A proposal to manage multi-task dialogs in conversational interfaces

David Griol, José Manuel Molina
Computer Science Department, Carlos III University of Madrid, Avda. de la Universidad, 30. Leganés (Spain), 28911
{david.griol, josemanuel.molina}@uc3m.es

KEYWORD

Human-machine interaction; Conversational Interfaces; Dialog Management.

ABSTRACT

The emergence of smart devices and recent advances in spoken language technology are currently extending the use of conversational interfaces and spoken interaction to perform many tasks. The dialog management task of a conversational interface consists of selecting the next system response considering the user's actions, the dialog history, and the results of accessing the data repositories. In this paper we describe a dialog management technique adapted to multi-task conversational systems. In our proposal, specialized dialog models are used to deal with each specific subtask of dialog objective for which the dialog system has been designed. The practical application of the proposed technique to develop a dialog system acting as a customer support service shows that the use of these specialized dialog models increases the quality and number of successful interactions with the system in comparison with developing a single dialog model.

1. Introduction

Conversational interfaces are computer programs that engage the user in a dialog that aims to be similar to that between humans (McTear et al., 2016; Lee et al., 2015; Ota and Kimura, 2014; Pieraccini, 2012). Different processes must be completed to achieve this complex objective, such as speech and feedback recognition, natural language processing, dialog management or speech synthesis.

The dialog management process of a dialog system relies on the fundamental task of deciding the next action of the system, interpreting the incoming semantic representation of the user input in the context of the dialog. In addition, it resolves ellipsis and anaphora, evaluates the relevance and completeness of user requests, identifies and recovers from recognition and understanding errors, retrieves information from data repositories, and decides about the next system's response.

The design of the dialog manager has been traditionally carried out by hand-crafting dialog strategies tightly coupled to the application domain in order to optimize the behavior of the dialog system in that context. This has motivated the research community to find ways to develop dialog management strategies that have a more robust behavior, better portability, generalizable and are easier to adapt to different user profiles or tasks (Griol et al., 2014).

In this paper we describe a methodology for dialog management, in which specialized dialog models are employed to deal with each one of the dialog domains and/or dialog subtasks that form the complete set of functionalities provided by the dialog system. The statistical dialog model learned for each specialized task is based on a classification process that provides the probabilities of selecting each one of the system actions (i.e., system responses) for the current state of the dialog. To do this, the training data is divided into different subsets, each covering a specific dialog objective or subtask. Each specific dialog model is trained using the corresponding training subset and it is selected during the dialog once the specific dialog subtask has been detected.

We have applied the proposed methodology to develop two versions of a practical dialog system acting as a customer support service to help solving simple and routine software/hardware repairing problems, both at the domestic and professional levels. The first dialog system uses a single dialog model to complete the dialog management task and the second one integrates our proposal with specialized dialog models learned for each dialog objective. An in-depth comparative assessment of the developed dialog systems has been completed with recruited users. The results of the evaluation show that the use of the specific dialog models system allows a better selection of the next system responses, thus increasing the number and quality of successful interactions with the dialog system.

The rest of the paper is organized as follows. Section 2 describes the main techniques and proposals defined for dialog management in conversational interfaces. Section 3 describes our proposal to develop a dialog manager for multi-task dialog systems. Section 4 shows the practical implementation of our proposal to develop the two systems for the customer support service. In Section 5 we discuss the evaluation results obtained by comparing the two developed systems. Finally, in Section 6 we present the conclusions and outline guidelines for future work.

2. Related work

As described in the previous section, the main objective of the dialog management task in a conversational interface is to decide the next system response. The simplest dialog management strategy is programmatic dialog management, in which a generic program implements the application with an interaction model based on finite-state machines (Barnard et al., 1999).

Frame-based dialog managers do not have a predefined dialog path but use a frame structure comprised of one slot per piece of information that the system can gather from the user (McTear, 2004). The core idea is that humans communicate to achieve goals and during the interaction the mental state of the speakers may change. This strategy is suitable for form-filling tasks in which the system asks the user a series of questions to gather information, and then consults an external knowledge source, such as the ones that can be developed with VoiceXML.

A related approach is the so-called “information state” dialog theory (Traum and Larsson, 2003). The information state of a dialog represents the information needed to uniquely distinguish it from all others. It comprises the accumulated user interventions and previous dialog actions on which the next system response can be based. The information state is also sometimes known as the conversation store, discourse context, or mental state. Following this theory, the main tasks of the dialog manager are to update the information state based on the observed user actions, and select the next system action.

In the agent-based paradigm for dialog management, dialog is viewed as interaction between two agents, each of which is capable of reasoning about its own actions and beliefs. Agent-based dialog management is suitable for the design of more natural dialog systems in which the system and the user can share the initiative of the dialog (Knott and Vlugter, 2008). Thus, the dialog manager takes the preceding context into account and the dialog evolves dynamically as a sequence of related steps that build on top of each other. Automating the learning of agent-based dialog managers by using statistical models trained with real conversations also allows us to model the variability in user behaviors and explore a wider range of strategies and dialog movements, also reducing the time and effort required to develop the dialog manager (Ferreira and Lefevre, 2015).

Statistical approaches for dialog management present several important advantages. Rather than maintaining a single hypothesis for the dialog state, they maintain a distribution over many hypotheses for the correct dialog state. Statistical dialog models can be trained with corpora of human-computer dialogs with the goal of explicitly modeling the variance in user behavior that can be difficult to address by means of hand-written rules (Schatzmann et al., 2006).

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 (Levin et al., 2000). The main drawback of this approach is that the large state space of practical spoken dialog systems makes its direct representation intractable (Young et al., 2007). Partially Observable MDPs (POMDPs) outperform MDP-based dialog strategies since they provide an explicit representation of uncertainty (Roy et al., 2000). 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 Hidden Markov Models (Cuayáhuitl et al., 2005), stochastic Finite-State Transducers (Planells et al., 2012), or using Bayesian Networks (Meng et al., 2003). Also (Lee et al., 2010) 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.

3. Our proposal for multi-task dialog management

Figure 1 shows the complete architecture of a dialog system integrating our proposal for the use of specialized dialog models to manage the dialog. As can be observed, the architecture consists of a set components, linked together by communication channels, that can be classified into System Components, Input and Output Components, and Natural Language Processing and Dialog Components.

The system components of the architecture are based on the proposal described in (Serban and Pauchet, 2013) for the use of AgentSlang components. The Debug/Log component allows the logging mechanism to be entirely distributed and independent. The logger manages the reception of debug messages in a centralized way. The System Monitor receives status messages from all subscribing components. These status messages are re-broadcasted to its subscribers, providing the source feedback.

User inputs are processed in the proposed architecture by the Automatic Speech Recognition and Text Inputs

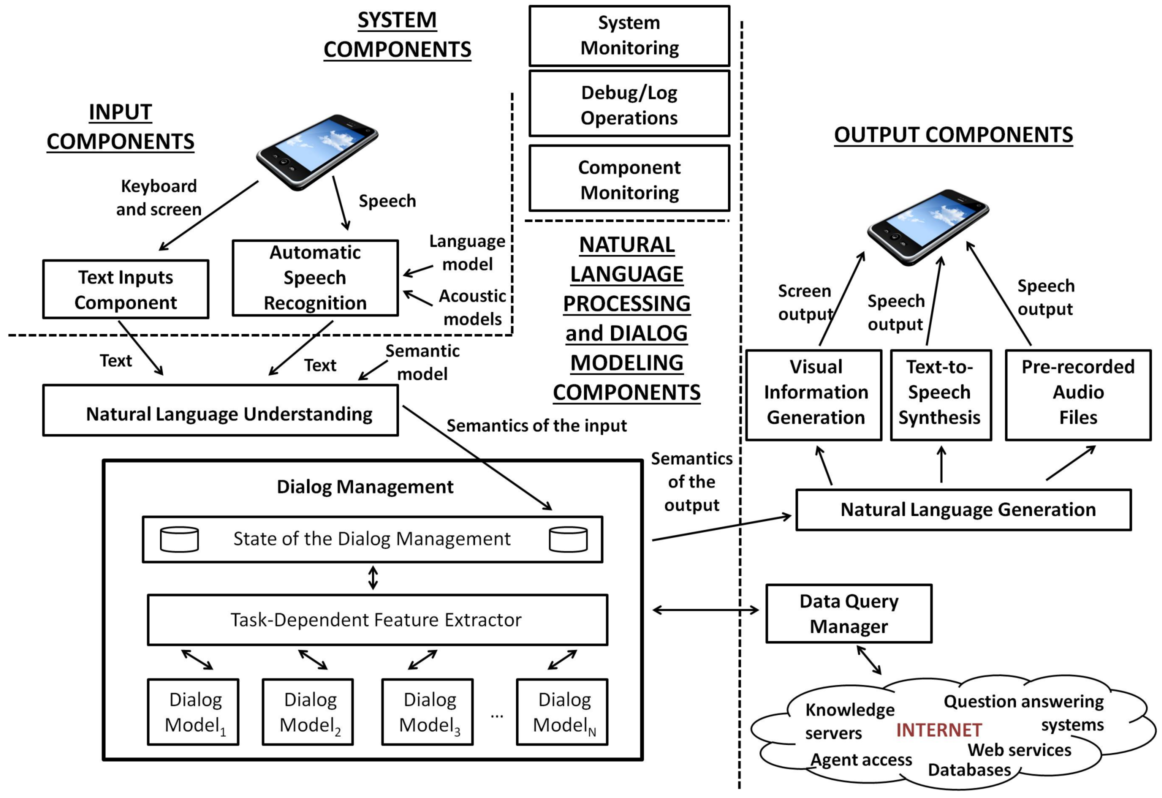


Figure 1: Scheme of the complete architecture of a dialog system with the integration of the proposed multi-task dialog management technique

Components. The Text Inputs Component allows to collect text inputs provided by means of the keyboard and tactile screen of portable devices. Moreover, it can subscribe to any channel and displays the received message as plain text.

The goal of speech recognition is to obtain the sequence of words uttered by a speaker. It is a very complex task, as there can be a great deal of variation in the input the recognizer must analyze, for example, in terms of the linguistics of the utterance, inter and intra speaker variation, the interaction context and the transmission channel (Tsilfidis et al., 2013). Once the dialog system has recognized what the user uttered, it is necessary to understand what they said. Natural language processing generally involves morphological, lexical, syntactical, semantic, and pragmatical knowledge (Wu et al., 2010).

3.1 Learning the task-specific statistical dialog models

As described in the introduction section, to develop the *Dialog Manager*, we propose the use of a multi-task system with specialized models dealing with each one of the subdomains or subtasks for which the dialog system has been designed.

Our proposed technique for statistical dialog modeling represents dialogs as a sequence of pairs (A_i, U_i), where A_i is the output of the system (the system response or turn) at time i, and U_i is the semantic representation of the user turn (the result of the understanding process of the user input) at time i; both expressed in terms of dialog acts (Griol et al., 2014). This way, each dialog is represented by:

$$(A_1, U_1), \cdots, (A_i, U_i), \cdots, (A_n, U_n)$$

where A_1 is the greeting turn of the system (e.g. Welcome to the system. How can I help you?), and U_n is the last user turn (i.e., semantic representation of the last user utterance provided by the natural language understanding component in terms of dialog acts).

The lexical, syntactic and semantic information associated with the speaker u's ith turn (U_i) is denoted as c_i^u. This information is usually represented by:

- the words uttered;
- part of speech tags, also called word classes or lexical categories. Common linguistic categories include noun, adjective, and verb, among others;
- predicate-argument structures, used by SLU modules in various contexts to represent relations within a sentence structure.
- named entities: sequences of words that refer to a unique identifier. This identifier may be a proper name (e.g., organization, person or location names), a time identifier (e.g., dates, time expressions or durations), or quantities and numerical expressions (e.g., monetary values, phone numbers).

Our model is based on the one proposed in (Bangalore et al., 2008). In this model, each system response is defined in terms of the subtask to which it contributes and the system dialog act to be performed.

The term A_i^a denotes the system dialog act (i.e., system action) in the ith turn, and ST_i^a denotes the subtask label to which the ith turn contributes. The interpretation process is modeled in two stages. In the first stage, the system dialog act is determined from the information about the user's turn and the previous dialog context, which is modeled by means of the k previous utterances. This process is shown in Eq.(1).

$$A_i^a = \underset{A^a \in \mathcal{A}}{\operatorname{argmax}} P(A^a | ST_i^a, ST_{i-1}^{i-k}, A_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (1)$$

where c_i^u represents the lexical, syntactic, and semantic information (e.g., words, part of speech tags, predicate-argument structures, and named entities) associated with speaker u's ith turn; ST_{i-1}^{i-k} represents the dialog subtask tags for utterances $i-1$ to $i-k$; and A_{i-1}^{i-k} represents the system dialog act tags for utterances $i-1$ to $i-k$.

In a second stage, the dialog subtask is determined from the lexical information, the dialog act computed according to Eq.(1), and the dialog context, as shown in Eq.(2).

$$ST_i^a = \underset{s^a \in \mathcal{S}}{\operatorname{argmax}}\, P(s^a | ST_{i-1}^{i-k}, A_{i-1}^{i-k}, c_{i-1}^{i-k}) \tag{2}$$

The prediction of the dialog subtask (ST_i^a) by means of Eq.(2) is carried out by a specific component in the architecture, which we have called the *Task-Dependent Feature Extractor*. This module is connected with the State of the Dialog Management component, which updates the current state of the dialog according to the semantic information provided by the *Natural Language Understanding* module after each user utterance. This information is provided to the Task-Dependent Feature Extractor for the prediction of the dialog subtask. According to this prediction, the *Task-Dependent Feature Extractor* selects the specialized dialog model that will be used by the dialog manager in the following turn of the dialog. Then, the selected specialized model is used to select the next action of the dialog system.

In our proposal, we consider static and dynamic features to estimate the conditional distributions shown in Eq.(1) and Eq.(2). Dynamic features include the dialog act and the task/subtask. Static features include the words in each utterance, the dialog acts in each utterance,and predicate-arguments in each utterance. All pieces of information are computed from corpora using n-grams, that is, computing the frequency of the combination of the n previous words, dialog acts, or predicate-arguments in the user turn.

The conditional distributions shown in Eq.(1) and Eq.(2) can be estimated by means of the general technique of choosing the maximum entropy (MaxEnt) distribution that properly estimates the average of each feature in the training data (Bangalore et al., 2008). This can be written as a Gibbs distribution parameterized with weights λ as Eq.(3) shows, where V is the size of the label set, X denotes the distribution of dialog acts or subtasks (DA_i^u or ST_i^u) and ϕ denotes the vector of the described static and dynamic features used for the user turns from $i-1 \cdots i-k$.

$$P(X = st_i | \phi) = \frac{e^{\lambda_{st_i} \cdot \phi}}{\sum_{st=1}^{V} e^{\lambda_{st_i} \cdot \phi}} \tag{3}$$

Such calculation outperforms other state of the art approaches (Bangalore et al., 2008), as it increases the speed of training and makes possible to deal with large data sets. Each of the classes can be encoded as a bit vector such that, in the vector corresponding to each class, the ith bit is one and all other bits are zero. Then, V-one-versus-other binary classifiers are used as Eq.(4) shows.

$$P(y|\phi) = 1 - P(\overline{y}|\phi) = \frac{e^{\lambda_y \cdot \phi}}{e^{\lambda_y \cdot \phi} + e^{\lambda_{\overline{y}} \cdot \phi}} = \frac{1}{1 + e^{-\lambda'_{\overline{y}} \cdot \phi}} \tag{4}$$

where $\lambda_{\overline{y}}$ is the parameter vector for the anti-label $\overline{y}$ and $\lambda'_{\overline{y}} = \lambda_y - \lambda_{\overline{y}}$.

4. Practical application

We have applied our proposal to the problem solving domain of a dialog system that acts as a customer support service to help solving simple and routine software/hardware repairing problems, both at the domestic and professional levels.

The definition of the system's functionalities and dialog strategy was carried out by means of the analysis of 250 human-human conversations (7215 user turns) provided by real assistants attending the calls of users with a software/hardware problem at the City Council of Leganés (Madrid, Spain). The labeling defined for this corpus contains different types of information, that have been annotated using a multilevel approach similar to the one proposed in the Luna Project (Stepanov et al., 2014). The first levels include segmentation of the corpus in dialog turns, transcription of the speech signal, and syntactic preprocessing with POS-tagging (i.e., word-category disambiguation) and shallow parsing (i.e., identification of the different constituents in a sentence). The next level consists of the annotation of main information using attribute-value pairs. The other levels of the annotation show contextual aspects of the semantic interpretation.

The attribute-value annotation uses a predefined domain ontology to specify concepts and their relations. The attributes defined for the task include *Concept*, *Computer-Hardware*, *Action*, *Person-Name*, *Location*, *Code*, *TelephoneNumber*, *Problem*, etc.

Dialog act (DA) annotation was performed manually by one annotator on speech transcriptions previously segmented into turns. The DAs defined to label the corpus are the following: i) Core DAs: *Action-request*, *Yes-answer*, *No-answer*, *Answer*, *Offer*, *ReportOnAction*, *Inform*; ii) Conventional DAs: *Greet*, *Quit*, *Apology*, *Thank*; iii) Feedback-Turn management DAs: *ClarificationRequest*, *Ack*, *Filler*; iv) Non interpretable DAs: *Other*.

The original FrameNet[1] description of frames and frame elements was adopted for the predicate-argument structure annotation, introducing new frames and roles related to hardware/software only in case of gaps in the FrameNet ontology. Some of the frames included in this representation are *Telling*, *Greeting*, *Contacting*, *Statement*, *Recording*, *Communication*, *Being operational*, *Change operational state*, etc.

The basic structure of the dialogs is usually composed by the sequence of the following tasks: *Opening*, *Problem-statement*, *User-identification*, *Problem-clarification*, *Problem-resolution*, and *Closing*. This set of tasks contains a list of subtasks, such as *Problem-description*, *Problem-Request*, *Problem-Confirmation*, *Brand-Identification*, *Model-Identification*, *Help-Request*, *Message-Confirmation*, *Name-Identification*, etc. The shared plan is represented as a data register that encapsulates the task structure, dialog act structure, attribute-values and predicate-argument structure of utterances. Figure 2 shows the transcription of a dialog included in the training corpus.

During the *Problem-statement* task, the caller explains the problem the reasons why he/she calls the help-desk. In the *User-identification* task, the operator asks for additional information regarding the identity of the caller. Once the caller has described the problem, the operator can ask for additional information to clarify it during the *Problem-clarification* task.

During the *Problem-resolution* task, the operator asks the user to perform specific tests. We have defined nine different subtasks inside this generic segment, given that our goal is to detect not only that the dialog is in this

[1] https://framenet.icsi.berkeley.edu/fndrupal/

A_1: Welcome to the system. How may I help you?
U_1: Hello, my scanner does not scan correctly.
A_2: Tell me your name and surname.
U_2: My name is Pedro López.
A_3: Can you tell me your phone extension?
U_3: Eight nine forty-five.
A_4: Can you tell me the brand and model?
U_4: HP Deskjet 3050.
A_5: Is the scanner on now?
U_5: It is on.
A_6: Can you tell me the message on the scanner display?
U_6: Network scanner connection is lost.
A_7: Can you see the scanner from the control panel of your PC?
U_7: It is active.
A_8: Please try the following steps. Say next after each step. Turn off the scanner and on again.
U_8: Next.
A_9: Make sure the scanner power cord connects directly to an electrical outlet.
U_9: Next.
A_{10}: Has this process solved your problem?
U_{10}: No.
A_{11}: Close all running programs on your computer, then shut down the computer.
U_{11}: Next.
A_{12}: Turn on the computer and the scanner.
U_{12}: Next.
A_{13}: Has this process solved your problem?
U_{13}: Yes.
A_{14}: Anything else?
U_{14}: No, thank you.
A_{15}: Most of these steps will also work for troubleshooting an office printer. Have a nice day!

Figure 2: Example of a dialog acquired with recruited users (translation from Spanish to English)

segment, but also what are the specific problem that has to be r esolved: *Printer* (P4), *Network connection* (P5), *PC going slow* (P6), *Monitor* (P7), *Keyboard* (P8), *Mouse* (P9), *CD-DVD player* (P10), *Power supply* (P11), and *Virus* (P12). The operator assigns a ticket number for the current call if the problem has not been solved after this task. The user must take note of this number and inform about this to the operator. The dialog ends at the *Closing* phase, in which the operator also tries to give a useful advice related to the described problem.

The complete set of human-human dialogs were manually labeled including this task/subtask information.

This information was incorporated for each user and system turn in the dialogs. Two versions of the dialog system have been developed. The first one (*Generic System*) uses a generic dialog model for the task, which employs a dialog model learned with the complete training corpus to select the next system response. The second one (*Multi-models System*) employs 27 specialized dialog models developed according our proposal for dialog management. Each one of the specialized dialog models deals with a specific subtask of the dialog (from the opening task, the problem statement, user identification, practical resolution of each specific problem, out of the task conversations, and closing of the dialog).

5. Experiments and results

We have completed a comparative evaluation of the two practical dialog systems developed for the task using a set of 25 scenarios covering the different problems that users can formulate to the system. A total of 150 dialogs were recorded from interactions of six recruited users for our department employing the two dialog systems. These users selected the specific scenarios in a random order and using a random assignment of both systems. An objective and subjective evaluation were carried out.

The following measures were defined in the objective evaluation to compare the dialogs acquired with the dialog systems: i) Dialog success rate; ii) Dialog length: average number of turns per dialog, number of turns of the shortest dialog, number of turns of the longest dialog, and number of turns of the most observed dialog; iii) Different dialogs: percentage of different dialogs with respect to the total number of dialogs, and number of repetitions of the most observed dialog; iv) Turn length: average number of actions per turn; v) Participant activity: number of turns in the most observed, shortest and longest dialogs; v) Confirmation rate, computed as the ratio between the number of explicit confirmation turns and the total number of turns in the dialog; and vi) Error correction rate, computed as the number of errors detected and corrected by the dialog manager divided by the total number of errors.

Table 1 presents the results of the objective evaluation. As can be observed, both dialog systems could interact correctly with the users in most cases for the two systems. However, the *Multi-models System* obtained a higher success rate, improving the initial results by a 6% absolute. Using the *Multi-models System*, the average number of required turns is also reduced from 29.1 to 24.3.

It can also be observed that when *Multi-models System* was used, there was a reduction in the average number of turns and in the number of turns in the longest, shortest and most observed dialogs. These results show that the use of specialized dialog models made it possible to reduce the number of necessary system actions to attain the dialog goals for the different tasks. In addition, the results show a higher variability in the dialogs generated with *Multi-models System* as there was a higher percentage of different dialogs and the most observed dialog was less repeated. There was also a slight increment in the mean values of the turn length for the dialogs collected with *Multi-models System* due to the better selection of the system actions in the improved strategy.

The confirmation and error correction rates were also improved by using *Multi-models System* as it required less data from the user, thus reducing the number of errors in the automatic speech recognition process. A problem occurred when the user input was misrecognized but it had high confidence score, in which case it was forwarded to the dialog manager. However, as the success rate shows, this problem did not have a remarkable

	Generic System	*Multi-models System*
Dialog success rate	89.0%	95.0%
Average number of turns per dialog	29.1	24.3
Percentage of different dialogs	84.6%	88.7%
Repetitions of the most observed dialog	4	3
Average number of actions per turn	1.2	1.5
Number of user turns of the most observed dialog	12	10
Number of user turns of the shortest dialog	9	6
Number of user turns of the longest dialog	15	11
Confirmation rate	38%	36%
Error correction rate	0.89%	0.94%

Table 1: Results of the high-level dialog measures for the Generic System and the Multi-models System

impact on the performance of the dialog systems.

Additionally, we grouped all user and system actions into three categories: "goal directed" (actions to provide or request information), "grounding" (confirmations and negations), and "other". Table 2 shows a comparison between these categories. As can be observed, the dialogs provided by the *Multi-models System* have a better quality, as the proportion of goal-directed actions is higher than the values obtained for the *Generic System*.

	Generic System	*Multi-models System*
Goal-directed actions	68.21%	74.35%
Grounding actions	30.76%	24.76%
Rest of actions	1.03%	0.89%

Table 2: Proportions of dialog spent on-goal directed actions, ground actions and other possible actions

We also asked the users to complete a questionnaire to assess their subjective opinion about the system performance. The questionnaire had six questions: i) Q1: *How well did the system understand you?*; ii)Q2: *How well did you understand the system messages?*; iii) Q3: *Was it easy for you to get the requested information?*; iv) Q4: *Was the interaction with the system quick enough?*; v) Q5: *If there were system errors, was it easy for you to correct them?*; vi) Q6: *In general, are you satisfied with the performance of the system?* The possible answers for each one of the questions were the same: *Never/Not at all*, *Seldom/In some measure*, *Sometimes/Acceptably*, *Usually/Well*, and *Always/Very Well*. All the answers were assigned a numeric value between one and five (in the same order as they appear in the questionnaire).

Table 3 shows the average results of the subjective evaluation using the described questionnaire. It can be observed that using either *Generic System* or *Multi-models System* the users perceived that the system understood them correctly. Moreover, they expressed a similar opinion regarding the easiness for correcting system errors.

However, users said that it was easier to obtain the information specified for the different objectives using *Multi-models System*, and that the interaction with the system was more adequate with this dialog manager. Finally, the users were more satisfied with the system employing *Multi-models System*.

	Generic System	*Multi-models System*
Q1	4.7	4.8
Q2	4.3	4.4
Q3	4.2	4.7
Q4	4.2	4.6
Q5	4.1	4.3
Q6	4.4	4.7

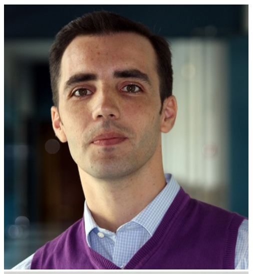
Table 3: Results of the subjective evaluation with recruited users (1 = lowest, 5 = highest).

6. Conclusions and future work

In this paper, we have described a technique to develop dialog managers for multi-task dialog systems. Our proposal is based on dealing with each one of the dialog subtasks or dialog objectives by means of specific dialog models specialized in each one of them. A statistical dialog model has been defined for the practical implementation of our proposal. This model considers the previous history of the dialog and it is selected according to the predicted dialog subtask, and the decision of the next system action. Although the construction and parameterization of the dialog model depends on expert knowledge of the task, by means of our proposal, we facilitate to develop dialog systems that have a more robust behavior, better portability, and are easier to be extended or adapted to different user profiles or tasks.

The results of the evaluation of our proposal for a dialog system acting as a help-desk system show that the number of successful dialogs is increased in comparison with using a generic dialog model learned for the complete task. Also, the dialogs acquired using the specific dialog models are statistically shorter and present a better quality in the selection of the system responses. For future work, we want to consider the incorporation of additional information regarding the user, such as specific user profiles adapted to the each application domain. Finally, we want also to apply our proposal to more complex dialog domains.

Vitae



David Griol obtained his Ph.D. degree in Computer Science from the Technical University of Valï¿½ncia (Spain) in 2007. He has also a B.S. in Telecommunication Science from this University. He is currently professor at the Department of Computer Science in the Carlos III University of Madrid (Spain). He has participated in several European and Spanish projects related to natural language processing and dialog systems. His research activities are mostly related to the development of statistical methodologies for the design of spoken dialog systems. His research interests include dialog management, corpus-based methodologies, user modeling, adaptation and evaluation of spoken dialog systems and machine learning approaches.

José Manuel Molina is Full Professor at the Carlos III University of Madrid (Spain) and director of the Department of Computer Science. Previously he has been director of the Ph.D. Program in Computer science and Technology of this University. He obtained Ph.D. and B.S. degrees in Telecommunication Science from the Technical University of Madrid. He is currently the director of the research group of Applied Artificial Intelligence (`www.giaa.inf.uc3m.es`). His research areas are focused on the application of Artificial intelligence in multi-agent systems, fuzzy logic, evolutionary computation, artificial vision, multi-sensor systems, signal processing, context-aware systems and user-centered applications to access information and services.

7. References

Bangalore, S., DiFabbrizio, G., and Stent, A., 2008. Learning the Structure of Task-Driven Human-Human Dialogs. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1249–1259.

Barnard, E., Halberstadt, A., Kotelly, C., and Phillips, M., 1999. A Consistent Approach To Designing Spoken-dialog Systems. In *Proc. of ASRU'99*, pages 1173–1176. Keystone, Colorado, USA.

Cuayáhuitl, H., Renals, S., Lemon, O., and Shimodaira, H., 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proc. of ASRU'05*, pages 290–295. San Juan, Puerto Rico.

Ferreira, E. and Lefevre, F., 2015. Reinforcement-learning based dialogue system for human-robot interactions with socially-inspired rewards. *Computer Speech and Language*, 34(1):256–274.

Griol, D., Callejas, Z., López-Cózar, R., and Riccardi, G., 2014. A domain-independent statistical methodology for dialog management in spoken dialog systems. *Computer, Speech and Language*, 28(3):743–768.

Knott, A. and Vlugter, P., 2008. Multi-agent human-machine dialogue: issues in dialogue management and referring expression semantics. *Artificial Intelligence*, 172(2-3):69–102.

Lee, C., Jung, S., Kim, K., and Lee, G. G., 2010. Hybrid approach to robust dialog management using agenda and dialog examples. *Computer Speech and Language*, 24(4):609–631.

Lee, G., Kim, H. K., Jeong, M., and Kim, J., 2015. *Natural Language Dialog Systems and Intelligent Assistants*. Springer.

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

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

McTear, M. F., Callejas, Z., and Griol, D., 2016. *The Conversational Interface*. Springer.

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

Ota, R. and Kimura, M., 2014. Proposal of open-ended dialog system based on topic maps. *Procedia Technology*, 17:122–129.

Pieraccini, R., 2012. *The Voice in the Machine: Building computers that understand speech*. MIT Press.

Planells, J., Hurtado, L., Sanchis, E., and Segarra, E., 2012. An Online Generated Transducer to Increase Dialog Manager Coverage. In *Proc. of Interspeech'12*, pages 234–237. Portland, USA.

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

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

Serban, O. and Pauchet, A., 2013. Agentslang: A fast and reliable platform for distributed interactive systems. In *Proc. of ICCP'13*, pages 35–42.

Stepanov, E., Riccardi, G., and Bayer, A., 2014. The Development of the Multilingual LUNA Corpus for Spoken Language System Porting. In *Proc. of LREC'14*, pages 2675–2678.

Traum, D. and Larsson, S., 2003. *The Information State Approach to Dialogue Management*, chapter Current and New Directions in Discourse and Dialogue, pages 325–353. Kluwer.

Tsilfidis, A., Mporas, I., Mourjopoulos, J., and Fakotakis, N., 2013. Automatic speech recognition performance in different room acoustic environments with and without dereverberation preprocessing. *Computer Speech & Language*, 27(1):380–395.

Wu, W.-L., Lu, R.-Z., Duan, J.-Y., Liu, H., Gao, F., and Chen, Y.-Q., 2010. Spoken language understanding using weakly supervised learning. *Computer Speech & Language*, 24(2):358–382.

Young, S., Schatzmann, J., Weilhammer, K., and Ye, H., 2007. The Hidden Information State Approach to Dialogue Management. In *Proc. of ICASSP'07*, pages 149–152. Honolulu, Haway, USA.