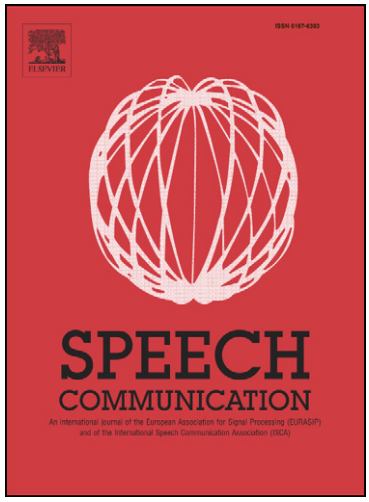
A Statistical Approach to Spoken Dialog Systems Design and Evaluation

David Griol, Lluís F. Hurtado, Encarna Segarra, Emilio Sanchis



A Statistical Approach to Spoken Dialog Systems Design and Evaluation

David Griol, Lluís F. Hurtado, Encarna Segarra, Emilio Sanchis

PII: S0167-6393(08)00045-9
DOI: 10.1016/j.specom.2008.04.001
Reference: SPECOM 1704

To appear in: *Speech Communication*

Received Date: 29 August 2007
Revised Date: 21 March 2008
Accepted Date: 5 April 2008

Please cite this article as: Griol, D., Hurtado, L.F., Segarra, E., Sanchis, E., A Statistical Approach to Spoken Dialog Systems Design and Evaluation, *Speech Communication* (2008), doi: 10.1016/j.specom.2008.04.001

A Statistical Approach to Spoken Dialog Systems Design and Evaluation

David Griol, Lluís F. Hurtado, Encarna Segarra, Emilio Sanchis

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València, E-46022 València, Spain

Abstract

In this paper, we present a statistical approach for the development of a dialog manager and for learning optimal dialog strategies. This methodology is based on a classification procedure that considers all of the previous history of the dialog to select the next system answer. To evaluate the performance of the dialog system, the statistical approach for dialog management has been extended to model the user behavior. The statistical user simulator has been used for the evaluation and improvement of the dialog strategy. Both the user model and the system model are automatically learned from a training corpus that is labeled in terms of dialog acts. New measures have been defined to evaluate the performance of the dialog system. Using these measures, we evaluate both the quality of the simulated dialogs and the improvement of the new dialog strategy that is obtained with the interaction of the two modules. This methodology has been applied to develop a dialog manager within the framework of the DIHANA project, whose goal is the design and development of a dialog system to access a railway information system using spontaneous speech in Spanish. We propose the use of corpus-based methodologies to develop the main modules in the dialog system.

Key words: Spoken Dialog Systems, Statistical Models, Dialog Management, User Simulation, System Evaluation

1 Introduction

Nowadays, there are many projects that have developed dialog systems to provide information and other services automatically. In a dialog system of this

Email address: {dgriol,lhurtado,esegarra,esanchis}@dsic.upv.es (David Griol, Lluís F. Hurtado, Encarna Segarra, Emilio Sanchis).

kind, several modules cooperate to perform the interaction with the user: the Speech Recognizer, the Language Understanding Module, the Dialog Manager, the Natural Language Generation module, and the Synthesizer. 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.

Learning statistical approaches to model the different modules that compose a dialog system has been of growing interest during the last decade (Young, 2002). Models of this kind have been widely used for speech recognition and also for language understanding (Levin and Pieraccini, 1995), (Minker et al., 1999), (Segarra et al., 2002), (He and Young, 2003), (Esteve et al., 2003). Even though 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 (Levin et al., 2000), (Torres et al., 2003), (Lemon et al., 2006), (Williams and Young, 2007). These approaches are usually based on modeling the different processes probabilistically and learning the parameters of the different statistical models from a dialog corpus.

Continuous advances in the field of spoken dialog systems make the processes of design, implementation and evaluation of dialog management strategies more and more complex. The motivations for automating dialog learning are focused on the time-consuming process that hand-crafted design involves and the ever-increasing problem of dialog complexity. Statistical models can be trained from real dialogs, modeling the variability in user behaviors. Although the construction and parameterization of the model depend on the expert knowledge of the task, the final objective is to develop dialog systems that have a more robust behavior, better portability, 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 Markov Decision Process (MDP) and reinforcement methods (Levin and Pieraccini, 1997), (Singh et al., 1999), (Levin et al., 2000). The main drawback of this approach is due to the large state space of practical spoken dialog systems, whose representation is intractable if represented directly (Young et al., 2007). Partially Observable MDPs (POMDPs) outperform MDP-based dialog strategies since they provide an explicit representation of uncertainty (Roy et al., 2000). However, they are limited to small-scale problems, since the state space would be huge and exact POMDP optimization is again intractable (Young et al., 2007). An approach that scales the POMDP framework for implementing practical spoken dialog systems by the definition of two state spaces is presented in (Young et al., 2005). Other interesting approaches for

statistical dialog management are based on modeling the system by means of Hidden Markov Models (HMMs) (Cuayáhuitl et al., 2005) or using Bayesian networks (Paek and Horvitz, 2000) (Meng et al., 2003).

Recently, we have presented a statistical approach for the construction of a dialog manager (Hurtado et al., 2006). Our dialog manager is mainly based on the modelization of the sequences of the system and user dialog acts and the introduction of a partition in the space of all the possible sequences of dialog acts. This partition, which is defined taking into account the data supplied by the user throughout the dialog, makes the estimation of a statistical model from the training data manageable.

The confidence measures provided by the recognition and the understanding modules are also taken into account in the definition of this partition. The new system utterance is selected by means of a classification procedure. Specifically, we use neural networks for the implementation of this classification process.

The success of statistical 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 attracted increasing interest in the last decade 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. The user simulator makes it possible to generate a large number of dialogs in a very simple way. Therefore, this technique reduces the time and effort that would be needed for the evaluation of a dialog system each time the system is modified.

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. In the literature, there are several corpus-based approaches for developing user simulators, learning optimal management strategies, and evaluating the dialog system (Scheffler and Young, 2001a) (Pietquin and Dutoit, 2005) (Georgila et al., 2006) (Cuayáhuitl et al., 2006). 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 a statistical approach to dialog management and user simulation. The methodology for developing a user simulator extends our work to model the system behavior. The user turn, which is represented as dialog acts, is selected using the probability distribution provided by a neural network. By means of the interaction of the dialog manager and the user

simulator, an initial dialog corpus can be extended by increasing its variability and detecting dialog situations in which the dialog manager does not provide an appropriate answer. We propose the use of this corpus for evaluating and improving the dialog manager strategy.

Our dialog Manager and user simulator are integrated in a dialog system developed within the framework of the DIHANA project (Benedí et al., 2006). This project undertakes the design and development of a dialog system for the access to an information system using spontaneous speech. The domain of the project is the query to an information system about railway timetables, prices and services in Spanish by telephone. The main goal of the DIHANA project is the development of a robust, distributed and modular dialog system for access to information systems. Specifically, we have tried to make an in-depth study of the methodological aspects in the fields of treatment of spontaneous speech, natural language modeling, language understanding, and dialog management.

The paper is organized as follows. Sections 2 reviews different approaches related to the evaluation of spoken dialog systems. This section focuses on the description of statistical techniques for user simulation. Section 3 briefly presents the main characteristics of the dialog system developed for the DIHANA project. It also describes the corpus and the semantic and dialog-act labeling that is used for learning the statistical models. Section 4 presents our statistical methodology for dialog management. Section 5 describes the extension of this methodology to develop a statistical user simulator. Section 6 presents the evaluation of a dialog corpus acquired using the proposed methodology. Section 7 describes the measures and evaluation of the dialog strategy. Finally, our conclusions are presented.

2 Related work. Evaluation of dialog systems

As the study and development of dialog systems become more complex, it is necessary to develop new measures for their evaluation in order to verify whether or not these systems are effective. It is very difficult to define new procedures and measures that will be unanimously accepted by the scientific community. This field can be considered to be in an initial phase of development. PARADISE (PARAdigm for DIalogue Evaluation System) is the most widely proposed methodology to perform a global evaluation of a dialog system (Walker et al., 1998) (Dybkjaer et al., 2004). This methodology combines different measures regarding task success, dialog efficiency and dialog quality in a single function that measures the yield of the system in direct correlation with user satisfaction. The EAGLES evaluation working group (Expert Advisory Group on Language Engineering Standards) proposes different quantitative and qualitative measures (EAGLES, 1996). In the same line, the DISC

project (Spoken Language Dialogue Systems and Components) (Failenschmid et al., 1999) proposes different measures and criteria to be considered in the evaluation. Finally, a set of 15 criteria to evaluate the system usability can be found in (Dybkjaer and Bernsen, 2000).

Research in techniques for user modeling has a long history within the fields of language processing and spoken dialog systems. Statistical models for modeling user behavior have been suggested as the solution to the lack of the data that is required for training and evaluating dialog strategies. Using this approach, the dialog manager can explore the space of possible dialog situations and learn new potentially better strategies. Methodologies based on learning user intentions have the purpose of optimizing dialog strategies.

In (Eckert et al., 1997, 1998), Eckert, Levin and Pieraccini introduced the use of statistical models to predict the next user action by means of a n-gram model. The proposed model has the advantage of being both statistical and task-independent. Its weak point consists of approximating the complete history of the dialog by a bigram model. In (Levin et al., 2000), the bigram model is modified by considering only a set of possible user answers following a given system action (the Levin model). Both models have the drawback of considering that every user response depends only on the previous system turn. Therefore, the simulated user can change objectives continuously or repeat information previously provided.

In (Scheffler and Young, 1999, 2000, 2001a,b), Scheffler and Young propose a graph-based model. The arcs of the network symbolize actions, and each node represents user decisions (*choice points*). In-depth knowledge of the task and great manual effort are necessary for the specification of all possible dialog paths.

Pietquin, Beaufort and Dutoit combine characteristics of the Scheffler and Young model and Levin model. The main objective is to reduce the manual effort necessary for the construction of the networks (Pietquin and Beaufort, 2005) (Pietquin and Dutoit, 2005). A Bayesian network is suggested for user modeling. All model parameters are hand-selected.

Georgila, Henderson and Lemon propose the use of HMMs, defining a more detailed description of the states and considering an extended representation of the history of the dialog (Georgila et al., 2005). Dialog is described as a sequence of *Information States* (Bos et al., 2003). Two different methodologies are described to select the next user action given a history of information states. The first method uses n-grams (Eckert et al., 1997), but with values of n from 2 to 5 to consider a longer history of the dialog. The best results are obtained with 4-grams. The second methodology is based on the use of a linear combination of 290 characteristics to calculate the probability of every

action for a specific state.

Cuayáhuitl et al. (2005) present a method for dialog simulation based on HMMs in which both user and system behaviors are simulated. Instead of training only a generic HMM model to simulate any type of dialog, the dialogs of an initial corpus are grouped according to the different objectives. A submodel is trained for each one of the objectives, and a bigram model is used to predict the sequence of objectives.

In (Schatzmann et al., 2007a), a new technique for user simulation based on explicit representations of the user goal and the user agenda is presented. The user agenda is a structure that contains the pending user dialog acts that are needed to elicit the information specified in the goal. This model formalizes human-machine dialogs at a semantic level as a sequence of states and dialog acts. An EM-based algorithm is used to estimate optimal parameter values iteratively. In (Schatzmann et al., 2007b), the agenda-based simulator is used to train a statistical POMDP-based dialog manager.

2.1 Evaluation of the simulation techniques

There are no generally accepted criteria for what constitutes a good user simulation model in dialog systems. Typically used methods are adopted from other research fields such as Information Retrieval and Machine Learning. A first classification consists of dividing these techniques into direct evaluation methods and indirect methods (Schatzmann et al., 2006).

Direct methods evaluate the user model by measuring the quality of its predictions. *Recall* measures how many of the actions in the real response are predicted correctly. *Precision* measures the proportion of correct actions among all the predicted actions. The results of the precision and recall obtained from the evaluation of different user models can be found in (Schatzmann et al., 2005a). One drawback of these measures is that they consider a high penalty for the actions that are unseen in the simulated answer, although they could be potentially provided by a real user.

In (Scheffler and Young, 2001b) and (Schatzmann et al., 2006), a set of statistical measures to evaluate the quality of the simulated corpus is proposed. Three dimensions are defined: high-level features (dialog and turn lengths), dialog style (speech-act frequency; proportion of goal-directed actions, grounding, formalities, and unrecognized actions; proportion of information provided, reprovided, requested and rerequested), and dialog efficiency (goal completion rates and times). The simulation presented in (Schatzmann et al., 2007a) is evaluated by testing the similarity between real and simulated data by means of statistical measures (dialog length, task completion rate and dialog perfor-

mance).

In (Georgila et al., 2005), the use of *Perplexity* for the evaluation of the user model is introduced. It determines whether the simulated dialogs contain sequences of actions that are similar to those contained in the real dialogs.

In (Cuayáhuitl et al., 2005), the comparison between the simulated corpus and a corpus acquired with real users is carried out by training a HMM with each corpus and then measuring the similarity between the two corpora on the basis of the distance between the two HMM.

The main objective of indirect methods of evaluation is to measure the *Utility* of the user model within the framework of the operation of the complete system. These methods try to evaluate the operation of the dialog strategy learned by means of the simulator. This evaluation is usually carried out by verifying the operation of the new strategy through a new interaction with the user simulator. Then, the initial strategy is compared with the learned one using the simulator. The main problem with this evaluation resides in the dependence of the acquired corpus on the user model.

Schatzmann et al. (2005b) present a series of experiments that investigate the effect of the user model on simulation-based reinforcement learning of dialog strategies. The bigram, Pietquin and Levin models are trained and tested. The results indicate that the choice of the user model has a significant impact on the learned strategy. The results also demonstrate that a strategy learned with a high-quality user model generalizes well to other types of user models. Lemon and Liu (2007) extend this work by evaluating only one type of stochastic user simulation but with different types of users and under different environmental conditions. This study concludes that dialog policies trained in high-noise conditions perform significantly better than those trained for low-noise conditions.

In (Rieser and Lemon, 2006), an evaluation metric call Simulated User Pragmatic Error Rate (SUPER) is introduced. The consistency, completeness and variation of the user simulation is evaluated.

3 The DIHANA dialog system

Within the framework of the DIHANA project, we have developed a mixed-initiative dialog system to access information systems using spontaneous speech (Griol et al., 2006b). We have built an architecture that is based on the client-server paradigm. The system consists of six modules: an automatic speech recognition (ASR) module, a natural language understanding (NLU) mod-

ule, a dialog manager (DM), a database query manager, a natural language generation module (NLG), and a text-to-speech converter.

We are currently using the CMU Sphinx-II system (*cmusphinx.sourceforge.net*) in our speech recognition module. As in many other dialog systems, the semantic representation chosen for the task is based on the concept of frame (Minsky, 1975). Frames are a way of representing semantic knowledge. A frame is a structure for representing a concept or situation. Each concept in a domain has usually associated a group of attributes (slots) and values (Fikes and Kehler, 1985). In the semantic representation defined for DIHANA, one or more concepts represent the intention of the utterance, and a sequence of attribute-value pairs contains the information about the values given by the user. Therefore, the NLU module takes the sentence supplied by the recognition process as input and generates one or more frames as output.

The NLG module translates the semantic representations of the system dialog acts to sentences in Spanish. It uses templates and combines rules to make this translation. The input of this module is composed of concepts and attributes (as in the NLU module) with confidence measures associated to each one of the system dialog acts. These measures allow us to generate detailed answers in natural language. In these answers, the attributes may or may not be mentioned depending on their associated confidence.

The technique that we use consists of having a set of templates associated to each one of the different dialog acts, in which the names of the attributes are reflected. These names are replaced by the values recognized in order to generate the final answer for the user. Each dialog act has its set of associated templates so that the most accurate answer is given in every possible situation for each one of the queries.

For speech output, we have integrated the Festival speech synthesis system (*www.cstr.ed.ac.uk/projects/festival*). The specific information relative to our task is stored in a PostGres database using information that is dynamically extracted from the web.

Our dialog system has two operation modes. First, the system uses the ASR and the NLU modules for the normal interaction between the system and the real users. Second, the system allows the automatic acquisition of dialogs by means of the user simulator module. Figure 1 shows the modular architecture of our system: (1) the interaction with real users and (2) the operation with the user simulator.

The behavior of the main modules of the dialog system is based on statistical models that are learned from a dialog corpus that was acquired and labeled within the framework of the DIHANA project.

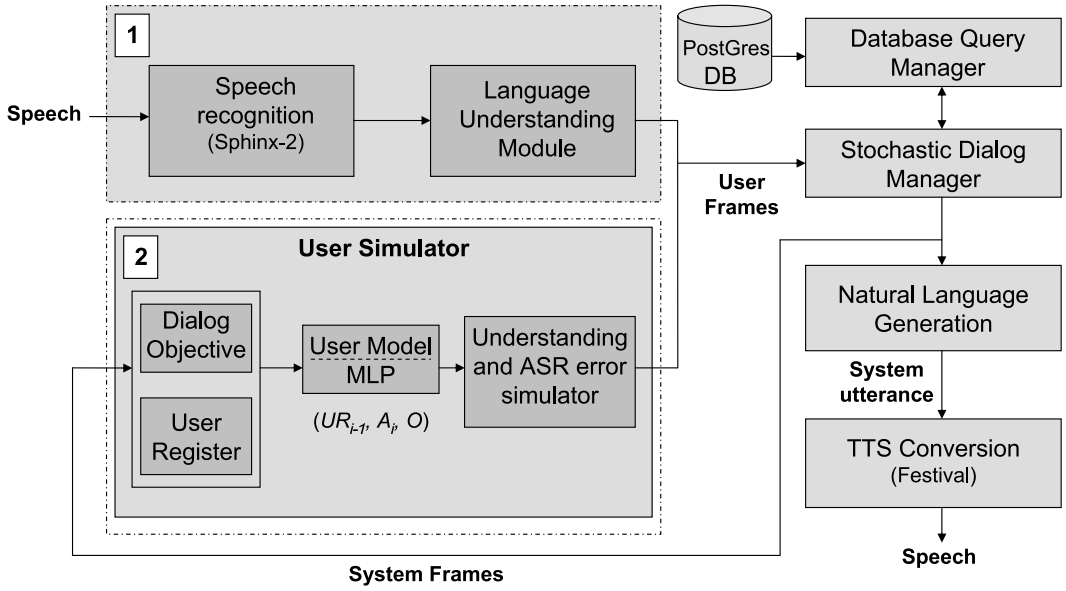


Fig. 1. Architecture of the DIHANA dialog system. (1) Interaction with real users. (2) Operation with the user simulator

3.1 The DIHANA dialog corpus

A set of 900 dialogs was acquired in the DIHANA project. Although this corpus was acquired using a Wizard of Oz technique (WOz), real speech recognition and understanding modules were used. A corpus of 200 dialogs acquired for a previous project with a similar task (Bonafonte et al., 2000) was used to generate the language and acoustic models for the ASR module and to train a statistical model for the NLU module to carry out the acquisition. Some categories were incorporated to increase the coverage of the language model of the ASR module and for the NLU module. However, there were some situations in the acquisition of the DIHANA corpus in which these two modules failed. In these situations, the WOz worked considering only the speech output.

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; that is to say, the user must obtain information about only one type of the possible queries to the system (for instance, to obtain timetable information from an origin city to a destination for a specific date). Type S2 defined two objectives for the dialog. In these scenarios, the user must obtain information about two queries defined in the task; for instance, asking for timetables and prices given a specific origin, destination and date.

Five files were stored for each acquired dialog: the output of the recognizer,

the output of the understanding module, the answer (dialog act) generated by the system, the values of the attributes during the successive turns, and the queries made to the database. This information is used to model the behavior of the system depending on the succession of dialog acts, the semantic representation of the user turn (information provided by the NLU module, including confidence scores).

The characteristics of the acquired corpus are shown in Table 1.

Number of users	225
Number of dialogs per user	4
Number of user turns	6,280
Average number of user turns per dialog	7
Average number of words per user turn	7.7
Vocabulary	823
Duration of the recording (hours)	10.8

Table 1
Main characteristics of the DIHANA corpus

3.2 The Wizard of Oz strategy

The WOz technique (Fraser and Gilbert, 1991) allows the acquisition of a dialog corpus with real users without having a complete dialog system, for which the dialog corpus would be necessary. We chose human-wizard rather than human-human dialogs since people behave differently toward (what they perceive to be) machines and other people. This is discussed in (Jönsson and Dahlbick, 1988) and validated in (Doran et al., 2001) and (Lane et al., 2004).

Three Spanish universities participated in the acquisition of the corpus for the DIHANA project. Each university used a different WOz to carry out the acquisition. The three WOz have multiple information sources to determine the next system action: they heard the sentence pronounced by the user, received the output generated by the ASR module (sequence of words and confidence scores), the semantic interpretation generated by the NLU module the sequence of words recognized, and had a data structure that contains the complete history of the dialog.

In the acquisition, the WOz strategy was not constrained by a script. The three WOz were instructed with only a basic set of rules defined to acquire a corpus without excessive dispersion among them. These rules were recommended to be used by the WOz and are based on considering the confidence scores provided by the NLU module and a data structure that we call Dialog

Register (DR). The DR contains the concepts and attributes provided by the user throughout the previous history of the dialog. This information also includes confidence scores (García et al., 2003), which are used by the WOz to evaluate the reliability of the concepts and attributes generated by the NLU module.

Two different situations for the dialog were considered. The dialog is in a *safe state* when all the data of the DR have a confidence score that is higher than the fixed threshold. The dialog is in a *uncertain state* when one or more data of the DR have a confidence that is lower than the threshold. A different set of recommended rules was defined for each state.

The recommendations for a *safe state* were:

- To make an implicit confirmation and a query to the database if the user has already provided the objective of the dialog and, at least, the minimum necessary information (e.g. *I provide you with railway timetables from Madrid to Bilbao in first class*).
- To request the dialog objective or some of the required information.
- To select a mixed confirmation to give naturalness to the dialog. This selection is made on a variable number of safe turns instead of an implicit confirmation and query to the database. In a mixed confirmation, there are several items, and the confirmation only affects one of them (e.g. *You want railway timetables to Valencia. Do you want to leave from Madrid?*).

The recommendations for a *uncertain state* were:

- To make an explicit confirmation of the first uncertain item that appears in the DR (e.g. *Do you want to travel to Barcelona?*).
- To select a mixed confirmation to give naturalness to the dialog instead of a explicit confirmation. This is done on a variable number of uncertain turns of dialog instead of an explicit confirmation.

A set of possible system answers were defined for each of the interactions stated above.

3.3 Corpus labeling

In order to learn statistical models, the dialogs of the DIHANA corpus were labeled in terms of dialog acts. As stated above, in the case of user turns, the dialog acts correspond to the classical frame representation of the meaning of the utterance. For the DIHANA task, we defined eight concepts and ten attributes. The eight concepts are divided into two groups:

(1) *Task-dependent concepts*: they represent the concepts the user can ask for (*Hour*, *Price*, *Train-Type*, *Trip-Time*, and *Services*).
(2) *Task-independent concepts*: they represent typical interactions in a dialog (*Affirmation*, *Negation*, and *Not-Understood*).

The attributes are: *Origin*, *Destination*, *Departure-Date*, *Arrival-Date*, *Class*, *Departure-Hour*, *Arrival-Hour*, *Train-Type*, *Order-Number*, and *Services*.

Figure 2 shows an example of the semantic interpretation of an input sentence.

Input sentence:

[SPANISH] Sí, me gustaría conocer los horarios para mañana por la tarde desde Valencia.
[ENGLISH] *Yes, I would like to know the timetables for tomorrow evening leaving from Valencia.*

Semantic interpretation:

(*Affirmation*)
(*Hour*)
Origin: Valencia
Departure-Date: Tomorrow
Departure-Hour: Evening

Fig. 2. An example of the labeling of a user turn in the DIHANA corpus

Three levels of labeling were defined for the system dialog acts. The first level describes the general acts of any dialog, independently of the task. The second level represents the concepts and attributes involved in the turn and is task-dependent. The third level represents the values of the attributes given in the turn. The following labels were defined for the first level: *Opening*, *Closing*, *Undefined*, *Not-Understood*, *Waiting*, *New-Query*, *Acceptance*, *Rejection*, *Question*, *Confirmation*, and *Answer*. The labels defined for the second and third level were the following: *Departure-Hour*, *Arrival-Hour*, *Price*, *Train-Type*, *Origin*, *Destination*, *Date*, *Order-Number*, *Number-Trains*, *Services*, *Class*, *Trip-Type*, *Trip-Time* and, *Nil*. Each turn of the dialog was labeled with one or more dialog acts. Having this kind of detailed dialog act labeling and the values of attributes obtained during a dialog, it is straightforward to construct a sentence in natural language.

Some examples of the dialog act labeling of the system turns are shown in Figure 3. In these examples, the third level contains the sequence of attribute-value pairs (e.g. *Origin[Valencia]*) involved in the system turns. From now on, in the interest of clarity, in the examples of the labeling of the system turns, we will omit the values of the attributes.

[SPANISH] Bienvenido al servicio de información de trenes. ¿En qué puedo ayudarle? [ENGLISH] *Welcome to the railway information system. How can I help you?* (*Opening:Nil:Nil*)
[SPANISH] Quiere horarios de trenes a Granada, ¿desde Valencia? [ENGLISH] *Do you want timetables to Granada, from Valencia?* (*Confirmation:Departure-Hour:Destination[Granada]*) (*Confirmation:Origin:Origin[Valencia]*)
[SPANISH] El único tren es un Euromed que sale a las 0:27. ¿Desea algo más? [ENGLISH] *There is only one train, which is a Euromed, that leaves at 0:27. Anything else?* (*Answer:Departure-Hour:Departure-Hour:Departure-Hour[0.27],Number-Trains[1],Train-Type[Euromed]*) (*New-Query:Nil:Nil*)

Fig. 3. Labeling examples of system turns from the DIHANA corpus

4 Our approach for dialog management

We have developed a Dialog Manager (DM) based on the statistical modelization of the sequences of dialog acts (user and system dialog acts). A labeled corpus of dialogs is used to estimate the statistical DM. Depending on the number of dialog acts, and thus, on the amount of information represented in a dialog act, the possibility of obtaining a good model can vary. If we consider only a small number of dialog acts representing general actions in a dialog, we could obtain a well-trained model. However, the information represented in that model is not enough to completely manage the dialog, and the specific information related to the task must be provided to the DM through a set of hand-made rules. If we label a turn using dialog acts that take into account not only the general purpose of the sentences but also the specific request related to the task (the concepts and attribute values observed in the turn), we could use this detailed representation to learn an operative DM. The problem in this last case is that the number of dialog acts increases exponentially in relation to the number of concepts (and attributes), and the space of the different situations of the dialog to be taken into account is too large.

We have developed a statistical DM that can generate system turns based only on the information supplied by the user turns and the information contained in the model. All this information is acquired from the labeled corpus in the training phase. Some techniques have been applied to tackle the problem of the size of the space of different situations of the dialog. A formal description of the proposed statistical model is as follows:

Let A_i be the output of the dialog system (the system answer) at time i, expressed in terms of dialog acts. Let U_i be the semantic representation of the user turn (the result of the understanding process of the user input) at time i, expressed in terms of frames. A dialog begins with a system turn that welcomes the user and offers him/her its services. We consider a dialog to be 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. From now on, we refer to a pair (A_i, U_i) as S_i, the state of the dialog sequence at time i.

In this framework, we consider that, at time i, the objective of the dialog manager is to find the best system answer A_i. This selection is a local process for each time i and takes into account the sequence of dialog states preceding time i. This selection is made by maximizing:

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

where set $\mathcal{A}$ contains all the possible system answers. As the number of all possible sequences of states is very large, we establish a partition in the space of sequences of states (i.e., in the history of the dialog preceding time i).

Let DR_i be the dialog register at time i. As stated in the previous section, the dialog register is defined as a data structure that contains the information about concepts and attribute values provided by the user throughout the previous history of the dialog. All the information captured by the DR_i at a given time i is a summary of the information provided by the sequence $S_1, \cdots, S_{i-1}$. Note that different state sequences can lead to the same DR.

For a sequence of states of a dialog, there is a corresponding sequence of DR:

$$\begin{array}{ccccccccc} & S_1, & & \cdots, & & S_i, & \cdots, & & S_n \\ \uparrow & & \uparrow & & \uparrow & & & \uparrow & \\ DR_0 & & DR_1 & & DR_{i-1} & & & DR_{n-1} & \end{array}$$

where DR_0 captures the default information of the dialog manager (*Origin* and *Class* in our system), and the values of the following DR are updated taking into account the information supplied by the evolution of the dialog.

Taking into account the concept of the DR, we establish a partition in the space of sequences of states such that: two different sequences of states are

considered equivalent if they lead to the same DR_i. We obtain a great reduction in the number of different histories in the dialogs at the expense of a loss in the chronological information. We consider this to be a minor loss because the order in which the information is supplied by the user is not a relevant factor in determining the next system answer A_i.

After applying the above considerations and establishing the equivalence relation in the histories of dialogs, the selection of the best A_i is given by:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | DR_{i-1}, S_{i-1}) \tag{2}$$

Each user turn supplies the system with information about the task; that is, he/she asks for a specific concept and/or provides specific values for certain attributes. However, a user turn could also provide other kinds of information, such as task-independent information. This is the case of turns corresponding to *Affirmation*, *Negation* and *Not-Understood* dialog acts. This kind of information implies some decisions which are different from simply updating the DR_{i-1}. For that reason, for the selection of the best system answer A_i, we take into account the DR that results from turn 1 to turn $i-1$, and we explicitly consider the last state S_{i-1}. Our model can be extended by incorporating additional information to the DR, such as some chronological information (e.g. number of turns up to the current turn) or user profiles (e.g. naïve or experimented users or user preferences).

Statistical approaches must tackle the problem of modeling all the possible situations that can occur during a dialog (the problem of coverage of the model) using only the training corpus. The possibility of the user uttering an unexpected sentence must also be considered in the design of the dialog manager. In the first version of our dialog manager (Hurtado et al., 2005), we assumed that if the dialog situation was already observed in the training corpus, the assigned system answer was the same as the corresponding answer observed in training. However, if this situation was not observed in the training corpus, we defined a distance measure in order to assign it an observed event, and consequently, a system answer. The objective of the distance is to select the closest pair (DR', S') that is included in the statistical model given a pair (DR, S) that was unseen in the training phase. The definition of the distance measure is the following:

$$d((DR, S), (DR', S')) \approx d(DR, DR') = \sum_{k=1}^{n} f(dr_k, dr'_k)$$

First, we assume that the distance is independent of the terms S and S'. Second, in relation to the distance between codified DRs in the definition

of function f, we assume that: the insertion of an attribute value that is not actually provided by the user in the dialog is more penalized than the deletion of such an attribute value. It is better to ask repeatedly about some information previously given than to ask the user about some values not given by him/her. The evaluation of this function is presented in (Hurtado et al., 2005).

In (Hurtado et al., 2006), we present a new proposal for adapting the model to these unseen situations. The partitioned space of the possible sequences of dialog acts that is estimated during the training phase is partitioned a second time into classes. Each class groups together all the sequences that provide the same set of system actions (answers). After the training phase is finished, a set of classes $\mathcal{C}$ is defined. In this paper, we propose that, given a new user turn, the statistical dialog model makes the assignation of a system answer according to the result of a classification process. Every dialog situation is classified into a class of this set $c \in \mathcal{C}$, and the answer of the system at that moment is the answer associated with this selected class.

The classification function can be defined in several ways. We have evaluated four different definitions of such a function: a multinomial naive Bayes classifier, n-gram based classifier, a classifier based on grammatical inference techniques and a classifier based on neural networks (Griol et al., 2006a). The best results were obtained using a multilayer perceptron (MLP) (Rumelhart et al., 1986) where the input layer holds the input pair (DR_{i-1}, S_{i-1}) corresponding to the dialog register and the state. The values of the output layer can be seen as an approximation of the a posteriori probability of belonging to the associated class $c \in \mathcal{C}$.

Figure 4 shows the operation of the dialog manager developed for the DIHANA project. The frames generated by the NLU module after each user turn and the last system answer are used to generate the pair (DR_{i-1}, S_{i-1}). The codification of this pair is the input of a MLP that provides the probabilities of selecting each one of the system answers defined for the DIHANA task, given the current situation of the dialog (represented by this pair).

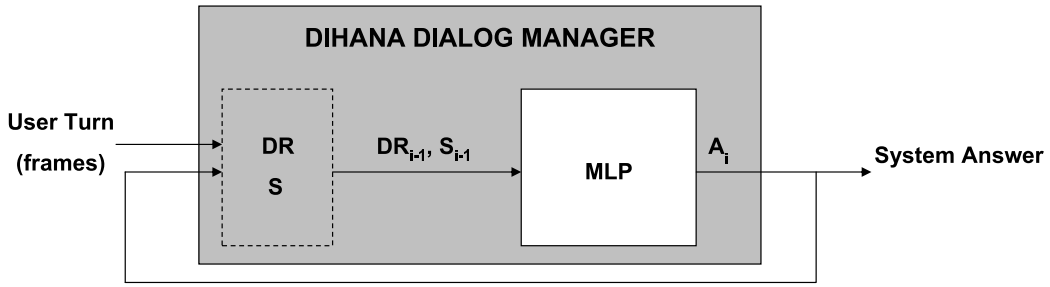


Fig. 4. Graphical scheme of the dialog manager developed for the DIHANA project

4.1 Dialog Register representation

For the DM to determine the next answer, we have assumed that the exact values of the attributes are not significant. They are important for accessing the Database and for constructing the output sentences of the system. However, the only information necessary to determine the next action by the system is the presence or absence of concepts and attributes. Therefore, the information we used from the DR is a codification of this data in terms of three values, $\{0, 1, 2\}$, for each field in the DR according to the following criteria:

- **0**: The concept is unknown or the value of the attribute is not given.
- **1**: The concept or attribute is known with a confidence score that is higher than a given threshold. The confidence score is given during the recognition and understanding processes and can be increased by means of confirmation turns.
- **2**: The concept or attribute is activated with a confidence score that is lower than the given threshold.

The DR defined for the DIHANA task is a sequence of 15 fields, corresponding to the five concepts (*Hour, Price, Train-Type, Trip-Time, Services*) and ten attributes (*Origin, Destination, Departure-Date, Arrival-Date, Departure-Hour, Arrival-Hour, Class, Train-Type, Order-Number, Services*) defined for the task.

Table 2 shows the reduction in the number of states that is achieved for the DIHANA corpus with the introduction of the DR.

Different sequences $S_1, \cdots, S_{i-1}$	4,290
Different DR	261
Different pairs (DR, S)	1,212

Table 2
Reduction in the space of states with the introduction of the DR in the DIHANA task

4.2 MLP classifier

MLPs are the most common artificial neural networks used for classification (Castro et al., 2003). In order to apply a MLP to select the system answer, as previously stated, the input layer holds a codification of the input pair (DR_{i-1}, S_{i-1}). The representation defined for this pair is as follows:

- The first two levels of the labeling of the last system answer (A_{i-1}): This information is modeled using a variable, which has as many bits as possible

combinations of the values of these two levels (51) (see Section 3.3).

$$\vec{x}_1 = (x_{1_1}, x_{1_2}, x_{1_3}, \cdots, x_{1_{51}}) \in \{0,1\}^{51}$$

- Dialog register (DR_{i-1}): As previously stated, fifteen characteristics can be observed in the DR (5 concepts and 10 attributes). Each one of these characteristics can take the values $\{0,1,2\}$. Therefore, every characteristic has been modeled using a variable with three bits.

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0,1\}^3 \, i = 2, ..., 16$$

- Task-independent information (*Affirmation*, *Negation*, and *Not-Understood* dialog acts): These three dialog acts have been coded with the same codification used for the information in the DR; that is, each one of these three dialog acts can take the values $\{0,1,2\}$. Therefore, this information is modeled using three variables with three bits.

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0,1\}^3 \, i = 17, ..., 19$$

For the process of classification, the number of output units of the MLP is defined as the number of classes, $|C|$, and the input layer must hold the input samples (DR_{i-1}, S_{i-1}). For uniclass samples, the activation level of an output unit in the MLP can be interpreted as an approximation of the a posteriori probability that the input sample belongs to the corresponding class (Rumelhart et al., 1986) (Bishop, 1995). Therefore, given an input sample x, the trained MLP computes $g_c(\mathbf{x}, \omega)$ (the c-th output of the MLP with parameters ω given the input sample $\mathbf{x}$), which is an approximation of the a posteriori probability $P(c|\mathbf{x})$. Thus, for MLP classifiers we can use the uniclass classification rule as:

$$\hat{c} = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \, P(c|\mathbf{x}) \approx \underset{c \in \mathcal{C}}{\operatorname{argmax}} \, g_c(\mathbf{x}, \omega)$$

where the variable x, which holds for the pair (DR_{i-1}, S_{i-1}), can be represented using the vector of characteristics:

$$\vec{x} = (\vec{x}_1, \vec{x}_2, \vec{x}_3, \cdots, \vec{x}_{19})$$

4.3 An example of a dialog

This section shows an example of a dialog acquired using the statistical dialog manager presented in this paper. S stands for "System turn" and U for "User turn". TDI and TII respectively make reference to the Task-Dependent and Task-Independent Information provided by the NLU module. A 0.5 threshold

is used in the example to determine the data reliability. This value can be modified depending on the reliability that the DM must have based on the information provided by the NLU module. The higher the threshold value, the higher the number of data confirmations. The confidence scores provided by the NLU module to determine the reliability are shown between brackets.

The dialog begins with a greeting turn (S_1). The three-level labeling of this turn is (*Opening:Nil:Nil*). The initial DR contains the *origin* and the *ticket class* as a default information (represented by the two "1" that can be observed in DR_0).

S_1: **Welcome to the railway information system. How can I help you?**

A_1: (Opening:Nil:Nil)

DR_0: 00000-1000001000

In the first turn, the user provides the concept *Hour* and the attribute *destination*. This information is used to update DR_0 and to obtain DR_1. The input of the MLP is generated using DR_1, the last two levels of the labeling of the last system turn (A_1), and the task-independent information (none in this case). The output selected for the MLP consists of requiring the departure date.

U_1: **I want to know timetables to Barcelona.**

TDI: (*Hour*) [0.7] *Destination*:Barcelona [0.9]

TII: ()

DR_1: 10000-1100001000

10000-1100001000 + Opening:Nil + () → A_2: (Question:Departure-Date:Nil)

S_2: **Tell me the departure date.**

In the following turn, the user provides the date. A low confidence score is associated with this data. Then, DR_2 is obtained by adding a "2" value at the *departure-date* slot. The input of the MLP is generated as stated above. A confirmation of the *departure-date* is selected as an output.

U_2: **Tomorrow.**

TDI: *Date*:Tomorrow [0.3]

TII: ()

DR_2: 10000-1120001000

10000-1120001000 + Question:Departure-Date + () → A_3: (Confirmation:Departure-Date:Departure-Date)

S_3: **Do you want to leave tomorrow?**

In the third turn, the user confirms this value. The NLU module supplies an *Affirmation* dialog act. As a result of the classification made by the MLP, a query to the database is selected. The system provides the timetable information required by the user.

Finally, the user mentions that s/he does not want anything else. A *closing* dialog act is selected as the result of the classification.

U_3: **Yes.**

TDI:

TII: (*Affirmation*) [0.7]

DR_3: 10000-1120001000

10000-1110001000 + Confirmation:Departure-Date + (*Affirmation*) → A_4: (Answer:Departure-Hour:Number-Trains,Train-Type,Departure-Hour) (New-Query:Nil:Nil)

S_4: **There are several trains. The first one leaves at 08:54 and the last one at 23:45. Anything else?**

U_4: **No, thank you.**

TDI: ()

TII: *(Negation)* [0.8]

DR_4: 10000-1120001000

10000-1110001100 + Answer-New-Query:Departure-Nil + (*Negation*) → A_5: (Closing:Nil:Nil)

S_5: **Thanks for using this service. Have a good trip.**

5 Extending our approach to model user behavior

In our system, the user simulator replaces the functions performed by the ASR and the NLU modules. It generates frames in the same format defined for the output of the NLU module, i.e, in the format expected by the DM.

The methodology that we have developed for user simulation extends our work for developing a statistical methodology for dialog management. The user answers are generated taking into account the information provided by the simulator throughout the history of the dialog, the last system turn, and the objective(s) predefined for the dialog. A labeled corpus of dialogs is used to estimate the user model.

Given the representation of a dialog as a sequence of pairs (*system-turn, user-turn*), the objective of the user simulator at time i is to find an appropriate user answer U_i. This selection, which is a local process for each time i, takes into account the sequence of dialog states that precede time i, the system answer at time i, and the objective of the dialog $\mathcal{O}$. If the most probable user answer U_i is selected at each time i, the selection is made using the following maximization:

$$\hat{U}_i = \underset{U_i \in \mathcal{U}}{\operatorname{argmax}} P(U_i | S_1, \cdots, S_{i-1}, A_i, \mathcal{O}) \tag{3}$$

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

As the number of possible sequences of states is very large, we establish a partition in this space (i.e., in the history of the dialog preceding time i).

Let UR_i be the user register at time i. The user register is defined as a data structure that contains the information about concepts and attribute values

provided by the user throughout the previous history of the dialog. The information contained in UR_i is a summary of the information provided by the user until time i.

The partition that we establish in this space is based on the assumption that two different sequences of states are equivalent if they lead to the same UR. After applying the above considerations and establishing the equivalence relations in the histories of the dialogs, the selection of the best U_i is given by:

$$\hat{U}_i = \underset{U_i \in \mathcal{U}}{\operatorname{argmax}} P(U_i | UR_{i-1}, A_i, \mathcal{O}) \tag{4}$$

For the DIHANA task, the variable $\mathcal{O}$ is modeled taking into account the different types of scenarios defined for the acquisition of the DIHANA corpus. Type S1 scenarios can be decomposed into five cases, depending on the objective defined for the scenario (*Hour*, *Price*, *Train-Type*, *Trip-Time*, or *Services*). Type S2 can be decomposed into ten cases, depending on the two objectives defined for the scenario (the different combinations of the previous five objectives in pairs). Thus, the variable $\mathcal{O}$ has been modeled for the DIHANA task using a variable that has the same number of bits as the number of possible objectives defined for the S1 and S2 scenarios (15).

Table 3 shows the reduction obtained in the number of states for the DIHANA corpus by introducing the UR.

Different sequences $S_1, \cdots, S_{i-1}$	4,290
Different UR	232
Different pairs $(UR, A, \mathcal{O})$	933

Table 3
Reduction in the space of states with the introduction of the UR in the DIHANA task

As in our previous work on dialog management, we propose using a MLP to make the assignation of a user turn. The input layer receives the current situation of the dialog, which is represented by the term $(UR_{i-1}, A_i, \mathcal{O})$ in Equation 4. The values of the output layer can be viewed as the a posteriori probability of selecting the different user answers defined for the simulator given the current situation of the dialog. The choice of the most probable user answer of this probability distribution leads to Equation 4. In this case, the user simulator will always generate the same answer for the same situation of the dialog. Since we want to provide the user simulator with a richer variability of behaviors, we base our choice on the probability distribution supplied by the MLP on all the feasible user answers.

The DIHANA corpus includes information about the errors that were introduced by the ASR and the NLU modules during the acquisition. This information also includes confidence measures, which are used by the DM to evaluate the reliability of the concepts and attributes generated by the NLU module.

An error simulator module has been designed to perform error generation. The error simulator modifies the frames generated by the user simulator once the UR is updated. These modified frames (with errors) are used by the dialog manager to update the dialog register. Therefore, the UR does not include errors but the DR could. In addition, the error simulator adds a confidence score to each concept and attribute in the frames. 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. As future work, we want to make a more detailed study of the errors introduced in our corpus.

6 Acquisition and evaluation of a simulated dialog corpus

We have used the interaction of the statistical user simulator and dialog manager developed for the DIHANA project to acquire a simulated dialog corpus, following the architecture presented in Figure 1. To carry out the evaluation of the simulation process, 50,000 dialogs of each one of the two types of scenarios defined (Type S1 and Type S2) were generated.

Three criteria were defined for closing the dialog. The first criterion consists of finalizing the dialog when the number of system turns exceeds a threshold. The second criterion is used when an error warning is generated by the database query module or the NLG module. The third criterion is applied to generate a user request to close the dialog when the objective of the dialog has been achieved. The successful dialogs are those that end when the third criterion is applied.

We defined five measures for the evaluation of the simulated dialogs:

(1) the number of successful dialogs (SD),
(2) the average number of turns per dialog (NT),
(3) the number of different successful dialogs (DD),
(4) the number of turns of the shortest dialog (TS),
(5) the number of simulated dialogs that are contained in our initial corpus (CD).

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 objectives.

Table 4 shows the values of the different measures obtained from the simulation of 100,000 dialogs.

Type	SD	NT	DD	TS	CD
S1	18,400	10.4	11,550	5	224
S2	4,214	12.1	3,833	7	42

Table 4
Evaluation of the simulated corpus

Taking the two types of scenarios into account, the simulator generated a total of 22,614 dialogs that achieved their objectives (22.6%). The number of simulated dialogs that were initially contained in the WOz corpus, *CD*, corresponds to only a small partition of 266 dialogs (1.2%). It can be observed that Type S2 dialogs are more complicated to simulate since two objectives must be satisfied.

6.1 Dialog examples

Next, two examples of dialogs that compose the acquired corpus are shown. A sentence in natural language, which is equivalent to the labeled system turn selected by the dialog manager, has been incorporated to clarify the explanation.

Table 5 shows an example of a dialog corresponding to a scenario with only one objective (to know timetables from Valencia to Barcelona for May the 30th).

S_1**:** (Opening:Nil:Nil)
Welcome to the railway information system. How can I help you?
U_1**:** Departure-Hour [0.9]
Destination: Barcelona [0.9]
Departure-Date: 05-31-2007 [0.1]
Arrival-Hour: 08.00-23.00 [0.9]
S_2**:** (Confirmation:Date:Date)
Do you want to leave on May the 31st?
U_2**:** Negation[0.9]
Departure-Date: 05-30-2007 [0.9]
S_3**:** (Answer:Departure-Hour:Departure-Hour)(New-Query:Nil:Nil)
{*Timetables information*} *Anything else?*
U_3**:** Negation
S_4**:** (Closing:Nil:Nil)
Thanks for using this service. Have a good trip.

Table 5
An example of a Type S1 dialog extracted from the simulated 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).

In this first turn, the error simulator introduces an error value in the *Departure-Date* slot (it changes day 30 for day 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 (U_2). In this turn, the error simulator assigns a high confidence value to the information provided by the user. In the following system turn (S_3), 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 to close the dialog in the following turn (U_3).

Table 6 shows an example of a dialog corresponding to a scenario with two objectives (to know timetables and prices from Valencia to Barcelona for May the 30th).

In the first turn, the user simulator selects to make a query about timetables and provides the *Origin*, *Destination* and the *Departure-Date*. In addition, it provides additional information defined in the scenario (the *Train-Type* attribute). For this turn, the error simulator introduces an error in the *Origin* (it changes Valencia for Madrid) and assigns a high confidence to this value.

In the following system turn (S_2), the dialog manager asks the simulated user to consult timetables leaving from Madrid. In the following turn (U_2), the user simulator consults the objective and provides the *Origin* again. After this turn, the system makes a query about timetables to the database (S_3).

The user simulator verifies in the U_3 turn that the objective of the dialog has not been completed. In this turn, it selects to make a query about prices, providing the *Origin* and the *Destination* again. It also incorporates the *Departure-Hour* as additional information. In the following system turn (S_4), the dialog manager makes a confirmation about prices leaving from Valencia. Once the objective of the dialog is verified, the user simulator selects *Affirmation* (U_4). Then, the system selects to carry out a database query about prices (S_5). As the necessary information is available, the database query module carries out the query, and the dialog manager completes the objectives for the dialog. Having this information, the user simulator selects a request to close

S_1: (Opening:Nil:Nil)
Welcome to the railway information system. How can I help you?
U_1: Departure-Hour [0.9]
Origin: Madrid [0.9]
Destination: Barcelona [0.9]
Departure-Date: 05-30-2007 [0.9]
Train-Type: Euromed [0.9]
S_2: (Confirmation:Departure-Hour:Nil)(Confirmation: Origin:Origin)
Do you want timetables leaving from Madrid?
U_2: Negation[0.9]
Origin:Valencia [0.9]
S_3: (Answer:Departure-Hour:Departure-Hour)(New-Query:Nil:Nil)
{*Timetables information*} *Anything else?*
U_3: Price [0.1]
Origin: Valencia [0.9]
Destination: Barcelona [0.9]
Departure-Hour: 08.00-23.00 [0.9]
S_4: (Confirmation:Price:Nil)(Confirmation:Origin:Origin)
Do you want to know the price leaving from Valencia?
U_4: Affirmation [0.9]
S_5: (Answer:Price:Price) (New-Query:Nil:Nil)
{*Prices information*} *Anything else?*
U_5: Negation
S_6: (Closing:Nil:Nil)
Thanks for using this service. Have a good trip.

Table 6
An example of a Type S2 dialog extracted from the simulated corpus

the dialog in the following turn (U_5).

7 Evaluation of the dialog strategy

We propose four measures to evaluate the evolution of the dialog strategy once the simulated dialogs are used to reestimate it.

The first measure, which we call *%unseen*, makes reference to the percentage of unseen situations, i.e., the dialog situations that are present in the test partition but are not present in the corpus used for learning the DM.

The other three 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 corpus (the answer provided by the WOz). This way, the evaluation is carried out turn by turn. These three measures are:

- *%strategy*: the percentage of answers provided by the DM that exactly follow the strategy defined for the WOz to acquire the training corpus;
- *%coherent*: 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 defined for the WOz.
- *%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 WOz. On the other hand, the measures *%coherent* 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 dialog manager suddenly ends the interaction with the user, a query to the database is generated without the required information, etc).

Figure 5 shows the measures defined to evaluate the behavior of the dialog manager and the improvement in the dialog strategy.

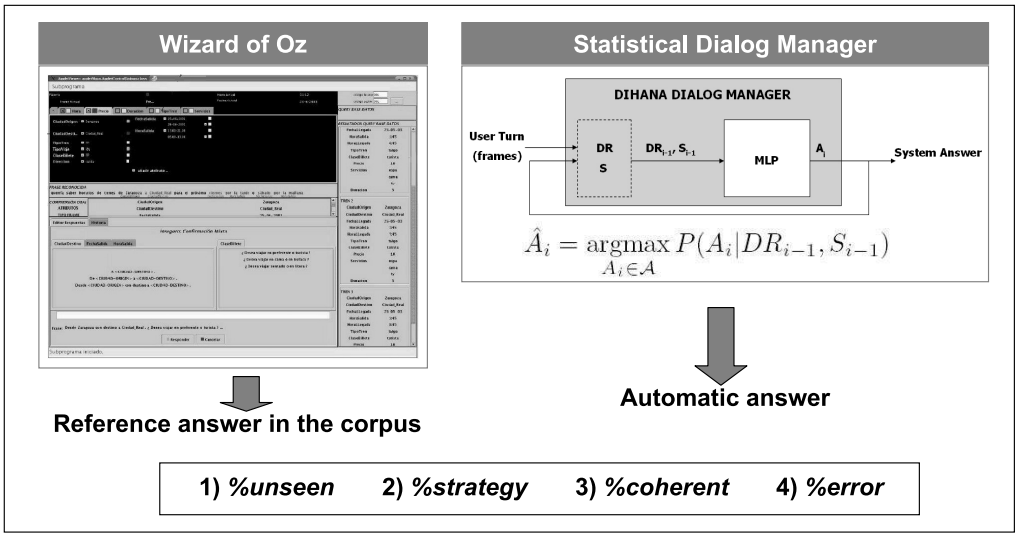


Fig. 5. Measures defined for the evaluation of the statistical dialog manager

First, we evaluated the behavior of the original DM that was learned using the corpus obtained by WOz. 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 1,232 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. MLPs were trained using the backpropagation with momentum algorithm (Rumelhart et al., 1986). The topology used was two hidden layers with 110 units each. Table 7 shows the results of the evaluation.

	%unseen	%strategy	%coherent	%error
System answer	19.68%	90.11%	97.45%	2.55%

Table 7
Results of the evaluation of the initial DM

The results of the *%strategy* and *%coherent* measures show the satisfactory operation of the developed dialog manager. 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 90.11%.

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.

7.1 Evolution of the dialog strategy

We have evaluated the evolution of the DM when the successful simulated dialogs were incorporated to 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 DIHANA corpus (20% of the samples). Table 8 shows the results of the evaluation of the DM model after the successful dialogs were incorporated to the training corpus.

	%unseen	%strategy	%coherent	%error
System answer	14.25%	83.64%	98.84%	1.16%

Table 8
Results of the evaluation of the DM obtained after the dialog simulation

It can be observed that the number of unseen situations was reduced by 5%, as expected with the addition of the simulated dialogs. The evolution of the *%strategy* and *%coherent* measures shows how the dialog manager can move away from an initial strategy by increasing the number of answers that are coherent with the current situation in the dialog. The simulated dialogs also show the improvement of the dialog strategy with a reduction in the *%error* measure (from 2.56% to 1.16%).

Figure 6 and Figure 7 respectively show how the number of unseen situations (*%unseen*) and erroneous system answers (*%error*) decreased when the training corpus was enriched by adding the simulated dialogs, which is the

expected behavior. These measures continued to decrease until 60,000 dialogs were simulated.

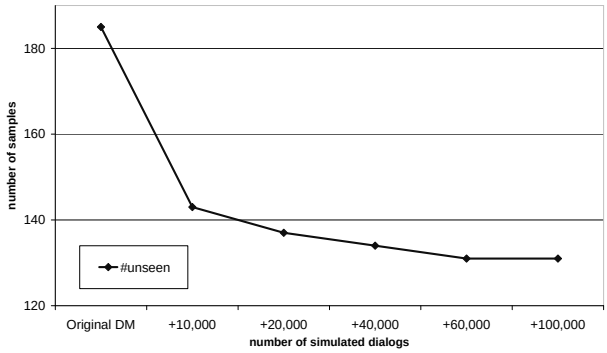


Fig. 6. Evolution of the number of unseen situations (*#unseen*) with regard to the incorporation of new simulated dialogs.

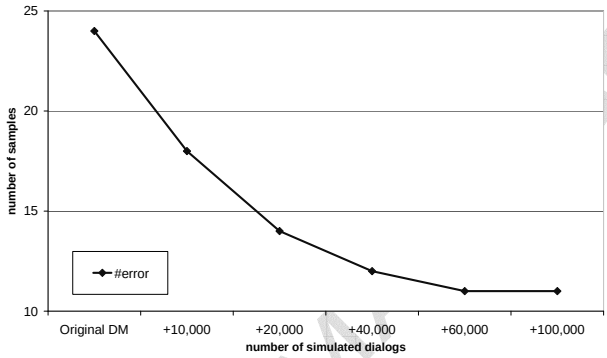


Fig. 7. Evolution of the number of erroneous system answers (*#error*) with regard to the incorporation of new simulated dialogs.

Figure 8 shows the evolution of *%strategy* and *%coherent*. It can be observed that the DM improved the generation of coherent answers when the new dialogs were incorporated. In addition, the number of coherent answers that are different from those defined in the WOz strategy increased. In other words, the original strategy was modified, thereby allowing the DM to tackle new situations and generate new coherent answers. Thus, the variability of the dialog model is increased by detecting new dialog situations that are not present in the initial model and new valid answers for the situations that were already contained in the initial corpus.

7.2 Evaluation with real users

Finally, we evaluated the behavior of our DM with real users using 15 scenarios consisting of the different queries defined for type S1 and S2 scenarios. A total of 150 dialogs were performed by six users using the complete dialog system presented in this paper. The threshold of the confidence measures used for the

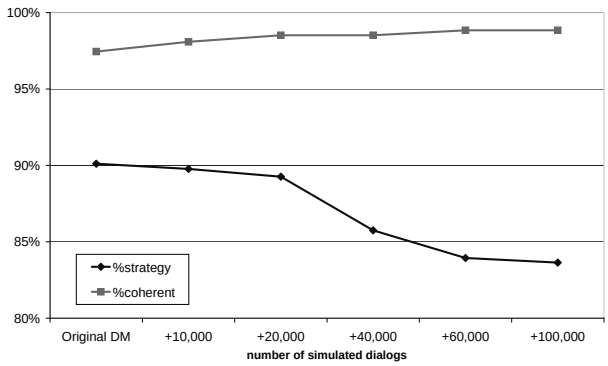


Fig. 8. Evolution of the *%strategy* and *%coherent* with regard to the incorporation of new simulated dialogs.

codification of the DR was 0.5. We considered the following measures for the evaluation:

(1) Dialog success rate (%success). This is the percentage of successfully completed tasks. In each scenario, the user has to obtain one or several items of information, and the dialog success depends on whether the system provides correct data (according to the aims of the scenario) or incorrect data to the user.
(2) Average number of turns per dialog (nT).
(3) Confirmation rate (%confirm). This was obtained by counting the explicit confirmation turns, nCT, per dialog system turn, that is, nCT/nT.
(4) Average number of corrected errors per dialog (nCE). This is the average of errors detected and corrected by the dialog manager. We have counted only those errors that modify the values of the attributes (and that could cause the failure of the dialog).
(5) Average number of uncorrected errors per dialog (nNCE). This is the average of errors not corrected by the dialog manager. As above, only errors that modify the values of the attributes are considered.
(6) Error correction rate (percentage of corrected errors, that is, nCE/ (nCE + nNCE).

The results presented in Table 9 show that in most cases the automatically learnt DM has the capability of correctly interacting with the user. The dialog success depends on whether the system provides the correct data for every objective defined in the scenario. All of the objectives defined in each scenario are achieved in 93.0% of the dialogs. The analysis of the main problems detected in the acquired dialogs shows that, in some cases, the system did not detect that the user wanted to finish the dialog. A second problem was related to the introduction of data in the DR with a high confidence value due to errors generated by the ASR that were not detected by the DM. However, the evaluation confirms a good operation of the approach since the information is correctly given to the user in the majority of cases.

Dialogs	%success	nT	%confirm	%correct	nCE	nNCE
150	93.0%	12.4	49%	0.97%	0.23	81

Table 9
Results of the evaluation of the statistical DM with real users

8 Conclusions

In this paper, we have presented a corpus-based methodology for the development of statistical dialog managers and the optimization of the dialog strategy. Our methodology for dialog management is based on the estimation of a statistical model from the sequences of the system and user dialog acts obtained from a set of training data. We have studied different proposals to tackle the problem of the size of the space of situations, and the problem of lack of coverage of the model.

Our approach is based on the use of a classification process to select the system answer. Another main characteristic consists of using a data structure that stores the information provided by the user regarding the task. This information, the last system answer, and the task-independent information of the last user turn are taken into account as input for the classification process. Thus, the complete history of the dialog is considered to determine the next system answer.

We have defined a codification of this information to facilitate the correct operation of the classification function. This representation allows the system to automatically generate a specialized answer that takes into account the current situation of the dialog. Several approaches have been evaluated for the definition of the classification function. The results of this evaluation have shown the good operation of a classifier based on neural networks.

The statistical methodology for dialog management has been extended to develop a statistical user simulator. Thus, the complete process of dialog is statistically modeled, from the determination of the user turn to the generation of the new system answer. The proposed methodology allows the generation of new dialogs with little effort.

We have described an evaluation of this methodology within the framework of a Spanish project called DIHANA. A complete dialog system for information access using spontaneous speech in a restricted domain task has been developed for this project. The main characteristic of this system is the definition of statistical methodologies to model the main modules that make up the dialog system. Using these approaches, we try to facilitate the adaptation of the different modules to new tasks.

Error detection and correction techniques have also been developed. These

techniques, which are based on the use of confidence scores and the definition of different kinds of confirmations, allow us to distinguish the situations in which errors appear and to make the necessary corrections to satisfactorily complete the task.

A set of 100,000 dialogs has been simulated by means of the interaction of the user simulator and the dialog manager. Successful dialogs have been incorporated to the training corpus for evaluating the evolution of the dialog model. By means of the user simulation technique, it has been possible to obtain a total of 26,000 successful dialogs. In addition, the simulated dialogs are generated automatically labeled. Therefore, the effort that would be required to manually acquire and label this high number of dialogs is greatly reduced. The results of the evaluation demonstrate that the coverage of the DM is increased by incorporating the successful simulated dialogs and that the number of unseen situations can be reduced. A study of the evolution of the strategy followed by the DM has also been carried out. This study shows how the DM modifies its strategy by detecting new correct answers that were not defined in the initial strategy. An evaluation of the DM with real users has been carried out to corroborate these results. This preliminary evaluation shows the correct operation of the learned DM since the users obtained all the information required in the objectives in 93% of the dialogs.

The methodology that we have developed permits an easy modelization of dialog management in slot-filling tasks, which are very common in dialog systems. For more difficult domains, a previous plan recognition phase would be necessary. Information regarding the task is centralized in our approach in the DR. Thus, the adaptation to new tasks consists of adapting the structure of this register to the requirements of the task and training the dialog model with the corresponding corpus. As future work, we want to apply this technique 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. Users can ask for the availability, the booking or cancellation of a facility, and the information about his/her current bookings. Using this approach, we want to acquire a corpus that makes the learning of a dialog manager possible for the domain of the EDECAN project. This dialog manager will be used in a supervised acquisition of a dialog corpus with real users.

Acknowledgements

This work has been partially funded by Spanish MEC and FEDER under project TIN2005-08660-C04-02, Spain.

References

Benedí, J., Lleida, E., Varona, A., Castro, M., Galiano, I., Justo, R., López, I., Miguel, A., 2006. Design and acquisition of a telephone spontaneous speech dialogue corpus in Spanish: DIHANA. In: Proc. of the 5th International Conference on Language Resources and Evaluation (LREC'06). Genoa (Italy), pp. 1636–1639.

Bishop, C. M., 1995. Neural networks for pattern recognition. Oxford University Press.

Bonafonte, A., Aibar, P., Castell, E., Lleida, E., Mariño, J., Sanchís, E., Torres, M. I., 2000. Desarrollo de un sistema de diálogo oral en dominios restringidos. In: Proc. of Primeras Jornadas en Tecnología del Habla. Sevilla (Spain).

Bos, J., Klein, E., Lemon, O., Oka, T., 2003. DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture. In: Proc. of the 4th SIGdial Workshop on Discourse and Dialogue. Sapporo (Japan), pp. 115–124.

Castro, M. J., Vilar, D., Sanchis, E., Aibar, P., 2003. Uniclass and Multiclass Connectionist Classification of Dialogue Acts. In: Proc. of the 8th Iberoamerican Congress on Pattern Recognition (CIARP'03). Vol. 2527 of LNCS. Springer-Verlag, pp. 664–673.

Cuayáhuitl, H., Renals, S., Lemon, O., Shimodaira, H., 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. In: Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'05). San Juan (Puerto Rico), pp. 290–295.

Cuayáhuitl, H., Renals, S., Lemon, O., Shimodaira, H., 2006. Learning Multi-Goal Dialogue Strategies Using Reinforcement Learning with Reduced State-Action Spaces. In: Proc. of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP). Pittsburgh (USA), pp. 469–472.

Doran, C., Aberdeen, J., Damianos, L., Hirschman, L., 2001. Comparing several aspects of human-computer and human-human dialogues. In: Proc. of the 2th SIGdial Workshop on Discourse and Dialogue. Aalborg (Denmark), pp. 1–10.

Dybkjaer, L., Bernsen, N., 2000. Usability issues in spoken language dialogue systems. In: Natural Language Engineering (2000). Cambridge University Press. Vol. 6. pp. 243–271.

Dybkjaer, L., Bernsen, N., Minker, W., 2004. Evaluation and usability of multimodal spoken language dialogue systems. In: Speech Communication. Vol. 43. pp. 33–54.

EAGLES, 1996. Evaluation of Natural Language Processing Systems. Tech. rep., EAGLES Document EAG-EWG-PR2. Center for Sprogteknologi, Copenhagen (Denmark).

Eckert, W., Levin, E., Pieraccini, R., 1997. User modeling for spoken dialogue system evaluation. In: Proc. of IEEE Automatic Speech Recognition and

Understanding Workshop (ASRU'97). Santa Barbara (USA), pp. 80–87.

Eckert, W., Levin, E., Pieraccini, R., 1998. Automatic evaluation of spoken dialogue systems. Tech. rep., TR98.9.1, ATT Labs Research.

Esteve, Y., Raymond, C., Bechet, F., Mori, R. D., 2003. Conceptual Decoding for Spoken Dialog systems. In: Proc. of European Conference on Speech Communications and Technology (Eurospeech'03). Vol. 1. Geneva (Switzerland), pp. 617–620.

Failenschmid, K., Williams, D., Dybkjaer, L., Bernsen, N., 1999. DISC Deliverable D3.6. Tech. rep., NISLab, University of Southern Denmark.

Fikes, R., Kehler, T., 1985. The role of frame-based representation in knowledge representation and reasoning. In: Communications of the ACM. Vol. 28. pp. 904–920.

Fraser, M., Gilbert, G., 1991. Simulating speech systems. In: Computer Speech and Language. Vol. 5. pp. 81–99.

García, F., Hurtado, L., Sanchis, E., Segarra, E., 2003. The incorporation of Confidence Measures to Language Understanding. In: International Conference on Text Speech and Dialogue (TSD'03). LNCS series 2807. Ceské Budejovice (Czech Republic), pp. 165–172.

Georgila, K., Henderson, J., Lemon, O., 2005. Learning user simulations for information state update dialogue systems. In: Proc. of the 9th European Conference on Speech Communication and Technology (Eurospeech'05). Lisbon (Portugal), pp. 893–896.

Georgila, K., Henderson, J., Lemon, O., 2006. User Simulation for Spoken Dialogue Systems: Learning and Evaluation. In: Proc. of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP). Pittsburgh (USA), pp. 1065–1068.

Griol, D., Hurtado, L. F., Segarra, E., Sanchis, E., 2006a. Managing unseen situations in a Stochastic Dialog Model. In: Proc. of AAAI Workshop Stadistical and Empirical Approaches for Spoken Dialogue Systems. Boston (USA), pp. 25–30.

Griol, D., Torres, F., Hurtado, L., Grau, S., García, F., Sanchis, E., Segarra, E., 2006b. A dialog system for the DIHANA project. In: Proc. of International Conference Speech and Computer (SPECOM'06). Saint Petersburg (Russia), pp. 131–136.

He, Y., Young, S., 2003. A data-driven spoken language understanding system. In: Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'03). St. Thomas (U.S. Virgin Islands), pp. 583–588.

Hurtado, L. F., Griol, D., Sanchis, E., Segarra, E., 2005. A stochastic approach to dialog management. In: Proc. of IEEE Workshop Automatic Speech Recognition and Understanding (ASRU'05). San Juan (Puerto Rico), pp. 226–231.

Hurtado, L. F., Griol, D., Segarra, E., Sanchis, E., 2006. A Stochastic Approach for Dialog Management based on Neural Networks. In: Proc. of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP). Pittsburgh (USA), pp. 49–52.

Jönsson, A., Dahlbick, N., 1988. Talking to A Computer is Not Like Talking To Your Best Friend. In: Proc. of the Scandinavian Conference on Artificial Intelligence (SCAI'88). Sapporo (Japan), pp. 53–68.

Lane, I., Ueno, S., Kawahara, T., 2004. Cooperative dialogue planning with user and situation models via example-based training. In: Proc. of Workshop on Man-Machine Symbiotic Systems. Kyoto (Japan), pp. 2837–2840.

Lemon, O., Georgila, K., Henderson, J., 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'06). Palm Beach (Aruba), pp. 178–181.

Lemon, O., Liu, X., 2007. Dialogue Policy Learning for Combinations of Noise and User Simulation: Transfer Results. In: Proc. of the 8th SIGdial Workshop on Discourse and Dialogue. Antwerp (Belgium), pp. 55–58.

Levin, E., Pieraccini, R., 1995. Concept-Based Spontaneous Speech Understanding System. In: Proc. of European Conference on Speech Communications and Technology (Eurospeech'95). Madrid (Spain), pp. 555–558.

Levin, E., Pieraccini, R., 1997. A stochastic model of human-machine interaction for learning dialog strategies. In: Proc. of European Conference on Speech Communications and Technology (Eurospeech'97). Rhodes (Greece), pp. 1883–1896.

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

Meng, H. H., Wai, C., Pieraccini, R., 2003. The Use of Belief Networks for Mixed-Initiative Dialog Modeling. In: IEEE Transactions on Speech and Audio Processing. Vol. 11(6). pp. 757–773.

Minker, W., Waibel, A., Mariani, J., 1999. Stochastically-Based Semantic Analysis. Kluwer Academic Publishers, Dordrecht (Holland).

Minsky, M., 1975. The Psychology of Computer Vision. McGraw-Hill, Ch. A Framework for Representing Knowledge, pp. 211–277.

Paek, T., Horvitz, E., 2000. Conversation as action under uncertainty. In: Proc. of the 16th Conference on Uncertainty in Artificial Intelligence. San Francisco (USA), pp. 455–464.

Pietquin, O., Beaufort, R., 2005. Comparing ASR modeling methods for spoken dialogue simulation and optimal strategy learning. In: Proc. of the 9th European Conference on Speech Communication and Technology (Eurospeech'05). Lisbon (Portugal), pp. 861–864.

Pietquin, O., Dutoit, T., 2005. A probabilistic framework for dialog simulation and optimal strategy learning. In: IEEE Transactions on Speech and Audio Processing, Special Issue on Data Mining of Speech, Audio and Dialog. Vol. 14. pp. 589–599.

Rieser, V., Lemon, O., 2006. Cluster-based User Simulations for Learning Dialogue Strategies. In: Proc. of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP). Pittsburgh (USA), pp. 1766–1769.

Roy, N., Pineau, J., Thrun, S., 2000. Spoken dialogue management using probabilistic reasoning. In: Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00). Hong Kong (China), pp. 93–100.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986. PDP: Computational models of cognition and perception, I. MIT Press, Ch. Learning internal representations by error propagation, pp. 319–362.

Schatzmann, J., Georgila, K., Young, S., 2005a. 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.

Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., Young, S., 2007a. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In: Proc. of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL). Rochester, NY (USA), pp. 149–152.

Schatzmann, J., Thomson, B., Young, S., 2007b. Statistical User Simulation with a Hidden Agenda. In: Proc. of the 8th SIGdial Workshop on Discourse and Dialogue. Antwerp (Belgium), pp. 273–282.

Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S., 2005b. Effects of the User Model on Simulation-based Learning of Dialogue Strategies. In: Proc. of IEEE Workshop Automatic Speech Recognition and Understanding (ASRU'05). San Juan (Puerto Rico), pp. 220–225.

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

Scheffler, K., Young, S., 1999. Simulation of human-machine dialogues. Tech. rep., CUED/F-INFENG/TR 355, Cambridge University Engineering Dept., Cambridge (UK).

Scheffler, K., Young, S., 2000. Probabilistic simulation of human-machine dialogues. In: Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00). Istanbul (Turkey), pp. 1217–1220.

Scheffler, K., Young, S., 2001a. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In: Proc. of Human Language Technology (HLT'02). San Diego (USA), pp. 12–18.

Scheffler, K., Young, S., 2001b. Corpus-based Dialogue Simulation for Automatic Strategy Learning and Evaluation. In: Proc. of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001). Workshop on Adaptation in Dialogue Systems. Pittsburgh (USA), pp. 64–70.

Segarra, E., et al., 2002. Extracting Semantic Information Through Automatic Learning Techniques. International Journal on Pattern Recognition and Artificial Intelligence 16 (3), 301–307.

Singh, S., Kearns, M., Litman, D., Walker, M., 1999. Reinforcement learning for spoken dialogue systems. In: Proc. of Neural Information Processing

Systems (NIPS'99). Denver (USA), pp. 956–962.

Torres, F., Sanchis, E., Segarra, E., 2003. Development of a stochastic dialog manager driven by semantics. In: Proc. of European Conference on Speech Communications and Technology (Eurospeech'03). Geneva (Switzerland), pp. 605–608.

Walker, M., Litman, D., Kamm, C., Abella, A., 1998. Evaluating spoken dialogue agents with PARADISE: Two case studies. In: Computer Speech and Language. Vol. 12. pp. 317–347.

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

Young, S., 2002. The Statistical Approach to the Design of Spoken Dialogue Systems. Tech. rep., CUED/F-INFENG/TR.433, Cambridge University Engineering Department, Cambridge (UK).

Young, S., Schatzmann, J., Weilhammer, K., Ye, H., 2007. The Hidden Information State Approach to Dialogue Management. In: Proc. of 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Vol. 4. Honolulu, Haway (USA), pp. 149–152.

Young, S., Williams, J., Schatzmann, J., Stuttle, M., Weilhammer, K., 2005. The Hidden Information State Approach to Dialogue Management. Tech. rep., Department of Engineering. University of Cambridge, Cambridge (UK).