Behavior Models for Learning and Receptionist Dialogs

Hartwig Holzapfel, Alex Waibel

interACT Research, Universität Karlsruhe, Germany

hartwig,waibel@ira.uka.de

Abstract

We present a dialog model for identifying persons, learning person names, and associated face IDs in a receptionist dialog. The proposed model allows a decomposition of the main dialog task into separate dialog behaviors which can be implemented separately and allow a mixture of handcrafted models and dialog strategies trained with reinforcement learning. The dialog model was implemented on our robot and tested in a number of experiments in a receptionist task. A Wizard-of-Oz experiment is used to evaluate the dialog structure, delivers information for the definition of metrics, and delivers a data corpus which is used to train a user simulation and component error model. Using these models we train a dialog module for learning a person's name with reinforcement learning.

Index Terms: dialog management, behavior models, reinforcement learning, wizard-of-OZ, learning dialogs

10.21437/Interspeech.2007-596

1. Introduction

In this paper we present a dialog model which implements a decomposition of the overall dialog task into separate dialog behaviors. The dialog model controls the behavior of a receptionist robot, which is given the task to receive persons, conduct a receptionist dialog, identify persons and learn names and associated voice and face IDs of unknown persons. A robot that works as a receptionist should be able to communicate with known and unknown persons equally, verify if he knows the person he talks to, and obtain the name of an unknown person in a task which requires registration of a person. He should then store collected user data to be able to recognize the person again later on. Our design goal was to implement a strategy for learning a person's name embedded into the more complex receptionist dialog.

The dialog model features two different types of learning. First, it implements learning dialogs for multimodal acquisition of new knowledge. Second, the model implements reinforcement learning as an integral part of defining a dialog strategy. As mentioned above, the dialog model suggests a decomposition of the main dialog task into separate dialog behaviors. Their coexistence allows to mix handcrafted dialog strategies and dialog strategies trained with reinforcement learning. The decision which behavior to select and to execute the next action is done per situation by a situation model. All behaviors share a common discourse, state- and slot- model and can select within a shared action space for execution of dialog moves.

A decomposition of a dialog architecture is also proposed by [1] who use agents for distributed interaction tasks, and among others [2], who define agents in a hierarchical task structure. Related architectures are e.g. multimodal user registration [3] and [4] who employ voice ID to identify people in dialog and furthermore use reinforcement learning for this task.

Based on a Wizard-of-Oz experiment we could evaluate the dialog structure, which provides a motivation for decomposition into separate behaviors and modules. In the following we use the term module, when we refer to the technical implementation of a behavior or want to describe how a behavior is implemented. We use the term behavior when we refer to what a module does and describe its higher level usage. The Wizard-of-Oz experiment also delivers information for the definition of metrics which are important to evaluate the dialogs and later to define reward functions for reinforcement learning. It also delivers a data corpus which is used to train a user simulation and component error model, used for a dialog simulation. Further experiments were conducted with an automated system integrating insights we gained from the Wizard-of-Oz experiment and fully implemented behavior models, and also with mixing handcrafted strategies and optimized models trained with reinforcement learning. The following sections describe the dialog architecture with dialog behaviors and experiments conducted with the system.

2. Dialog System Architecture

The dialog architecture is based on the TAPAS dialog manager, which has been developed for multimodal human-robot communication, to control a humanoid robot, interact with it, and support its learning. In recent experiments we have added functionality for interactive learning, as well as proactive dialog behavior to initiate conversations with persons [5].

The dialog manager is embedded in an integrated perceptual system with audio and speech recognition and visual processing. Speech recognition is done with the IBIS decoder [6] in JANUS using context free grammars. The speech recognizer can detect unknown words, which in our application is especially important to recognize unknown person names. It supports spelling recognition which is used in combination with grapheme to phoneme conversion to learn new names. It offers a tight integration with the dialog manager, which improves context dependent natural language understanding and allows context dependent control for switching language models and weighting grammar rules. The robot is also equipped with visual processing which can track persons, detect and resolve pointing gestures, detect faces, recognize and learn face IDs.

2.1. Discourse Model

The dialog manager receives input from these recognition components, converts input events to semantics (e.g. natural language understanding for speech), interprets these semantics in the given context and updates the discourse model. A state model contains variables representing various characteristics of the current dialog state. Furthermore, a slot model contains variables representing important information which have been extracted from the dialog. The models are shown in figure 1.

After updating the discourse, slot- and state model the dialog strategy executes the next action(s). Different from our previous model, in which the strategy selects matching dialog goals and executes dialog moves to achieve such a goal, the new dialog strategy is split into different behaviors. To allow integration of these independent strategies, the system relies on a generic discourse model that can be used by all strategies, which also supports a shared context model influenced by system actions.

The discourse model is updated with generic algorithms using unification, and an expectation model which is filled by expectations generated by system actions. These expectations also form a generic mechanism to weight grammar rules in the speech recognizer and to interpret expected user input semantics within the generated context [7]. The discourse representation is used by application specific updates to fill the slot model and update the state model. While discourse representation and context resolution algorithms are domain independent, the slot model and its update need to be designed in accordance to the application specific dialog model. These generic constructs assist the separation of discourse update and dialog strategies and decouples their interdependence. As we will see in the next subsection, also the dialog strategy and the dialog moves/system actions are decoupled, which furthermore reduces the dialog strategy to a state- or situation-based selection of actions. Actions in turn are linked to the expectation model and discourse update.

2.2. Dialog Modules

After each discourse update a situation model selects a dialog behavior to execute its strategy. Each dialog behavior is implemented by a dialog module. It implements its own strategy, i.e. each behavior has its own controller and is responsible for action selection, but can access common actions. Each behavior can execute lower level actions, internally represented as dialog moves. These moves themselves are complex enough to generate TTS or multimodal output, influence the expectation model and abstract over a language specific layer [7], so that the behavior implementation is only concerned with the implementation of a strategy. By separation of concerns this allows a completely independent implementation of each dialog module where handwritten and statistical models can be mixed. The switching between the dialog behaviors is implemented by a situation model which decides for each dialog state, which behavior shall be applied. Integration into the architecture is shown in figure 2b.

2.3. Reinforcement Learning of Behaviors

The previous subsections describe how to decompose a complex dialog model into the task of controlling smaller parts. Each such module has its own implementation and control over its own state and action space. Thus, such a decomposition allows to apply different control strategies in each module. The first implementation of each module at first was a handcrafted strategy following our traditional goal-oriented approach. Each module can then separately be replaced by a different strategy. In our experiments we replaced only the name learning module, which qualified as the most complex one with the most actions and highest number of turns during the experiments. The easy transition from the handcrafted strategy to the reinforcement strategy is possible by a clear separation of the behavior modules with a common discourse model, slot-model and state space, and thus simply replaces the old module. Reinforcement learning uses a Markov-Decision-Process (MDP) which has its own state space. This state space is defined on top of the common slot model and state space, and is defined specifically for each behavior, which is illustrated in figure 1.

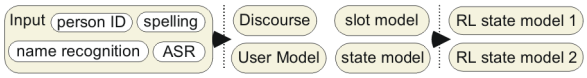

Figure 1: State model used by the RL agent.

Following previous experiments [8], we adopted the development process, including data collection in the WOZ-study, user modeling, design of the MDP and reward function and training of the dialog model in a user simulation. Details are described in the following experiments section.

3. Experiments

In our setup the robot plays the role of a receptionist. Its task is to wait in the corridor, greet arriving persons, find out what they want and help them with directions. In our setup each guest has to deliver a parcel to a predefined person. An average dialog during the first day follows the schema: (i) greeting, (ii) system explains what it can do, (iii) the guest says that he has to deliver a parcel to person X, (iv) the guest is asked for his name in order to be announced to person X, the name is learned and stored (v) the system gives information in which room person X can be found, optionally gives directions, (vi) after returning the guest is asked if he could deliver the parcel, (vii) goodbye. Step (vi) was not used in the automated experiment since it was impossible to recognize persons from behind without seeing the face. During the second day the dialog was much shorter depending on whether the person had been learned the previous day and could be recognized. Altogether, we created 20 different utterances as text-to-speech output. 16 of which are basic actions such as hello, request information or confirm information. The remaining four actions shall bring the user back on track if something doesn't work out in the interaction, e.g. a help utterance to pick up the hand-microphone for close speech, a hint to spell fluently, or explain system capabilities. At the first interaction the persons don't know that they are supposed to interact with the robot, so the robot needs to greet the persons on its own initiative. Figure 2a shows the view from the robot in the corridor with an arriving person.

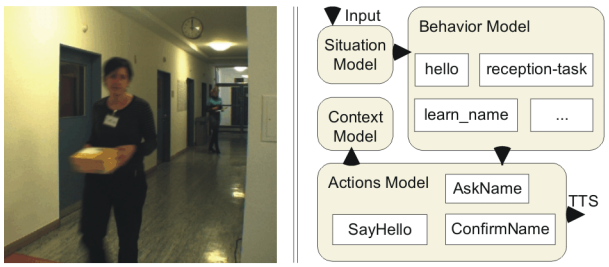

Figure 2: a.) truncated robot's view in the corridor, b.) dialog behaviors in the architecture

3.1. Wizard-of-Oz and Standalone Experiments

The first user study was conducted as a Wizard-of-Oz experiment with a fully integrated system, where the wizard only replaced the dialog decisions. Fully integrated means automatic person tracking and face ID, speech recognition, natural language understanding and user model. The wizard was provided with the spoken input, face ID hypothesis and user model, multimodal ID (combining face ID and spoken name), and a view

through the robot's camera, all integrated into a GUI window. An additional GUI panel provides the 20 utterances, all text-to-speech output, to communicate with the person in front of the robot. The utterances are predefined text strings, so that the wizard needn't type. Some utterances require parameters, e.g. when asking for a name, which are automatically resolved from the user model at execution. The camera view should have only been used to proactively initiate a conversation (engagement) when the person first arrives and after delivering the parcel, plus giving information to pick up the microphone (only distant speech activity). However, we found that the wizard was influenced by the picture in making own decisions when a wrong name or wrong face ID was recognized. The Wizard-of-Oz experiment was conducted on three consecutive days with 16 persons, where each person had to do one interaction per day and went through an interview after each interaction. The data, close and distant speech input, robot vision and system logging, were recorded for a data corpus. The experiment furthermore serves as a gold standard for successive automated experiments. The same setting was later used for the automated experiment, with 8 additional persons.

3.1.1. Wizard Questionnaire

In addition to the users, also the wizard was interviewed about which criteria were important for him to achieve by his strategy and how he perceived the interactions. The goal of these questions was to find problems with the current setup where the results didn't match the expectations, but even more to define measures for dialog strategy design and evaluation. We had five basic categories that should be scored in their importance from -2 to 2. Table 1 shows the categories for dialog length, friendliness, success in learning a person's name, naturalness and user acceptance, rated by the wizard by his goals (*Initial*) and how he perceived the dialogs after the experiment (*Perceived*). Friendliness has two numbers corresponding to one setup that was conducted with text-to-speech actions mimicking a clinical style and the second one mimicking an empathetic style. Due to the high difference this category was excluded from the reward function. Acceptance was not important to the wizard so it was also excluded. Thus, the three categories which the reward function is based on are dialog length, naturalness and success.

	length	**friendly**	**success**	**natural**	**accept**
Initial	1	-1/2	1	1	-1
Perceived	1	-1/2	2	-1	0

Table 1: Wizard ratings

3.1.2. Wizard-of-Oz Strategy Evaluation and Discussion

What can we learn from the conducted wizard of OZ strategy? First, the recorded interactions show a high variation in which actions are applied. While the wizard was pursuing one goal at a time, e.g. obtaining a persons name, the order of these goals varied. For example, sometimes the name was asked before asking for the (parcel) task and sometimes after it. High variation in the order of the goals, but a very consistent strategy in pursuing one goal at a time motivates the design of dialog behaviors.

The predefined answers helped the wizard to respond to the user input in time, and provided a similar and only slightly slower response time than in the fully automated experiment described later. The average dialog length in the Wizard-of-Oz experiment was 16,5 turns, the automated experiment provided comparable results, both with high variation depending if the users were recognized by the ID component, if their names were known or if they had to spell their names. The average interaction time was 190 seconds. Out of the 16 dialogs conducted during the first day (all unknown), 3 persons couldn't complete the name learning task, and 2 persons accepted a wrong name. During the second day one previously learned person, one unknown person and one person who accepted a wrong name could not be recognized. The other 13 dialog were successful. During the third day all persons could be recognized. In this experiment multiple names were stored that were not an exact match but sounded similar e.g. Michhel instead of Michael.

Some aspects, what can be referred to as human meta-knowledge, could neither be modeled in the automated experiment nor in the reinforcement learning optimization. For example, we applied spoken name recognition with unknown word detection, and in the case of an unknown word used a spelling recognizer with n-gram language model. The wizard sometimes rejected spelling results produced by the recognizer as impossibly being names. So despite the fact that we don't want to restrict the system to recognize only names within a fixed vocabulary, humans apply additional guesses what can be names and what not.

The following dialog is a sample from the experiments:
User: -approaches-
Syst: Hello?
User: -picks up microphone- Hello. ; *[act_hello]*
Syst: How may I help you?
User: I have to deliver a parcel for Mr. Brunn. ;
User: *[act_deliver [RECEIVER "Brunn"]]*
Syst: To announce you to Mr. Brunn please tell me your name.
User: My name is John. ; *[inform_name [NAME "OOV"]]*
Syst: Please spell your name!
User: J O H N. ; *[inform_spelling [SPELL "JOHN"]]*
Syst: Have I understood correctly that your name is John?
User: Yes. *[confirm_name]*
Syst: -gives room information-

3.1.3. Segmentation of Dialog Behaviors

For the second, fully automated experiment, we segmented the dialog task into the following behaviors: *hello*, *reception-task*, *learn-name*, *room-info-and-goodbye*. While in the previous experiment the text actions were selected by the wizard, now the dialog behavior which is selected by the situation model decides which action to take, as described in section 2, using the completed discourse model and dialog moves. Criteria for the segmentation were that all can be found in the wizard dialogs, that they form a partition of each dialog and that they minimize switching between behaviors. Each dialog behavior was implemented as a goal based strategy which selects a dialog move to execute. The suggested decomposition into modules seems appropriate. While the automated system uses a fixed set of rules which doesn't allow variation in the order of the behaviors for a given dialog state, it applies a different order if the person is recognized by face ID during the first few turns. The behaviors themselves show a high similarity to the segmented wizard's strategy.

3.2. Reinforcement Learning

After the first standalone experiment was conducted we started to optimize the dialog strategy by reinforcement learning. Here the modularization of the dialog model provides the basis to apply reinforcement learning single dialog modules. For rein-

forcement learning the *learn-name* module was selected. The modules task is verification of recognized persons and learning of names for new persons. The Module selects among actions to ask to speak the name, confirm a name, confirm a name where only face ID is given, ask to spell the name, help to spell fluently, and finally the decisions to accept a name or to abort the learning dialog without a result. Details of training exceed the scope of this paper and will be presented elsewhere.

3.2.1. Reward Function

Important for reinforcement learning is its reward function. It emerged on most parts from the wizard's feedback. The wizard rated length, success and naturalness highest. The reward function gives a high bonus (+20) for successful dialogs (name could be confirmed or be learned) and negatively rewards failed dialogs (-15) and incorrectly stored names (-30). Currently no difference is made between successful learning and successful confirmation. For name that are almost correct, we assigned a medium reward. These are names with a low Levenshtein distance to the real name (+10 for distance 1; 0 for distance 2). Further improvements should be possible with a low Levenshtein distance based on phonemes. To address 'dialog length', each turn is negatively rewarded by -1. For the category 'naturalness' subjective criteria need to be considered, from user feedback we found that repeating the same question is unnatural as are long dialogs. The first case also receives negative reward of -1, the second case is already negatively rewarded by dialog length. Some persons also perceived specific confirmation questions as unnatural. For confirmation of face ID, when no spoken name input was given, "did I understand correctly that your name is X?" was negatively rewarded and "you are X, right?" preferred in this case.

3.2.2. Training in User Simulation

Reinforcement learning of dialog strategies requires large amounts of data. This is problematic for dialog systems since usually real dialogs cannot be collected in such a magnitude. One way to solve this problem is to train the dialog strategy in a simulated environment which simulates user behavior as well as behavior of the system's recognition components (errors models). For strategy training we created a simulated environment using similar mechanisms as developed in previous work [8]. Once such models are created they can be used for reinforcement learning of the dialog strategy. For creating these models real data is required, which is provided by the Wizard-of-Oz experiment. The user simulation estimates a bi-gram probability for the user's speech act given the system's speech act, estimated from transcribed data. $p = P(\mathsf{action}_{\mathsf{user}}|\mathsf{action}_{\mathsf{system}})$

The input speech acts however, which are perceived by the system, have a higher variation due to errors made by the recognition components. The error model is thus modeled separately and is applied to the output of the user simulation. Typical errors in the receptionist system are incorrect recognitions like wrong spelling hypotheses, and speech segmentation errors.

In addition to speech input, the dialog system's ID component obtains face ID input which is provided by the simulation as a sequence of images for each turn, and currently also voice ID is integrated. For both ID components, samples of real recorded data is replayed in simulation by concatenating snippets of recorded data. To keep training time short, further optimizations are possible in pre-calculating classification results which are then actually used during simulation. A dialog turn with this configuration now takes roughly 2 ms on a standard 3 GHz Pentium processor.

4. Conclusion and Outlook

We have presented an approach for decomposition of a complex dialog strategy into smaller "behaviors" which are selected via a situation model. Each behavior is implemented as a dialog module which contains its own dialog strategy, either a handcrafted one or optimized by reinforcement learning. The decomposition into modules was motivated by a Wizard-of-Oz experiment, which furthermore was used to create a reward function and to collect a data corpus from which a user simulation and component error models are trained. It provides an appropriate means for separation of concerns and allows a mixture of reinforcement models and handcrafted dialogs.

The conducted experiments and user feedback suggest improvements of some recognition components, but also the integration of additional features for the dialog, e.g. to prevent interruption of the user. The user ID model is a straightforward implementation, we expect further improvements with an integrated multimodal user ID model, further voice ID integration and improved unknown person detection. While this paper focuses on the decomposition of the dialog and the implementation of behaviors, we are currently working to extend the control methods for behavior switching.

5. Acknowledgements

This work was supported in part by the German Research Foundation (DFG) as part of the Collaborative Research Center 588 "Humanoid Robots - Learning and Cooperating Multimodal Robots".

6. References

[1] M. Turunen and J. Hakulinen, "Jaspis2 - an architecture for supporting distributed spoken dialogues," in *Proceedings of Eurospeech*, 2003.

[2] D. Bohus and A. Rudnicky, "Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda," in *Proceedings of Eurospeech*, 2003.

[3] F. Huang, J. Yang, and A. Waibel, "Dialogue management for multimodal user registration," in *Proc Int Conf on Spoken Language Processing*, 2000.

[4] F. Krsmanovic, C. Spencer, D. Jurafsky, and A. Ng, "Have we met? mdp based speaker id for robot dialogue," in *Proceedings of Interspeech*, 2006.

[5] H. Holzapfel, T. Schaaf, H. K. Ekenel, C. Schaa, and A. Waibel, "A robot learns to know people - first contacts of a robot," *KI 2006: Advances in Artificial Intelligence, Springer LNCS*, vol. 4314, 2007.

[6] H. Soltau, F. Metze, C. Fuegen, and A. Waibel, "A one pass- decoder based on polymorphic linguistic context assignment," in *Proc. of ASRU*, 2001.

[7] H. Holzapfel and A. Waibel, "A multilingual expectations model for contextual utterances in mixed-initiative spoken dialogue," in *Proceedings of Interspeech*, 2006.

[8] T. Prommer, H. Holzapfel, and A. Waibel, "Rapid simulation-driven reinforcement learning of multimodal dialog strategies in human-robot interaction," in *Proceedings of Interspeech*, 2006.