A Stochastic Approach for Dialog Management based on Neural Networks

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

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València, E-46022 València, Spain
{lhurtado,dgriol,esegarra,esanchis}@dsic.upv.es

Abstract

In this article, we present an approach for the construction of a stochastic dialog manager, in which the system answer is selected by means of a classification procedure. In particular, we use neural networks for the implementation of this classification process, which takes into account the data supplied by the user and the last system turn. The stochastic model is automatically learnt from training data which are labeled in terms of dialog acts. An important characteristic of this approach is the introduction of a partition in the space of sequences of dialog acts in order to deal with the scarcity of available training data. This system has been developed in 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. An evaluation of this approach is also presented.

Index Terms: Spoken dialog systems, dialog management, stochastic models, unseen situations, MLP.

1. Introduction

Although there are models for dialog managers that are manually designed in the literature, over the last few years, approaches using stochastic models to represent the behavior of the dialog manager have also been developed [1] [2] [3]. The use of stochastic models that are automatically learnt from data has provided very interesting results in other tasks involved in a spoken dialog system. In particular, there have been several interesting contributions in language understanding [4] [5] [6] [7].

Recently, we have presented a stochastic approach for the construction of a dialog manager [8]. This approach is based mainly on the estimation of a stochastic model from the sequences of the system and user dialog acts obtained from a set of training data. In order to make the estimation of such a stochastic model from training data manageable, we propose the introduction of a partition in the space of all the possible sequences of dialog acts. This partition is defined taking into account the data supplied by the user throughout the dialog. The confidence measures provided by the recognition and the understanding modules are also taken into account in the definition of this partition of the space of sequences of dialog acts.

After the estimation process, given a user turn of a new dialog, this stochastic dialog manager must be able to assign a system answer. In the first version of our dialog manager [8], we assumed that if this user turn was already observed in the training corpus, the assigned system answer was the same as the corresponding answer observed in training. However, if this user turn was not observed in the training corpus, we applied a certain distance measure in order to assign it an observed event, and consequently, a system answer.

In this paper, we propose a different approach for the association of the system answer. Given a user turn, the stochastic dialog manager makes the assignation of a system answer according to the result of a classification process. This process is the same for both, observed and unobserved user turns. In this work, we have used neural networks to carry out this classification process.

Our Dialog Manager is integrated in a dialog system developed within the framework of the DIHANA project [9]. This project undertakes the design and development of a dialog system for access to an information system using spontaneous speech. The domain of the project is the query to an information system about railway timetables and prices in Spanish by telephone.

Sections 2 and 3 present a description of the corpus and its semantic and dialog-act labeling. Section 4 presents the proposed stochastic Dialog Manager. Section 5 describes the classification process. Sections 6 and 7 present an evaluation of this approach, and our conclusions.

2. DIHANA Corpus

A set of 900 dialogs was acquired in the DIHANA project. Three types of scenarios were defined: timetables for a one-way trip or a two-way trip, prices, and services. The number of users was 225 with 4 dialogs per user. The total number of user turns was 6280, and the vocabulary was 823 words.

Although this corpus was acquired using a Wizard of Oz technique (WO), real speech recognition and understanding modules were used. A strategy for the WO based on the confirmation of values with a low confidence was defined. Following this strategy, the WO interacts with the user on the basis of the information contained in a data structure that we call Dialog Register (DR). This structure incorporates all the information provided by the understanding module after each user turn, that is, concepts, attributes, and their confidence scores.

The WO strategy is as follows:

- Safe state. If all the data of the dialog register have a confidence score that is higher than the fixed threshold, the Wizard selects one of the following three interactions: a) implicit confirmation, query to the database, and answer to the user, if the dialog register contains all the necessary information; b) inquiry to the user if the dialog register does not store a value for the current concept and/or some of the minimum attributes; and c) mixed confirmation to give naturalness to the dialog, which includes the data to be confirmed and data that has a confidence score that is higher than the fixed threshold.

- Uncertain state. When one or more data of the dialog register have a confidence score that is lower than the fixed threshold, the Wizard selects one of the following two interactions: a) explicit confirmation of the first uncertain item that appears in the dialog register; and b) mixed confirmation to give naturalness to the dialog.

3. Corpus labeling

The representation of user and system turns is done in terms of dialog acts. In the case of user turns, the dialog acts correspond to the classical frame representation of the meaning of the utterance. In other words, 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. The Understanding Module takes the sentence supplied by the recognition process as input and generates one or more frames as output. In this task, we defined eight concepts (*Hour*, *Price*, *Train-Type*, *Trip-Time*, *Services*, *Affirmation*, *Negation*, and *Not-Understood*) and ten attributes (*Origin*, *Destination*, *Departure-Date*, *Arrival-Date*, *Departure-Hour*, *Arrival-Hour, Class*, *Train-Type*, *Order-Number*, and *Services*). An example of the semantic interpretation of an input sentence is shown below:

Input sentence:
Yes, I would like to know the timetables and the train types leaving from Barcelona.
Semantic interpretation:
(Affirmation)
(Hour)
Origin: Barcelona
(Train-Type)
Origin: Barcelona

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 involved in the turn and is specific to the task. 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 1.

4. The stochastic Dialog Manager

We have developed a Dialog Manager (DM) based on the stochastic modelization of the sequences of dialog acts (user and system dialog acts) [8]. We have obtained a Stochastic DM that can generate system turns based only on the information supplied by the user turns and the information contained in the model. A labeled corpus of dialogs is used to estimate the stochastic DM.

A formal description of the proposed stochastic model is:

Let A_i be the output of the dialog system (the system answer or the system turn) at time i, expressed in terms of dialog acts. Let

Do you want to know timetables? **(Confirmation:Departure-Hour:Nil)** Do you want train types to Valencia, from Barcelona? **(Confirmation:Train-Type:Destination)** **(Confirmation:Origin:Origin)** There is only one train, which is a Euromed, that leaves at 0:27 at night. Anything else? **(Answer:Departure-Hour:Departure-Hour,** **Number-Trains, Train-Type)(New-Query:Nil:Nil)**

Figure 1: Labeling examples from the DIHANA corpus

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 call this turn A_1. 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:

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

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. 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:

$$\begin{array}{ccccccc} S_1, & & \cdots, & & S_i, & \cdots, & S_n \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ DR_1 & & DR_2 & & DR_i & & DR_n \end{array}$$

where DR_1 captures the default information of the dialog manager (*Origin* and *Class*), and the following values DR_i are updated, considering 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 to be 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:

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

Each user turn supplies the system with information about the task; that is, the user 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 this reason, for the selection of the best system answer A_i, we take into account the DR that results from turn 1 to turn $i-2$, and we explicitly consider the last state S_{i-1}.

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 stochastic dialog model makes the assignation of a system answer according to the result of a classification process. During a new dialog, when a user turn is observed, it 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 to this selected class. This classification process is carried out using a multilayer perceptron (MLP) [10] 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 $c \in \mathcal{C}$.

4.1. Dialog Register representation

For the DIHANA task, the DR is a sequence of 15 fields, where each concept or attribute has a field associated to it. The sequence of fields for concepts is *Hour*, *Price*, *Train-Type*, *Trip-Time*, and *Services*. The sequence of fields for attributes is *Origin*, *Destination*, *Departure-Date*, *Arrival-Date*, *Departure-Hour*, *Arrival-Hour*, *Class*, *Train-Type*, *Order-Number*, and *Services*.

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 access to 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 not activated, or the value of the attribute is not given.
- **1**: The concept or attribute is activated with a confidence score that is higher than a given threshold (a value between 0 and 1). The confidence score is given during the recognition and understanding processes [11] 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.

Therefore, each DR can be represented as a 15-length string from $\{0, 1, 2\}^{15}$.

5. MLP classifier

Multilayer perceptrons (MLPs) are the most common artificial neural networks used for classification [12]. In order to apply a MLP to the search for the answer of the dialog manager (as we stated in Section 4), the input layer holds a codification of the input pair (DR_{i-1}, S_{i-1}), and the output layer is defined according to the number of possible system answers and represents the class $c \in \mathcal{C}$ in which the input is classified. The result of this classification gives the corresponding system answer A_i associated to that class.

The representation defined for the input pair (DR_{i-1}, S_{i-1}) 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).

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

- 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.

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

Given an input pair (DR_{i-1}, S_{i-1}), the MLP classifies it according to the following equation:

$$\hat{c} = \underset{c \in \mathcal{C}}{\operatorname{argmax}} P(c|x) \approx \underset{c \in \mathcal{C}}{\operatorname{argmax}} g_c(x, \omega) \quad (1)$$

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

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

The output layer of the MLP has the size of $|C|$ and represents the result of function $g_c(x, \omega)$, where c is the c-th output unit of the MLP with parameters ω, given the input sample x. This function approaches the a posterior probability $P(c|x)$.

6. Evaluation

The evaluation of the stochastic dialog model was carried out using a cross validation process. The corpus was randomly split into five subsets of 1232 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.

The number of different classes in the corpus (that is, the number of possible system answers) was 51. The average of different (DR, S) pairs in the training sets was 1126.

Software developed in our labs was used to model and train the MLPs. A validation subset (20%) was extracted from each training set. MLPs were trained using the backpropagation with momentum algorithm [10]. The topology used was two hidden layers with 110 units each one.

We defined four measures to evaluate the performance of the methodology. The first one is the percentage of answers that are equal to those generated by the WO ($\%exact$). The second one is the percentage of answers that follows the strategy defined for the acquisition of the DIHANA corpus ($\%strategy$). The third one is the percentage of answers that are coherent with the current state of the dialog ($\%correct$). Finally, the fourth one is the percentage of answers that are not compatible with the current state of the dialog ($\%error$). Table 1 shows the results of the evaluation.

	%exact	%strategy	%correct	%error
System answer	76.62%	97.34%	99.33%	0.42%

Table 1: DM evaluation results

Taking into account that the WO strategy presents several answer possibilities given a certain dialog state, the results that are relevant within the framework of dialog management are $\%strategy$ and $\%correct$. These results 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 97.34%. Moreover, the answer generated is exactly the one selected by the WO (%exact) in 76.62% of the cases.

Finally, the number of answers generated by the MLP that can cause the failure of the system is only 0.42%. An answer that is coherent with the current state of the dialog is generated in 99.33% of cases. These last two results also demonstrate the correct operation of the classification methodology.

7. Conclusions

In this paper, we have presented an approach for the development of stochastic Dialog Managers learnt from training samples. We have developed a detailed representation of the user and system dialog acts. This representation allows the system to automatically generate a specialized answer that takes into account the current situation of the dialog. From this representation, a classification methodology based on MLPs is used in order to generate the system answers. Some experiments have been performed to test the behavior of the system. The results show the satisfactory operation of the developed approach. As future work, an evaluation of the behavior of the system using real users is going to be made to compare the results with those presented in this paper.

8. Acknowledgements

Work partially supported by the Spanish CICYT under contract TIN2005-08660-C04-02

9. References

[1] Steve Young, "The Statistical Approach to the Design of Spoken Dialogue Systems," in *Technical Report CUED/F-INFENG/TR.433, Cambridge UK*, 2002, pp. 1–25.

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

[3] F. Torres, L.F. Hurtado, F. García, E. Sanchis, and E. Segarra, "Error handling in a stochastic dialog system through confidence measures," in *Speech Communication*, 2005, pp. (45):211–229.

[4] W. Minker, A. Waibel, and J. Mariani, *Stocastically-Based Semantic Analysis*, Kluwer Academic Publishers, Boston, 1999.

[5] E. Segarra, E. Sanchis, F. García, and L.F. Hurtado, "Extracting semantic information through automatic learning techniques," in *International Journal of Pattern Recognition and Artificial Intelligence*, Salt Lake City (USA), 2002, pp. 16(3):301–307.

[6] Yulan He and S. Young, "A data-driven spoken language understanding system," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'03)*, 2003, pp. 583–588.

[7] Y. Esteve, C. Raymond, F. Bechet, and R. De Mori, "Conceptual Decoding for Spoken Dialog systems," in *Proc. of Eurospeech*, 2003, vol. 1, pp. 617–620.

[8] Lluís Hurtado, David Griol, Emilio Sanchis, and Encarna Segarra, "A stochastic approach to dialog management," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'05)*, 2005, pp. 226–231.

[9] J.M. Benedí, A. Varona, and E. Lleida, "DIHANA: Sistema de diálogo para el acceso a la información en habla espontánea en diferentes entornos," in *Actas de las III Jornadas en Tecnología del Habla*, Valencia (España), 2004, pp. 141–146.

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

[11] F. García, L. Hurtado, E. Sanchis, and E. Segarra, "The incorporation of Confidence Measures to Language Understanding," in *International Conference on Text Speech and Dialogue (TSD 2003). Lecture Notes in Artificial Intelligence series 2807*, Ceské Budejovice (Czech Republic), 2003, pp. 165–172.

[12] M. J. Castro, D. Vilar, E. Sanchis, and P. Aibar, "Uniclass and Multiclass Connectionist Classification of Dialogue Acts," in *Proc. 8th Iberoamerican Congress on Pattern Recognition (CIARP'03)*, vol. 2527 of *LNCS*, pp. 664–673. Springer-Verlag, 2003.