Statistical Dialog Management Methodologies for Real Applications

David Griol
Dept. of Computer Science
Carlos III University of Madrid
Av. Universidad, 30, 28911, Leganés
dgriol@inf.uc3m.es

Zoraida Callejas, Ramón López-Cózar
Dept. of Languages and Computer Systems, CITIC-UGR
University of Granada
C/ Pdta. Daniel Saucedo Aranda, 18071, Granada
{zoraida, rlopezc}@ugr.es

Abstract

In this paper we present a proposal for the development of dialog systems that, on the one hand, takes into account the benefits of using standards like VoiceXML, whilst on the other, includes a statistical dialog module to avoid the effort of manually defining the dialog strategy. This module is trained using a labeled dialog corpus, and selects the next system response considering a classification process that takes into account the dialog history. Thus, system developers only need to define a set of VoiceXML files, each including a system prompt and the associated grammar to recognize the users responses to the prompt. We have applied this technique to develop a dialog system in VoiceXML that provides railway information in Spanish.

1 Introduction

When designing a spoken dialog system, developers need to specify the system actions in response to user utterances and environmental states that, for example, can be based on observed or inferred events or beliefs. In addition, the dialog manager needs a dialog strategy that defines the conversational behavior of the system. This is the fundamental task of dialog management (Paek and Pieraccini, 2008), as the performance of the system is highly dependent on the quality of this strategy. Thus, a great effort is employed to empirically design dialog strategies for commercial systems. In fact, the design of a good strategy is far from being a trivial task since there is no clear definition of what constitutes a good strategy (Schatzmann et al., 2006). Once the strategy has been designed, the implementation of the system is leveraged by programming languages such as VoiceXML, for which different programming environments and tools have been created to help developers.

As an alternative of the previously described rule-based approaches, the application of statistical approaches to dialog management makes it possible to consider a wider space of dialog strategies (Georgila et al., 2006; Williams and Young, 2007; Griol et al., 2009). The main reason is that statistical models can be trained from real dialogs, modeling the variability in user behaviors. The final objective is to develop dialog systems that have a more robust behavior and are easier to adapt to different user profiles or tasks.

(Pieraccini et al., 2009) highlights the impracticality of applying statistical learning approaches to develop commercial applications, in the sense that it is difficult to consider the expert knowledge of human designers. From his perspective, a hybrid approach, combining statistical and rule-based approaches, could be a good solution. The reason is that statistical approaches can offer a wider range of alternatives at each dialog state, whereas rule based approaches may offer knowledge on best practices.

For example, (Williams, 2008) proposes taking advantage of POMDPs and rule-based approaches by using POMDPs to foster robustness and at the same time being able to incorporate handcrafted constraints which cover expert knowledge in the application domain. Also (Lee et al., 2010) have recently proposed a different hybrid approach to dialog modeling in which n-best recognition hypotheses are weighted using a mixture of expert knowledge and data-driven measures by using an agenda and an example-based machine translation approach respectively. In both approaches, the hybrid method achieved significant improvements.

Additionally, speech recognition grammars for commercial systems have been usually built on the basis of handcrafted rules that are tested recursively, which in complex applications is very costly (McTear, 2004). However, as stated by (Pieraccini et al., 2009), many sophisticated com-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 269–272,
The University of Tokyo, September 24-25, 2010. ©2010 Association for Computational Linguistics

mercial systems already available receive a large volume of interactions. Therefore, industry is becoming more interested in substituting rule based grammars with statistical approaches based on the large amounts of data available.

As an attempt to improve the current technology, we propose to merge statistical approaches with VoiceXML. Our goal is to combine the flexibility of statistical dialog management with the facilities that VoiceXML offers, which would help to introduce statistical approaches for the development of commercial (and not strictly academic) dialog systems. To this end, our technique employs a statistical dialog manager that takes into account the history of the dialog up to the current dialog state in order to decide the next system prompt. In addition, the system prompts and the grammars for ASR are implemented in VoiceXML-compliant formats, for example, JSGF or SRGS. As it is often difficult to find or gather a human-machine corpus which cover an identical domain as the system which is to be implemented, our approach is also based on the compilation of corpora of interactions of simulated users, which is a common practice when using machine learning approaches for system development.

In contrast with other hybrid approaches, our main aim is not to incorporate knowledge about best strategies in statistical dialog management, but rather to take advantage of an implementation language which has been traditionally used to build rule-based systems (such as VoiceXML), for the development of statistical dialog strategies. Expert knowledge about deployment of VoiceXML applications, development environments and tools can still be exploited using our technique. The only change is in the transition between states, which is carried out on a data-driven basis (i.e., is not deterministic). We have applied our technique to develop a dialog system that provides railway information, for which we have developed a statistical dialog management technique in a previous study.

2 Our Proposal to Introduce Statistical Methodologies in Commercial Applications

As stated in the introduction, our approach to integrate statistical methodologies in commercial applications is based on the automatic learning of the dialog strategy using a statistical dialog management methodology. In most dialog systems, the dialog manager makes decisions based only on the information provided by the user in the previous turns and its own dialog model. For example, this is the case with most dialog systems for slot-filling tasks. The methodology that we propose for the selection of the next system response for this kind of task is detailed in (Griol et al., 2008). It is based on the definition of a data structure that we call Dialog Register (DR), which contains the information provided by the user throughout the dialog history. In brief, it is as follows: for each time i, the selection of the next system prompt A_i is carried out by means of the following maximization:

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

where the set $\mathcal{A}$ contains all the possible system responses and S_{i-1} is the state of the dialog sequence (*system-turn, user-turn*) at time i.

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}. Hence, for the selection of the best system response 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. user experience or preferences).

The selection of the system response is carried out through a classification process, for which a multilayer perceptron (MLP) is used. The input layer receives the codification of the pair (DR_{i-1}, S_{i-1}). The output generated by the MLP can be seen as the probability of selecting each of the different system answers defined for a specific task.

To learn the dialog model we use dialog simulation techniques. Our approach for acquiring a dialog corpus is based on the interaction of a user simulator and a dialog manager simulator (Griol et al., 2007). The user simulation replaces the user intention level, that is, it provides concepts and attributes that represent the intention of the user. This way, the user simulator carries out the functions of the ASR and NLU modules. Errors and confidence scores are simulated by a specific module in the simulator. The acquired dialogs are employed to automatically generate VoiceXML code for each system prompt and create the grammar needed to recognize the possible user utterances after each one of the system prompts.

3 Development of a railway information system using the proposed technique

To test our proposal, we have used the definitions taken to develop the DIHANA dialog system, which was developed in a previous study to provide information about train services, schedules and fares in Spanish (Griol et al., 2009; Griol et al., 2008). The DR defined for the our railway information system 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*). The system generates a total of 51 different prompts.

Three levels of labeling are defined for the labeling of the system dialog acts. The first level describes general acts which are task independent. The second level is used to represent concepts and attributes involved in dialog turns that are task-dependent. The third level represents values of attributes given in the turns. The following labels are 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*. There are dialog turns which are labeled with several dialog acts.

Having this kind of 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.

Two million dialogs were simulated using a set of two types of scenarios. Type S1 defines one objective for the dialog, whereas Type S2 defines two. Table 1 summarizes the statistics of the acquisition for the two types of scenarios.

[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] 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*)

Figure 1: Labeling examples of system turns from the DIHANA corpus

	Type S1	**Type S2**
Simulated dialogs	10^6	10^6
Successful dialogs	15,383	1,010
Different dialogs	14,921	998
Number of user turns per dialog	4.9	6.2

Table 1: Statistics of the new corpus acquisition

The 51 different system prompts have been automatically generated in VoiceXML using the proposed technique. For example, Figure 2 shows the VXML document to prompt the user for the origin city, whereas Figure 3 shows the obtained grammar for ASR.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
    xmlns:xsi="http://www.w3.org/2001/
    XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2001/vxml
    http://www.w3.org/TR/voicexml20/vxml.xsd"
    version="2.0" application="app-dihana.vxml">
 <form id="origin_form">
  <field name="origin">
     <grammar type="application/srgs+xml"
          src="/grammars/origin.grxml"/>
     <prompt>Tell me the origin city.</prompt>
     <filled>
       <return namelist="origin"/>
     </filled>
  </field>
 </form>
</vxml>
```

Figure 2: VXML document to require the origin city

4 Conclusions

In this paper, we have described a technique for developing dialog systems using a well known

```
#JSGF V1.0;
grammar origin;
public <origin> = [<desire>]
[<travel> <city> {this.destination=$city}]
[<proceed> <city> {this.origin=$city}];
<desire> = I want [to know] | I would like
[to know] | I would like | I want | I need
| I have to;
<travel> = go to | travel to | to go to
| to travel to;
<city> = Jaén | Córdoba | Sevilla | Huelva |
Cádiz | Málaga | Granada | Almería |
Valencia | Alicante | Castellón | Barcelona
| Madrid;
<proceed> = from | going from | go from;
```

Figure 3: Grammar defined to capture the origin city

standard like VoiceXML, and considering a statistical dialog model that is automatically learnt from a dialog corpus.

The main objective of our work is to reduce the gap between academic and commercial systems by reducing the effort required to define optimal dialog strategies and implement the system. Our proposal works on the benefits of statistical methods for dialog management and VoiceXML, respectively. The former provide an efficient means to exploring a wider range of dialog strategies, whereas the latter makes it possible to benefit from the advantages of using the different tools and platforms that are already available to simplify system development. We have applied our technique to develop a dialog system that provides railway information, and have shown that it enables creating automatically VoiceXML documents to prompt the user for data, as well as the necessary grammars for ASR. As a future work, we plan to study ways for adapting the proposed dialog management technique to more complex domains.

Additionally, we are interested in investigating possible ways for easing the adoption of our technique in industry, and the main challenges that might arise in using it to develop commercial systems.

Acknowledgments

This research has been funded by the Spanish Ministry of Science and Technology, under project HADA TIN2007-64718.

References

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

D. Griol, L.F. Hurtado, E. Sanchis, and E. Segarra. 2007. Acquiring and Evaluating a Dialog Corpus through a Dialog Simulation Technique. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 39–42, Antwerp (Belgium).

D. Griol, L.F. Hurtado, E. Segarra, and E. Sanchis. 2008. A Statistical Approach to Spoken Dialog Systems Design and Evaluation. *Speech Communication*, 50(8–9):666–682.

D. Griol, G. Riccardi, and Emilio Sanchis. 2009. A Statistical Dialog Manager for the LUNA project. In *Proc. of Interspeech/ICSLP'09*, pages 272–275, Brighton (UK).

Cheongjae Lee, Sangkeun Jung, Kyungduk Kim, and Gary Geunbae Lee. 2010. Hybrid approach to robust dialog management using agenda and dialog examples. *Computer Speech and Language*, 24(4):609–631.

Michael F. McTear, 2004. *Spoken Dialogue Technology: Towards the Conversational User Interface*. Springer.

T. Paek and R. Pieraccini. 2008. Automating spoken dialogue management design using machine learning: An industry perspective . *Speech Communication*, 50(8–9):716–729.

Roberto Pieraccini, David Suendermann, Krishna Dayanidhi, and Jackson Liscombe. 2009. Are We There Yet? Research in Commercial Spoken Dialog Systems. *Lecture Notes in Computer Science*, 5729:3–13.

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

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

Jason D. Williams. 2008. The best of both worlds: Unifying conventional dialog systems and POMDPs. In *Proceedings of the International Conference on Spoken Language Processing*.