Managing Unseen Situations in a Stochastic Dialog Model

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

Abstract

In this article, we present an approach for enriching a stochastic dialog manager to be able to manage unseen situations. As the model is estimated using a training corpus, the problem of augmenting the coverage of the model must be tackled. We modeled the problem of coverage as a classification problem, and we present several approaches for the definition of the classification function. 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. A corpus of 900 dialogs was acquired through the Wizard of Oz technique. An evaluation of these approaches is also presented.

Introduction

Nowadays, there are diverse projects that have developed dialog systems to provide information and other services automatically. In a dialog system of this kind, several modules cooperate to perform the interaction with the user: the Speech Recognizer, the Language Understanding Module, the Dialog Manager, the Answer Generator, and the Synthesizer. Each one has its own characteristics and the selection of the most convenient model varies depending on certain factors: the goal of each module, the possibility of manually defining the behavior of the module, or the capability of automatically obtaining models from training samples. The use of stochastic models that are automatically learnt from data has provided very interesting results. These stochastic models have been widely used, not only for speech recognition, but also for language understanding (Minker, Waibel, & Mariani 1999), (Levin & Pieraccini 1995) (He & Young 2003) (Segarra & others 2002) (Esteve, Raymond, & Bechet 2003).

Although, in the literature, there are models for dialog managers that are manually designed, over the last few years, approaches using stochastic models to represent the behavior of the dialog manager (Young 2002) (Levin, Pieraccini, & Eckert 2000) (Torres *et al.* 2005) have also been developed.

Recently, we have presented a stochastic approach for the construction of a dialog manager (Hurtado *et al.* 2005). Our dialog manager is based mainly 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 is to make the estimation of a stochastic model from training data manageable. 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.

Approaches of this kind 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 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 present new proposals for adapting the model to these unseen situations. We model the problem of coverage as a classification problem. The partitioned space of the possible sequences of dialog acts that is estimated in the model 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 is defined. During a new dialog, when a sequence that has been unseen in the training phase is observed, it is classified into a class of this set, and the answer of the system at that moment is the answer associated to this selected class. The classification function can be defined in several ways. In this paper, we present three different definitions of such a function: a multinomial naive Bayes classifier, n-gram based classifier, a classifier based on grammatical inference techniques. An approach that uses Support Vector Machines for the classification process can be found in (Denecke & Yasuda 2005).

Our Dialog Manager is integrated in a dialog system

developed within the framework of the DIHANA project (Benedí, Varona, & Lleida 2004). 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 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 stochastic Dialog Manager proposed. In Section 5, different definitions of the classification function are proposed. In Section 6, an evaluation of these classification functions is presented, and finally, our conclusions are presented.

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 characteristics of the acquired corpus are shown in Table 1.

Number of users	225
Number of dialogs/user	4
Number of user turns	6280
Average number of user turns/dialog	7
Average number of words/user turn	7.74
Vocabulary	823
Duration of the recording (hours)	10.8

Table 1: Main characteristics of the DIHANA corpus.

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. Although the WO heard the user, its strategy was based on the confirmation of those values to which the understanding module assigned a low confidence. 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 utterance, 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:
  - Implicit confirmation, query to the database, and answer to the user, if the dialog register contains all the necessary information.
  - Inquiry to the user if the dialog register does not store a value for the current concept and/or some of the minimum attributes.
  - 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:
  - Explicit confirmation of the first uncertain item that appears in the dialog register.
  - Mixed confirmation to give naturalness to the dialog.

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 corresponds 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 and ten attributes. The eight concepts are divided into two groups:

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

The attributes are: *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:
[SPANISH] Sí, me gustaría conocer los horarios y los precios desde Valencia.
[ENGLISH] *Yes, I would like to know the timetables and the prices leaving from Valencia.*

Semantic interpretation:
(Affirmation)
(Hour)
Origin: Valencia
(Price)
Origin: Valencia

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.

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) (Hurtado *et al.* 2005).

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 as follows:

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 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 that 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:

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

where DR_1 captures the default information of the dialog manager (*Origin* and *Class*), and the following values DR_i are updated, 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 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 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 $i-2$, and we explicitly consider the last state S_{i-1}.

The probabilities of the proposed model are obtained from a labeled training corpus through a maximum likelihood estimation.

Dialog Register representation

The DR is defined as a data structure that contains information about concepts and attributes provided by the user throughout the previous history of the dialog. This DR is a sequence of 15 fields, where each concept or attribute has a field associated to it. The sequence of fields of the DR is shown in Figure 2.

Concepts	Attributes
Hour	Origin
Price	Destination
Train-Type	Departure-Date
Trip-Time	Arrival-Date
Services	Departure-Hour
	Arrival-Hour
	Class
	Train-Type
	Order-Number
	Services

Figure 2: Fields in the DR

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

[SPANISH] Quiere horarios de trenes a Granada, desde Valencia?
[ENGLISH] Do you want timetables to Granada, from Valencia?
(Confirmation:Departure-Hour:Destination)(Confirmation:Origin:Origin)
[SPANISH] Desea salir el día 24 de marzo?.
[ENGLISH] Do you want to leave on March the 24th?.
(Confirmation:Date:Date)
[SPANISH] El único tren es un Euromed que sale a las 0 y 27 de la noche. Desea algo más?
[ENGLISH] 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

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 (García *et al.* 2003) 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 of elements from $\{0, 1, 2\}$.

Managing unseen situations

In order to achieve full coverage of the model, we must consider the unseen situations, that is, the dialog manager must generate an answer even for pairs (DR, S) that were unseen in the training phase. We modeled the problem of coverage as a classification problem.

As stated in the introduction, the partitioned space of the possible sequences of dialog acts that is estimated in the model 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 answers. After the training phase is finished, a set of classes $\mathcal{C}$ is defined. During a new dialog, when an unseen situation occurs, the DM will use a classification function in order to select an answer. In this work, we present three approaches for the definition of the classification function, which are based on the learning of a model for each class $c \in \mathcal{C}$. These approaches differ in the methodology used in the learning process.

Let (DR, S) be an unseen pair, and let x be a string that encodes this pair. The classification is done with the following maximization:

$$\begin{aligned} \hat{c} &= \operatorname*{argmax}_{c \in \mathcal{C}} P(c|x) \\ &= \operatorname*{argmax}_{c \in \mathcal{C}} \frac{P(c)P(x|c)}{P(x)} \\ &= \operatorname*{argmax}_{c \in \mathcal{C}} P(c)P(x|c) \end{aligned} \quad (1)$$

Multinomial naive Bayes classifier

The naive Bayes classifier has long been a core technique in information retrieval, text classification, and machine learning (McCallum 1999), (Juan & Ney 2002). In this work, we use the naive Bayes classifier in its multinomial event model. A set of labeled training samples is used to estimate the parameters of the model. The classification of new samples is carried out by the Bayes decision rule through a selection of the class that has produced the largest probability. The naive Bayes classifier assumes that all attributes are independent of each other given the context of the class.

The variable x that is shown in Equation 1 is composed by the following terms:

- The first two levels of the labelling of the last system answer (A_{i-1}): This information is modeled using a multinomial variable, which has as many bits as possible combinations of the values of these two levels (51).

$$\vec{x}_1 = \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ \vdots \\ x_{1,51} \end{pmatrix} \in \{0,1\}^{51}$$

- Dialog register (DR): The sequence of fields of the DR is shown in Figure 2. Fifteen characteristics can be observed (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 multinomial variable with three bits.

$$\vec{x}_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{pmatrix} \in \{0,1\}^3 \quad i = 2, ..., 16$$

- Task independent information (*Affirmation*, *Negation*, and *Not-Understood* dialog acts): These three dialog acts have

$$g(1\,2\,1\,0\,0\,1\,2\,0\,0\,0\,0\,0\,1\,0\,0) = 1\#1\,2\#2\,1\#3\,0\#4\,0\#5\,1\#6\,2\#7\,0\#8\,0\#9\,0\#10\,0\#11\,0\#12\,1\#13\,0\#14\,0\#15$$

Figure 3: Example of the MGGI renaming function g.

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 multinomial variables with three bits.

$$\vec{x}_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{pmatrix} \in \{0,1\}^3 \quad i = 17, ..., 19$$

Then, the variable x can be represented using the vector of characteristics:

$$\vec{x} = \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vdots \\ \vec{x}_{19} \end{pmatrix}$$

and its probability can be calculated using:

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

The probabilities of the multinomial variables are given by:

$$P(\vec{x}_1) = \prod_{d=1}^{50} p_{1,d}^{x_{1,d}}$$

$$i = 2, \cdots, 19 \quad P(\vec{x}_i) = \prod_{d=1}^{3} p_{i,d}^{x_{i,d}}$$

where the coefficients are calculated using:

$$p_{i,d} = \frac{N(x_{i,d} = 1)}{N}$$

where N is the number of samples for the class, and $N(x_{i,d} = 1)$ is the number of samples for the class with a value 1 in position d.

n-gram and MGGI classifier

For these classifiers, a finite-state automaton is estimated for each class, $c \in \mathcal{C}$ from its corresponding training samples. The variable x contains the same information as in the above section. In these classifiers, $P(c)$ is considered to be the same for all classes, and so it does not contribute to the maximization process. When an unseen situation appears in the dialog, the next system answer is selected by maximizing:

$$\hat{c} = \underset{c \in \mathcal{C}}{\operatorname{argmax}} P(x|c)$$

In this work, we defined three types of finite-state classifiers: bigram models, trigram models, and MGGI models. The MGGI (Morphic Generator Grammatical Inference) methodology (Segarra & Hurtado 1997) is a grammatical inference technique that allows us to obtain a variety of regular languages. It is based on the definition of a renaming function g; that is, each symbol of each input sample is specialized (renamed) following the renaming function. Different definitions of g will produce different models.

We defined the renaming function g in such a way that it specializes each field in the DR by adding information about its position. Figure 3 shows an example of the use of the renaming function g defined for this work. This renaming function is applied to the symbols from the DR representation but not to those from the S representation.

Evaluation

The evaluation of the different methodologies 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 classes in $\mathcal{C}$ (that is, the number of possible system answers) was 51. The average of different (DR, S) pairs in the training sets was 1126. The average of unseen situations was 14.6% of the test sets.

We defined three measures to evaluate the performance of the different methodologies. 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 ($\%correct$). Table 2 shows the results of the evaluation of the different classification functions.

	%exact	%strategy	%correct
multinomial classifier	56.3%	75.2%	88.5%
bigram classifier	25.7%	35.0%	45.7%
trigram classifier	28.7%	40.7%	51.2%
MGGI classifier	44.8%	66.1%	75.7%

Table 2: 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 $\%correct$. Table 2 shows that the multinomial classifier provides the best results. It also shows that among the finite-state model classifiers, the bigram and trigram classifiers are worse than the MGGI classifier, this is because they cannot capture long-term dependencies. The renaming function defined for the MGGI classifier seems to generate a model with too many states for the size of the training corpus, therefore, this classifier could be underestimated.

Conclusions

In this paper, we have presented an approach to the development of stochastic dialog managers. We have focused our interest on the coverage problem, which has been modeled in terms of a classification of the situations that are not represented in the model. Some experiments have been performed to test the behavior of the different definitions of the classification function. The results show the satisfactory operation of the approach based on the use of multinomial classifiers. As future work, we will explore other classification techniques, such as those based on Neural Networks. We will also explore other encoding schemes in order to achieve better results in the classification techniques presented in this paper.

Acknowledgements

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

References

Benedí, J.; Varona, A.; and Lleida, E. 2004. 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*, 141–146.

Denecke, M., and Yasuda, N. 2005. Does this answer your Question? Towards Dialogue Management for Restricted Domian Question Answering Systems. In *Proc. of 6th SIGdial, Workshop on Discurse and Dialogue*, volume 1, 65–76.

Esteve, Y.; Raymond, C.; and Bechet, F. De Mori, R. 2003. Conceptual Decoding for Spoken Dialog systems. In *Proc. of Eurospeech*, volume 1, 617–620.

García, F.; Hurtado, L.; Sanchis, E.; and Segarra, E. 2003. 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*, 165–172.

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

Hurtado, L.; Griol, D.; Sanchis, E.; and Segarra, E. 2005. A stochastic approach to dialog management. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'05)*, 226–231.

Juan, A., and Ney, H. 2002. Reversing and Smoothing the Multinomial Naive Bayes Text Classifier. In *Proc. of PRIS'02*, 200–212.

Levin, E., and Pieraccini, P. 1995. Concept-based spontaneous speech understanding system. In *Proc. of Eurospeech'95*, 555–558.

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

McCallum, A. K. 1999. Multi-Label Text Classification with a Mixture Model Trained by EM. In *Proc. of NIPS'99*.

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

Segarra, E., and Hurtado, L. 1997. Construction of Language Models using Morfic Generator Grammatical Inference MGGI Methodology. In *Proc. of Eurospeech*, 2695–2698.

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

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

Young, S. 2002. The Statistical Approach to the Design of Spoken Dialogue Systems. In *Technical Report CUED/F-INFENG/TR.433, Cambridge UK*, 1–25.