A Practical Application of Evolving Fuzzy-Rule-Based Classifiers for the Development of Spoken Dialog Systems*

David Griol, José Antonio Iglesias, Agapito Ledezma, and Araceli Sanchis

Computer Science Department
Carlos III University of Madrid
Leganés, Spain
{dgriol,jiglesia,ledezma,masm}@inf.uc3m.es

Abstract. In this paper we present a statistical approach based on evolving Fuzzy-rule-based (FRB) classifiers for the development of dialog managers for spoken dialog systems. The dialog managers developed by means of our proposal select the next system action by considering a set of dynamic fuzzy rules that are automatically obtained by means of the application of the FRB classification process. Our approach has the main advantage of taking into account the data supplied by the user throughout the complete dialog history without causing scalability problems, also considering confidence measures provided by the recognition and understanding modules. The use of EFS also allows to process streaming data on-line in real time, thus dynamically evolving the structure and operation of the dialog model based on the interaction of the dialog system with its users. We also describe the application of our proposal to develop a dialog system providing railway information.

Keywords: Spoken Dialog Systems, Evolving Classifiers, Spoken Human-Machine Interaction, Statistical Methodologies, Dialog Management.

1 Introduction

Spoken Dialog Systems (SDSs) are computer programs that receive speech as input and generate synthesized speech as output, engaging the user in a dialog that aims to be similar to that between humans [1]. These interfaces make technologies more usable, as they ease interaction, allow integration in different environments, and make technologies more accessible, especially for disabled and elderly people [2].

Usually, SDSs carry out five main tasks [1,3,4]: Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU), Dialog Management (DM), Natural Language Generation (NLG), and Text-To-Speech Synthesis (TTS).

* This work has been supported by the Spanish Government under i-Support (Intelligent Agent Based Driver Decision Support) Project (TRA2011-29454-C03-03).

L. Iliadis et al. (Eds.): AIAI 2014, IFIP AICT 436, pp. 307–316, 2014.

These tasks are typically implemented in different modules of the system's architecture.

Although dialog management is only a part of the development cycle of spoken dialog systems, it can be considered one of the most demanding tasks given that it decides the next action of the system and encapsulates the logic of the speech application [5]. The selection of a specific system action depends on multiple factors, such as the output of the speech recognizer (e.g., measures that define the reliability of the recognized information), the dialog interaction and previous dialog history (e.g., the number of repairs carried out so far), the application domain (e.g., guidelines for customer service), knowledge about the users, and the responses and status of external back-ends, devices, and data repositories. Given that the actions of the system directly impact users, the dialog manager is largely responsible for user satisfaction. This way, the design of an appropriate dialog management strategy is at the core of dialog system engineering.

As an attempt to reduce the time and effort required for system implementation and carry out rapid system prototyping, statistical approaches for dialog management are gaining increasing interest. These approaches enable automatic learning of dialog strategies, thus avoiding the time-consuming process that hand-crafted dialog design involves. Statistical models can be trained from real dialogs, modeling the variability in user behaviors. Although the construction and parameterization of these models depend on expert knowledge about the task to be carried out by the dialog system, the final objective is to develop systems that are more robust for real-world conditions, and that are easier to adapt to different users and tasks [3].

In this paper we present a statistical approach for the development of dialog managers, in which the next system prompt is selected by means of a novel approach for online classifying based on Evolving Fuzzy Systems (EFS) [6]. These systems have an evolving structure which is updated according to the input samples. A prototype for this kind of classifiers is a data sample that groups several samples. The classifier is initialized with the first data sample and then, each data sample is classified into one of the existing prototypes. Finally, based on the potential of the new data sample to become a prototype, form a new prototype, or replace an existing one. Unlike other statistical approaches, our approach has the advantage of taking into account the data supplied by the user throughout the complete dialog without causing scalability problems. Confidence measures provided by the recognition and the understanding modules are also taken into account in the selection of the next system response.

The most widespread methodology for machine-learning of dialog strategies consists of modeling human-computer interaction as an optimization problem using Markov Decision Processes (MDP) and reinforcement methods [7]. The main drawback of this approach is that the large state space of practical spoken dialog systems makes its direct representation intractable. As described in [8], Partially Observable MDPs (POMDPs) outperform MDP-based dialog strategies since they provide an explicit representation of uncertainty. This enables the dialog manager to avoid and recover from recognition errors by sharing and

shifting probability mass between multiple hypotheses of the current dialog state. Other interesting approaches for statistical dialog management are based on modeling the system by means of stochastic Finite-State Transducers [9], or Bayesian Networks [10].

The use of EFS allows us to cope with huge amounts of data, process streaming data on-line in real time, and evolve the structure of a dialog model that defines an activity based on the human-computer interaction. The dialog model is designed and treated as a changing model which constantly reflects the changes in the way the dialog system interacts with the users. Thus, it can start learning "from scratch" since the number of fuzzy rules do not need to be defined *a priori*. However, EFS can also be used to evolve an initial dialog model which has been previously defined taking into account expert knowledge about the specific interaction domain. In addition, the proposed solution addresses other important related requirements such as processing huge amounts of sequential data in an optimized way.

After this introduction, the remainder of the paper is organized as follows. Section 2 describes the proposed methodology for dialog management and its specific application for the *eclass0* classifier. This section also presents a practical application of our proposal to develop a spoken dialog system providing railway information and the results of a preliminary evaluation. Section 3 presents the conclusions and suggests some future work guidelines.

2 Our Proposed Methodology for Dialog Management

The process followed by a statistical dialog manager for the selection of the next system response is summarized in Figure 1 [11]. As this figure shows, the user has an internal state S_u corresponding to a goal that is trying to accomplish and the dialog state S_d represents the previous history of the dialog. Based on the user's goal prior to each turn, the user decides some communicative action (also called intention) A_u, expressed in terms of dialog acts and corresponding to an audio signal Y_u. Then, the speech recognition and language understanding modules take the audio signal Y_u and generate the pair $(\tilde{A}_u, C)$.

This pair consists of an estimate of the user's action A_u and a confidence score that provides an indication of the reliability of the recognition and semantic interpretation results. This pair is then passed to the dialog model, which is in an internal state S_m an decides what action A_m the dialog system should take. This action is also passed back to the dialog manager so that S_m may track both user and machine actions. The language generator and the text-to-speech synthesizer take A_m and generate an audio response Y_m. The user listens to Y_m and attempts to recover A_m. As a result of this process, users update their goal state S_u and their interpretation of the dialog history S_d. These steps are then repeated until the end of the dialog.

In our statistical approach for dialog management, we propose that, given a new user turn, the statistical dialog model makes the assignation of a system response A_m according to the result of a classification process. The classification

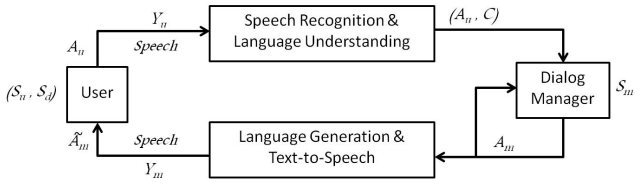

Fig. 1. Set of internal processes in a spoken dialog system

function can be defined in several ways. We propose the use of the *eClass* (evolving Classifier) family. During the training of these classifiers, a set of fuzzy rules that describes the most important observed features for the classification of each class (i.e., system prompt) is formed. These rules can also be constantly adjusted to the available training data. One of the advantages of *eClass* is that it does not require parameter optimization as its only parameter 'scale' can be directly inferred from the training data. This technique [12] is based on partitioning the data space into overlapping local regions through *Recursive Density Estimation* (RDE) and associating clusters (respectively fuzzy sets) to them.

The *eClass* family includes two different architectures and on-line learning methods:

- *eClass0* with the classifier consequents representing class label [12].
- *eClass1* for regression over the features using first order eTS fuzzy classifier.

In this paper, we focus on the *eClass0* classifier. *eClass0* possesses a zero-order Takagi-Sugeno consequent, so a fuzzy rule in the *eClass0* model has the following structure:

$$\begin{aligned} &Rule_i = IF(Feature_1 \ is \ P_1) \ AND \ldots \\ &\ldots AND \ (Feature_n \ is \ P_n) \\ &THEN \ Class \ = \ c_i \end{aligned} \tag{1}$$

where i represents the number of rule; n is the number of input features (observations corresponding to the different attributes and concepts defined for the semantic representation of the user's utterances); the vector $Feature$ stores the observed features, and the vector P stores the values of the features of one of the prototypes of the corresponding class $c_i \in \{$set of different classes$\}$. Each class is then associated to a specific system action (response).

The dialog manager must consider the concepts and values for the attributes provided by the user throughout the previous history of the dialog to select the next system action. For the dialog manager to take this decision, we have assumed that the exact values of the attributes are not significant. They are important for accessing databases and for constructing the output sentences of the system. However, the only information necessary to predict the next action by the system is the presence or absence of concepts and attributes. Therefore,

the codification we use for each concept and attribute provided by the SLU module is in terms of three values, $\{0, 1, 2\}$, 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;
- (2): The concept or attribute has a confidence score that is lower than the given threshold.

The *eClass0* model is composed of several fuzzy rules per class (the number of rules depends on the heterogeneity of the input data of the same class). During the training process, a set of rules is formed from scratch using an evolving clustering approach to decide when to create new rules. The inference in *eClass0* is produced using the "winner takes all" rule and the membership functions that describe the degree of association with a specific prototype are of Gaussian form.

The *potential* (Cauchy function of the sum of distances between a certain data sample and *all* other data samples in the feature space) is used in the partitioning algorithm. However, in these classifiers, the potential (P) is calculated recursively (which makes the algorithm faster and more efficient). The potential of the k^{th} data sample (x_k) is calculated by means of equation 2 [6]. The result of this function represents the *density* of the data that surrounds a certain data sample.

$$P(x_k) = \frac{1}{1 + \frac{\sum_{i=1}^{k-1} distance(x_k, x_i)}{k-1}} \tag{2}$$

where *distance* represents the distance between two samples in the data space.

The potential can be calculated using the euclidean or the cosine distance. In this case, cosine distance (*cosDist*) is used to measure the similarity between two samples; as it is described in equation 3.

$$cosDist(x_k, x_p) = 1 - \frac{\sum_{j=1}^{n} x_{kj} x_{pj}}{\sqrt{\sum_{j=1}^{n} x_{kj}^2 \sum_{j=1}^{n} x_{pj}^2}} \tag{3}$$

where x_k and x_p represent the two samples to measure its distance and n represents the number of different attributes in both samples.

Note that the resolution of equation 2 requires all the accumulated data sample available to be calculated, which contradicts to the requirement for real-time and on-line application needed in the proposed problem. For this reason, in [6] it is developed a recursive expression for the cosine distance. The proposed formula is as follows:

$$P_k(z_k) = \frac{1}{2 - \frac{1}{(k-1)\sqrt{\sum_{j=1}^{n}(z_k^j)^2}} B_k}; k = 2, 3...$$

$$where: \; B_k = \sum_{j=1}^{n} z_k^j b_k^j \; ; \; b_k^j = b_{(k-1)}^j + \sqrt{\frac{(z_k^j)^2}{\sum_{l=1}^{n}(z_k^l)^2}} \tag{4}$$

$$and \;\; b_1^j = \sqrt{\frac{(z_1^j)^2}{\sum_{l=1}^{n}(z_1^l)^2}} \; ; \; j = [1, n+1]; \; P_1(z_1) = 1$$

where z_k represents the k^{th} data sample (x_k) and its corresponding label ($z = [x, Label]$). Using this expression, it is only necessary to calculate *(n+1)* values where n is the number of different subsequences obtained; this value is represented by b, where $b_k^j, j = [1, n]$ represents the accumulated value for the k^{th} data sample.

In this case, a specific system action can be represented by several rules, depending on the heterogeneity of the samples that represent the same action. Thus, a class could be represented by one or several prototypes. The different prototypes that represent a system action are obtained from the input data and they are updated constantly. However, an initial rule-based model can be defined (if necessary) by hand as start point of the classifier. In this sense, new prototypes are created or existing prototypes are removed if necessary.

2.1 Practical Application: A Spoken Dialog System Providing Railway Information

We have applied our proposal to develop a mixed-initiative spoken dialog system to provide railway information system using spontaneous speech in Spanish. The system integrates the CMU Sphinx-II system speech recognition module[1]. As in many other conversational agents, the semantic representation chosen for dialog acts of the SLU module is based on the concept of frame [13]. This way, 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. For the 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 (*Timetables*, *Fares*, *Train-Type*, *Trip-Time*, and *Services*).
2. *Task-independent concepts*: they represent typical interactions in a dialog (*Acceptance*, *Rejection*, and *Not-Understood*).

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

A total of 51 system responses were defined for the task (classified into confirmations of concepts and attributes, questions to require data from the user, and answers obtained after a query to the database).

Using the previously described codification for the concepts and attributes, when a dialog starts (in the greeting turn) all the values are initialized to "0". The information provided by the users in each dialog turn is employed to update the previous values and obtain the current ones, as Figure 2 shows.

This figure shows the semantic interpretation and confidence scores (in brackets) for a user's utterance provided by the SLU module. In this case, the confidence score assigned to the attribute *Date* is very low. Thus, a "2" value is added in the corresponding position for this attribute. The concept (*Hour*) and

[1] `cmusphinx.sourceforge.net`

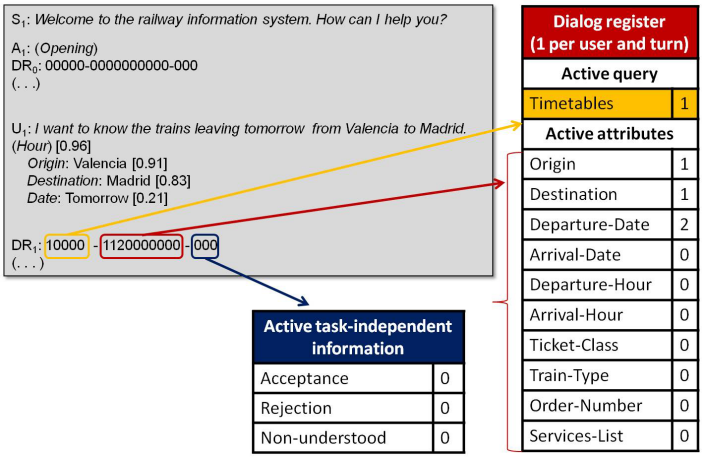

Fig. 2. Excerpt of a dialog with its correspondent representation of the task-dependent and active task-independent information for one of the dialog turns

the attribute *Destination* are recognized with a high confidence score, adding a "1" value in the corresponding positions.

The set of features for the classifier includes the codification of the different concepts and attributes that can be provided by the user and the task-independent information provided in the last user turn (none in this case). A total of 49 rules for the task were obtained with *eClass0*. Figure 3 shows the structure of these rules. Using them, the dialog manager would select the class $'SystemResponse23'$, which corresponds to a system confirmation of the departure date. This process is repeated to predict the next system response after each user turn.

An initial corpus of 900 dialogs (10.8 hours) was acquired for the task by means of the Wizard of Oz technique with 225 real users, for which an initial dialog strategy was defined by experts [4]. A set of 20 scenarios was used to carry out the acquisition. Each scenario defined one or two objectives to be completed by the user and the set of attributes that they must provide. The corpus consists of 6,280 user turns, with an average number of 7.7 words per turn. The corpus was split into a training subset of 4,928 samples (80% of the corpus) and a test subset of 1,232 samples (20% of the corpus).

We defined the following three measures to evaluate the statistical dialog model: i) *Matching*: percentage of responses provided by the system that match exactly the initial dialog strategy defined by the experts; ii) *Coherence*: percentage of responses that are coherent with the current dialog state but do not match the initial strategy; and iii) *Error*: percentage of responses that could cause a dialog failure.

We have test the behavior of our proposal comparing it with different definitions of the classification function used to determine the next system response.

$FRB - RailwayTask(eClass0)$:

IF ($Timetables$ is 1) AND ($Fares$ is 0) $AND \cdots AND$ ($Not - Understood$ is 0)
$THEN\ Class =' Ask - Date'$

IF ($Timetables$ is 2) AND ($Fares$ is 0) $AND \cdots AND$ ($Not - Understood$ is 1)
$THEN\ Class =' Confirm - Timetables'$

IF ($Timetables$ is 0) AND ($Fares$ is 1) $AND \cdots AND$ ($Not - Understood$ is 0)
$THEN\ Class =' Provide - Fares'$

...
IF ($Timetables$ is 1) AND ($Fares$ is 2) $AND \cdots AND$ ($Not - Understood$ is 1)
$THEN\ Class =' Close - Dialog'$

Fig. 3. Set of rules for the dialog manager obtained with *eClass0* for the railway task

In this work, we have used three approaches for the definition of the classification function: a multilayer perceptron (MLP), a multinomial naive Bayes classifier, and finite-state classifiers. We also defined three types of finite-state classifiers: bigram models, trigram models, and Morphic Generator Grammatical Inference (MGGI) models [14].

Table 1 shows the results obtained. As it can be observed, the Fuzzy-rule-based classifier provides satisfactory results in terms of the percentage of correct responses selected (*Matching* and *Coherence* measures) and responses that could cause the failure of the dialog (*Error* measure). With regard the rest of classifiers, the MLP classifier is the one providing the closest results to our proposal. The table 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.

Table 1. Results of the evaluation of the different classification functions

Dialog manager	*Matching*	*Coherence*	*Error*
Fuzzy-rule-based (FRB) classifier	76.7%	89.2%	5.6%
MLP classifier	76.8%	88.8%	5.8%
Multinomial classifier	63.4%	76.7%	10.6%
Bigram classifier	28.8%	37.3%	42.2%
Trigram classifier	31.7%	42.1%	44.1%
MGGI classifier	46.6%	67.2%	24.8%

Secondly, we have evaluated our proposal with the acquisition of 100 dialogs by means of a user simulator [4]. We considered the following measures: i) Dialog success rate (*Success*); ii) Average number of turns per dialog (nT); iii) Confirmation rate (*Confirmation*); and iv) Error correction rate (ECR). The confirmation rate was computed as the ratio between the number of explicit confirmation turns and the total number of turns in the dialog. The ECR was computed as the number of errors detected and corrected by the dialog manager divided by the total number of errors.

The results presented in Table 2 show that in most cases the automatically learned dialog model has the capability of correctly interacting with the user. The dialog success depends on whether the system provides the correct data for every objective user's query. All of the objectives defined are achieved in 93.5% of the dialogs. The analysis of the main problem detected in the acquired dialogs shows that, in some cases, the system did not detect the introduction of data with a high confidence value due to errors generated by the ASR that were not detected by the dialog manager. However, the evaluation confirms a good operation of the approach since the information is correctly given to the user in the majority of cases. The confirmation and error correction rates have also a remarkable impact on the described system performance.

Table 2. Results of the objective evaluation with real users

	Success	nT	*Confirmation*	ECR
Fuzzy-rule-based (FRB) dialog manager	93.5%	13.8	22%	0.87%

3 Conclusions and Future Work

In this paper, we have presented a statistical methodology for the development of dialog managers and the optimization of dialog strategies in spoken dialog systems. The selection of the following system response is based on a classification process that takes into account the history of the dialog. The most important contribution of our work consists of the use of Evolving Fuzzy Systems (EFS) to complete this classification. As a result of the application of our proposal, the dialog model is modeled by a set of automatically obtained dynamic rules that are applied to select the next system response. The use of EFS allows us to cope with huge amounts of data, and process streaming data on-line in real time.

We have described a practical application of our proposal to develop and evaluate a spoken dialog system providing railway information. A codification of the information sources has been proposed to facilitate the correct operation of the *eClass0* classification function. This representation allows the system to automatically generate a specialized answer that takes the current situation of the dialog into account. Task-dependent information is isolated from the model taking into account whether the user has provided a given piece of information related to the task and also the confidence scores assigned by the ASR and NLU

modules.This allows not only to cope with the situations observed the training corpus, but also to manage unseen situations by selecting the most convenient system action. The results of the evaluation shows the correct operation of the learned dialog manager with regard other definitions of the classification functions.

Future work will be oriented to deploy and evaluate our proposal in additional domains. As the dialog model is learned from a corpus of training samples, the performance of the dialog manager depends on the quality and size of the corpus used to learn the model. For this reason, we also want to evaluate the influence of the main features of the training corpus in the quality of the dialog model obtained by means of our proposal.

References

1. Pieraccini, R.: The Voice in the Machine: Building Computers that Understand Speech. The MIT Press (2012)
2. Vipperla, R., Wolters, M., Renals, S.: Spoken dialogue interfaces for older people. In: Advances in Home Care Technologies, pp. 118–137. IOS Press (2012)
3. Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S.: A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. Knowledge Engineering Review 21(2), 97–126 (2006)
4. Griol, D., Hurtado, L., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. Speech Communication 50(8-9), 666–682 (2008)
5. Wilks, Y., Catizone, R., Worgan, S., Turunen, M.: Some background on dialogue management and conversational speech for dialogue systems. Computer Speech and Language 25, 128–139 (2011)
6. Angelov, P., Zhou, X.: Evolving fuzzy-rule-based classifiers from data streams. IEEE T. Fuzzy Systems 16(6), 1462–1475 (2008)
7. Levin, E., Pieraccini, R., Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. IEEE T. Speech and Audio Processing 8(1), 11–23 (2000)
8. Young, S., Schatzmann, J., Weilhammer, K., Ye, H.: The Hidden Information State Approach to Dialogue Management. In: Proc. ICASSP 2007, pp. 149–152 (2007)
9. Planells, J., Hurtado, L., Sanchis, E., Segarra, E.: An Online Generated Transducer to Increase Dialog Manager Coverage. In: Proc. Interspeech 2012 (2012)
10. Meng, H.H., Wai, C., Pieraccini, R.: The Use of Belief Networks for Mixed-Initiative Dialog Modeling. IEEE T. Speech and Audio Processing 11(6), 757–773 (2003)
11. Young, S.: The Statistical Approach to the Design of Spoken Dialogue Systems. Technical report, Cambridge University Engineering Department (2002)
12. Angelov, P., Filev, D.: An approach to online identification of Takagi-Sugeno fuzzy models. IEEE T. Systems, Man and Cybernetics 34(1), 484–498 (2004)
13. Minsky, M.: A Framework for Representing Knowledge. In: The Psychology of Computer Vision, pp. 211–277. McGraw-Hill (1975)
14. Segarra, E., Hurtado, L.: Construction of Language Models using Morfic Generator Grammatical Inference MGGI Methodology. In: Proc. Eurospeech 1997, pp. 2695–2698 (1997)