoza, Spain, 291–296, 2006.

[10] Benedí, J.M., et al., "Design and acquisition of a telephone spontaneous speech dialogue corpus in Spanish: DIHANA", Proc. of 5th LREC, Genove, Italy, 1636–1639, 2006.

[11] Torres, F., et al., "Error handling in a stochastic dialog system through confidence measures", Speech Communication 45, 211–229, 2005.

[12] Torres, F., Sanchis, E., Segarra, E., "User simulation in a stochastic dialog system", Computer Speech & Language, 22, 230–255, 2008.

[13] Torres, F., "Prototype of the dialog system", available in website http://www.laesteladetanit.es/prototype.htm, 2010.