Using Graphical Models for an Intelligent Mixed-Initiative Dialog Management System

Stefan Schwärzler, Günther Ruske, Frank Wallhoff, and Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München
80290 Munich, Germany
{sts,rus,waf,ri}@mmk.ei.tum.de

Abstract. The main goal of dialog management is to provide all information needed to perform e. g. a SQL-query, a navigation task, etc. Two principal approaches for dialog management systems exist: system directed ones and mixed-initiative ones. In this paper, we combine both approaches mentioned above in a novel way, and address the problem of natural intuitive dialog management. The objective of our approach is to provide a natural dialog flow. The whole dialog is therefore represented in a finite state machine: the information gathered during the dialog is represented in the states of the finite state machine; the transitions within the state machine denote the dialog steps into which the dialog is separated. The information is obtained from each natural spoken sentence by hierarchical decoding into tags, e. g. the name-tag and the address-tag. These information tags are gathered during the dialog; either by human initiative or by distinct questioning by the dialog manager. The models use information from the semantic information tags, the dialog history, and the training corpus. From all these integrated parts we achieve the best path to the end of the dialog by Viterbi decoding through the transition network after each information step. From the Air Travel Information System (ATIS) database, we extract all 21650 naturally spoken questions and the SQL-queries as answers for the trainings phase. The experiments have been realized on 200 automatically generated dialog sentences. The system obtains the semantic information in all test-sentences and leads the dialogs successfully to the end. In 66.5% of the sample dialogs we achieve the minimum of the required dialog steps. Hence, 33.5% of the dialogs have over-length.

Keywords: dialog management, learning, knowledge management, intelligent systems.

1 Introduction

For a spoken dialog system one can distinguish between five task areas. Depending on the system these areas are more or less developed and often their transitions are fluent [6]. The speech recognizer recognizes spoken phonemes and returns a sequence of words according to a lexicon. The sentence analysis (parsing) assigns a meaning to

M.J. Smith and G. Salvendy (Eds.): Human Interface, Part II, HCII 2009, LNCS 5618, pp. 201–209, 2009.

this sequence and translates the ordered relevant information into system language [8]. The dialog management determines the dialog strategy and thus, how the system responds to the user's input. During the communication with external sources, the information is written to or read from databases. The generation answer is virtually the opposite of the sentence analysis and translates words from the system language into a sequence of words that the user understands. The audio front-end is formed by a text-to-speech system. A sequence of written words is converted into a spoken text by a speech synthesis. All components are depicted in Fig. 1.

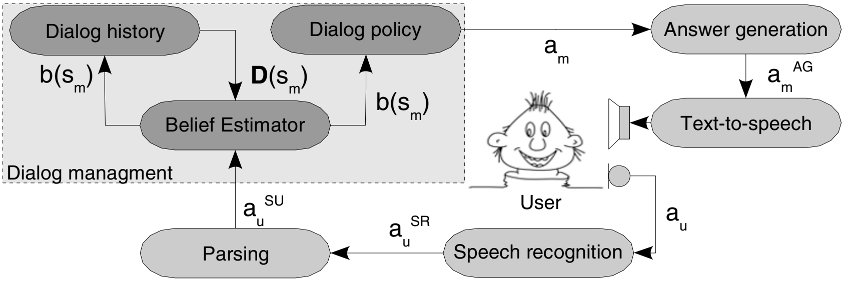

Fig. 1. Components of a speech dialog system: semantic slots and the dialog history estimates a dialog policy and ask the user over a text-to-speech system

The dialog management differs from all other components in Fig. 1. That is, why standard learning techniques are not suitable. A dialog management system requires a decision on the dialog step and which system answer should be generated in the next step.

Two principal approaches for dialog management systems exist: rule-based ones and adaptive probabilistic ones. With trindikit [4] exists a toolkit for rapid development of update rules, information states and dialog transitions. Beside the rule-based systems there are adaptive probabilistic approaches in the dialog procedure, published in [5, 7]. In contrast to the Partial Observable Markov Decision Process (POMDP) [11] we do not calculate the policy in every dialog step. Thereby we reduce the runtime-complexity.

In this paper, we use the following components for decision-making: inputs from the user $\mathbf{a}_t^{SR}$ run through a parsing system and deliver semantic slots $\mathbf{a}_t^{SU}$ to the belief estimator [8]. As the next belief state b(t+1) is dependent on the current belief state b(t), the dialog history D(sm) is augmented by the current belief state b(t). Simultaneously, the user is achieved of an action $\mathbf{a}_t^{m}$ by the text-to-speech system. This procedure is repeated until the user reaches the dialog goal or the dialog aborts.

The paper is organized as follows: after the introduction of dialog techniques and their strategies, the novel mixed-initiative dialog approach is presented in the next section. A corpus description with concrete use case follows in section 4. After the presentation of the experiments and their results, the treatise closes with a summary and additional experiments to be added in the future.

2 Dialog Strategies

A dialog strategy follows the target to obtain all semantic information to create a SQL[1] statement of an estimated user goal. To that end, different methods are described below.

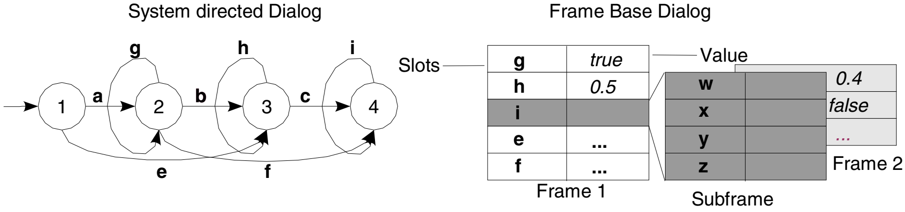

Fig. 2. The user inputs are controlled and verified in a system-directed dialog system by finite-state machines. Frame-based dialog systems catch the initiatives from the user [6]. The user is allowed to confront the system with any semantic slot in different time steps.

2.1 System Directed Systems

Semantic slots are specified by the system in a fixed sequence, realized by a finite-state machine (see Figure 2). The user has no possibility to form the dialog in run-time. In VoiceXML, the W3Consortium has created a standard for system-directed dialogs [10].

2.2 Frame Based Systems

Frame based systems use semantic templates. The system asks the user to fill open semantic slots. The strategy for filling can be realized in rule-based or probabilistic systems. The user is not constrained to answer directly system directed questions. Optionally the user can answer more semantic slots in one utterance.

2.3 Mixed Initiative Systems

In this paper, we combine the advantages of both approaches: An agent based system allows complex combinations between the user and the system, whereas both user and system can initiate the dialog. For passive or inexperienced users or even on problems in the dialog fluency, the system can control the dialog to the user goal. Beside the control of complex dialog actions, the mixed initiative system can change the dialog topic.

3 Novel Approaches

The primary goal of this paper is the estimation and the achievement of a user goal to create a SQL statement. To that end, we evolve a semantic frame (frame-based or

[1] SQL: Structured Query Language: A database computer language designed for the retrieval and management of data in relational database management systems.

mixed-initiative), which can be filled through pointed questions to the user. The dialog strategy is derived from a Graphical Model. The parameters of the model are learned by naturally spoken sentences in the ATIS corpus.

3.1 Graphical Model

A Graphical Model combines the theory of probabilities and graphs. Edges model the statistical dependencies between the variables (nodes). Fig. 3 shows the graphical model of a mixed-initiative dialog system. It is an extension of a discrete HMM, which has been realized with the Graphical Model Toolkit (GMTK) [2].

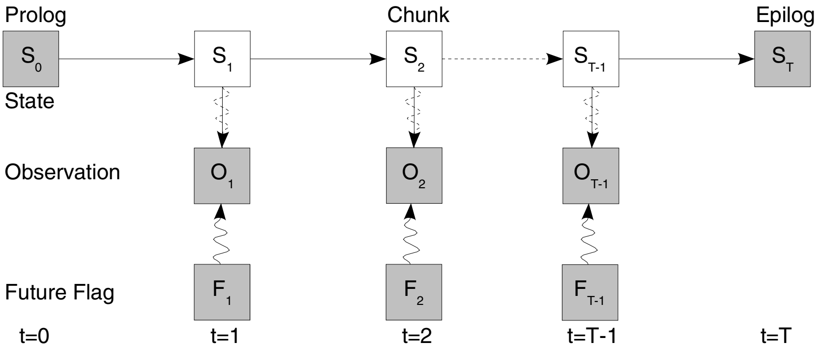

Fig. 3. Dialog strategy design by a stochastic model: The parameters of the states $\mathbf{S}^{m}_{t}$ from the dialog model are created from the previous model states $\mathbf{S}^{m}_{t-1}$ and the user utterances $\mathbf{a}^{SU}_{t}$. The future flag $\mathbf{F}_1$ controls the matrix of the observation $\mathbf{O}_t$.

The model contains the observable nodes $\mathbf{s}_0$ and $\mathbf{s}_T$, which are called as prolog and epilog of the graphical model. The chunks $\mathbf{s}_1$ till $\mathbf{s}_{T-1}$ are hidden nodes. The observations ($\mathbf{o}_t$) of the semantic slots $\mathbf{a}^{SU}_{t}$ are linked by an alternating transition with the $\mathbf{s}_t$. The statistical joint probability in Fig. 3 can be factorized with Eq. 1.

$$p(\mathbf{o}, \mathbf{s}) = p(s_0) \prod_{t=1}^{T-1} p(s_t|s_{t-1}\, p(\mathbf{o}_t|s_t)\, p(s_T) \tag{1}$$

3.2 Information Slots

In our novel approach for dialog management, information slots are introduced. They are capable of holding the information tags: if one certain information tag is available, its corresponding information slot is filled. Each dialog step is either initiated by the system or by the human dialog partner. The ending of the user's input also denotes the end of one dialog step. The information available for the current dialog task is evaluated after each dialog step by checking which additional information slots have been filled

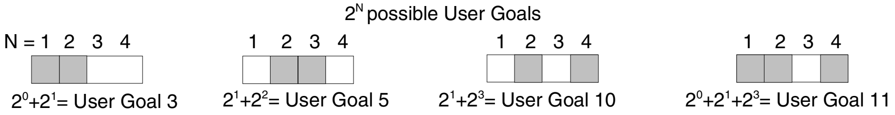

Fig. 4. User Goals, defined in binary configuration, are learned from a trainings corpus

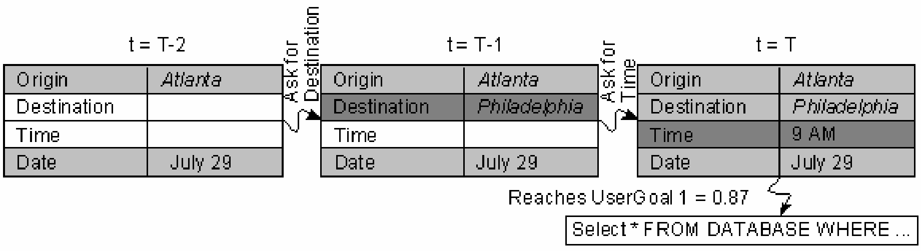

Fig. 5. On every time step t the semantic frame will be filled with a slot-value pair. Here the user will at first be asked for the destination, according the trained Graphical Model, before the time.

during the preceding dialog step. Each combination of available information tags, and hence, each filled information slot, is represented by one state in a finite state machine: if N information tags are required, 2^N states are defined (see Fig. 4.)

A transition from one state to another is made between two dialog steps, depending on the new information gathered and the missing information tags indicated by empty information slots. Missing information can now be asked for by the dialog system by a machine initiative (see Fig. 5). The information tags are thereby weighted depending on their relevance for the current dialog. The more important one certain information tag is for the dialog task, the earlier this information is asked for by the system if not provided during the preceding dialog steps. The transitions within the state machine can be either learned from a database or predefined by the dialog designer. By using a dataset, containing natural, task-specific example dialogs, as training corpus, the transitions can be adopted or learned in order to provide a natural dialog-flow. This depends on the information tags (and hence, the current state of the finite state machine) being already available or missing. Information tag-combination which can not be found in the training corpus can be provided by the above mentioned weighting and absolute discounting. Thereby, a probabilistic modeling of the dialog is achieved by using various models of dynamic Bayesian networks (DBNs).

3.3 Trellis Representation

All possible transitions from a belief state **b**(t-1) according to **b**(t) are learned iteratively by [1] and represented in a trellis diagram (see Fig. 6).

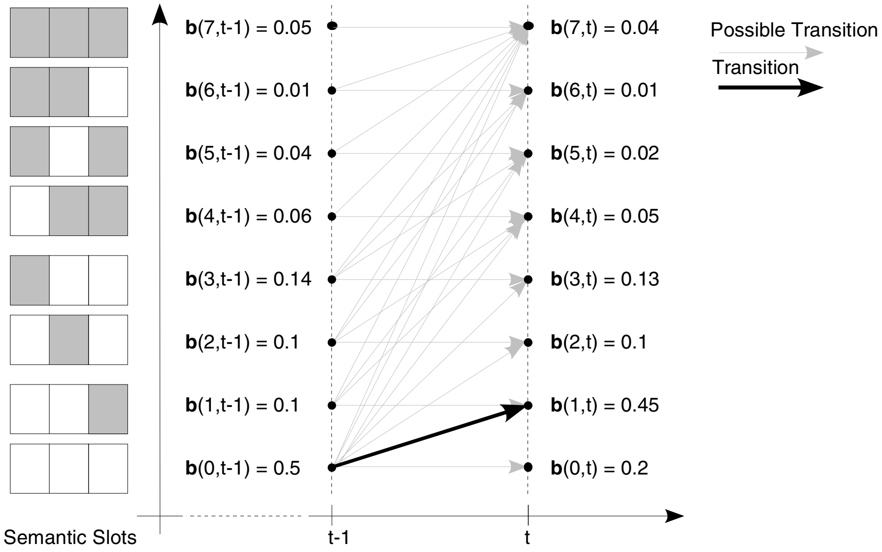

Fig. 6. Trellis representation of the semantic slots and their machine learned state transitions

The Viterbi algorithm computes the best path through the trellis diagram [9]. To that end the belief state **b**(t) describes the probability of one path at time t, which ends in state **s**(t). The Viterbi algorithm is defined as follows:

Initialization:

$$\begin{matrix} \delta_1(i) = p(b_i|w_1) \\ \psi_1(i) = 0 \end{matrix} \quad , 1 \leq i \leq N. \tag{2}$$

Recursion:

$$\begin{matrix} \delta_t(j) = \max\limits_{1\leq i\leq N}[\delta_{t-1}(i)\cdot a_{ij}]\cdot p(b_j|w_t) \\ \psi_t(j) = {}_{1\leq i\leq N}[\delta_{t-1}(i)\cdot a_{ij}]\cdot p(b_j|w_t) \end{matrix} \; , \; \begin{matrix} 2 \leq t \leq T \\ 1 \leq j \leq N \end{matrix} \tag{3}$$

Termination:

$$\hat{q}_T = \underset{1\leq i\leq N}{\operatorname{argmax}}\, \delta_T(i) \tag{4}$$

The best path is then found by backtracking:

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}) \tag{5}$$

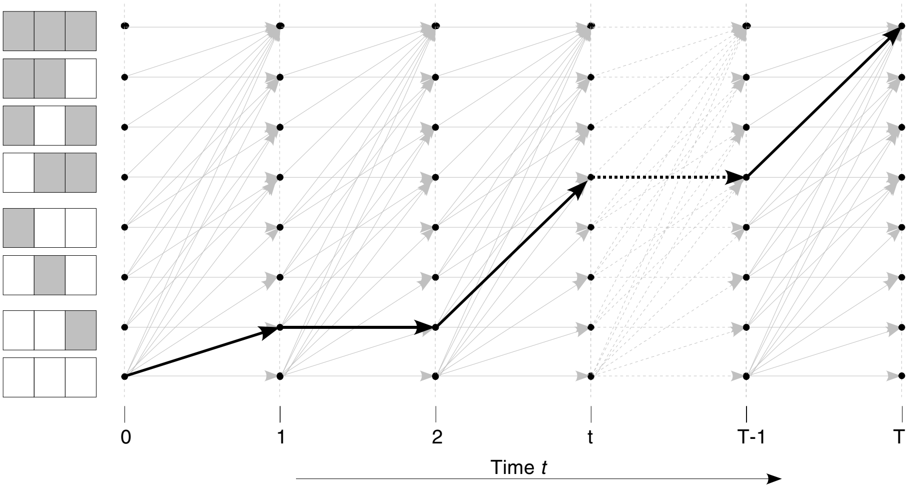

Fig. 7. The dialog progression through the user goals is described by the Viterbi algorithm

4 Corpus Description

This speech database is the first in a series of recordings of natural speech, in the Air Travel Information System (ATIS) domain. Queries collected for these corpora are spoken, without scripts or other constraints to ATIS [3].

Table 1. Example dialog between human and system derived from the ATIS Corpora

Speaker	Dialogue and Actions
Human	Hello.
System	What can I do for you?
Human	Show me all the nonstop flights from Atlanta to Philadelphia.
System	Lists/List of flights from cities whose city name is Atlanta and to cities whose city name is Philadelphia and whose stops is 0.
Human	Yes, I would like some information on the flights on April 22nd, evening.
System	Lists/List flights from cities whose city name is Dallas and to cities whose city name is Denver and whose departure time is between 1645 and 1715 and flying on flight days whose day name is Sunday.
System	Can't find any result.

A human wizard simulating the speech recognizer of the future gives the impression of a speech-recognizing computer system. ATIS contains 21650 user utterances and 2195 sessions. For each session, a problem was posed to a subject, such as “find the cheapest way to fly from Atlanta to Dallas by next Thursday”. The subject's queries by SQL to the computer system and the computer system's responses were saved as data.

The average dialog length is about eight question and answer steps. A dialog example is shown in Table 1.

5 Experiments and Results

We analyse the length of a dialog with 200 automatically generated and completely disjoint test-sentences. The experiments show reliability of our system; all dialogs reach the dialog goal with their users goals (stopover). In 66.5% of the tested dialogs, we achieve exactly the same user goals as described in the ATIS corpus. Hence, 33.5% of the dialogs have over-length. The system achieves more user goals than necessary to create a SQL statement to the database. To that end the user is confronted with more questions than required by our approach.

There are on average 6 dialog steps necessary for the achievement of the dialog goal. Our system requires about 8 dialog steps before a SQL statement is sent to the database.

6 Conclusion and Future Work

In this work a novel mixed-initiative dialog management system based on Graphical Models has been presented. Information slots are introduced and modeled within the Graphical Model to a DBN. Their parameters have been learned from naturally spoken sentences in the ATIS task. The Viterbi algorithm computes iteratively the best path through the trellis. Our approach achieves all dialog goals in 200 test sentences, but 33.5% of the tested dialogs have over-length.

However, the first results show, that the system works reliable. In future we plan to analyse, how the model could recognize errors in the information slots and how to correct them.

References

1. Baum, L.E., Petrie, T.: Statistical Inference for Probabilistic Functions of Finite State Markov Chains. The Annals of Mathematical Statistics 37, 1554–1563 (1966)
2. Bilmes, J., Zweig, G.: The Graphical Model Toolkit: An Open Source Software System for Speech and Time-Series Processing. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2002)
3. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The ATIS Spoken Language Systems Pilot Corpus (1990), http://acl.ldc.upenn.edu/H/H90/H90-1021.pdf
4. Larrson, S., Bernman, A., Hallenborg, J., Hjelm, D.: Trindikit Manual (2004)
5. Levin, E., Pieraccini, R., Eckert, W.: Using Markov Decision Processes For Learning Dialogue Strategies. In: Proc. Int. Conf. Acoustics, Speech and Signal Processing, Seattle, USA (1998)
6. McTear, M.F.: Spoken Dialogue Technology. Springer, London (2004)
7. Rieser, V., Lemon, O.: Using Machine Learning to Explore Human Multimodal Clarification Strategies. In: IEEE/ACL Workshop, Palm Beach, Aruba (2006)
8. Schwärzler, S., Geiger, J., Schenk, J., Al-Hames, M., Hörnler, B., Ruske, G., Rigoll, G.: Combining Statistical and Syntactical Systems for Spoken Language Understanding With Graphical Models. In: Proc. of the 9th International Speech Communication Association (Interspeech 2008), Brisbane, Australia (2008)

9. Viterbi, A.: Error Bounds for Convolutional Codes and an Asymptotically OptimumError Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Transactions on Information Theory, series 13, 260–267 (1967)
10. W3C Recommendation, Voice Extensible Markup Language (VXML), Version 2.0 (2004), http://www.w3.org/TR/2004/REC-voicexml20-20040316
11. Young, S.: Using POMDPs for dialog management. In: IEEE/ACL Workshop, Palm Beach, Aruba (2006)