

# Robust Dialog Management with N-best Hypotheses Using Dialog Examples and Agenda

Cheongjae Lee, Sangkeun Jung and Gary Geunbae Lee

Pohang University of Science and Technology  
Department of Computer Science and Engineering  
Pohang, Republic of Korea  
{lcj80, hugman, gblee}@postech.ac.kr

## Abstract

This work presents an agenda-based approach to improve the robustness of the dialog manager by using dialog examples and n-best recognition hypotheses. This approach supports n-best hypotheses in the dialog manager and keeps track of the dialog state using a discourse interpretation algorithm with the agenda graph and focus stack. Given the agenda graph and n-best hypotheses, the system can predict the next system actions to maximize multi-level score functions. To evaluate the proposed method, a spoken dialog system for a building guidance robot was developed. Preliminary evaluation shows this approach would be effective to improve the robustness of example-based dialog modeling.

## 1 Introduction

Development of spoken dialog systems involves human language technologies which must cooperate to answer user queries. Since the performance in human language technologies such as Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU)<sup>1</sup> have been improved, this advance has made it possible to develop spoken dialog systems for many different application domains.

Nevertheless, there are major problems for practical spoken dialog systems. One of them which must be considered by the Dialog Manager (DM) is the error propagation from ASR and NLU modules. In

<sup>1</sup>Through this paper, we will use the term *natural language* to include both *spoken language* and *written language*

general, errors in spoken dialog systems are prevalent due to errors in speech recognition or language understanding. These errors can cause the dialog system to misunderstand a user and in turn lead to an inappropriate response. To avoid these errors, a basic solution is to improve the accuracy and robustness of the recognition and understanding processes. However, it has been impossible to develop perfect ASR and NLU modules because of noisy environments and unexpected input. Therefore, the development of robust dialog management has also been one of the most important goals in research on practical spoken dialog systems.

In the dialog manager, a popular method to deal with these errors is to adopt dialog mechanisms for detecting and repairing potential errors at the conversational level (McTear et al., 2005; Torres et al., 2005; Lee et al., 2007). In human-computer communication, the goal of error recovery strategy is to maximize the user's satisfaction of using the system by guiding for the repair of the wrong information by human-computer interaction. On the other hand, there are different approaches to improve the robustness of dialog management using n-best hypotheses. Rather than Markov Decision Processes (MDPs), partially observable MDPs (POMDPs) potentially provide a much more powerful framework for robust dialog modeling since they consider n-best hypotheses to estimate the distribution of the belief state (Williams and Young, 2007).

In recent, we proposed another data-driven approach for the dialog modeling called Example-based Dialog Modeling (EBDM) (Lee et al., 2006a). However, difficulties occur when attempting to de-

ploy EBDM in practical spoken dialog systems in which ASR and NLU errors are frequent. Thus, this paper proposes a new method to improve the robustness of the EBDM framework using an agenda-based approach and n-best recognition hypotheses. We consider a domain-specific agenda to estimate the best dialog state and example because, in task-oriented systems, a current dialog state is highly correlated to the previous dialog state. We have also used the example-based error recovery approach to handle exceptional cases due to noisy input or unexpected focus shift.

This paper is organized as follows. Previous related work is described in Section 2, followed by the methodology and problems of the example-based dialog modeling in Section 3. An agenda-based approach for heuristics is presented in Section 4. Following that, we explain greedy selection with n-best hypotheses in Section 5. Section 6 describes the error recovery strategy to handle unexpected cases. Then, Section 7 provides the experimental results of a real user evaluation to verify our approach. Finally, we draw conclusions and make suggestions for future work in Section 8.

## 2 Related Work

In many spoken dialog systems that have been developed recently, various knowledge sources are used. One of the knowledge sources, which are usually application-dependent, is an agenda or task model. These are powerful representations for segmenting large tasks into more reasonable subtasks (Rich and Sidner, 1998; Bohus and Rudnicky, 2003; Young et al., 2007). These are manually designed for various purposes including dialog modeling, search space reduction, domain knowledge, and user simulation.

In Collagen (Rich and Sidner, 1998), a plan tree, which is an approximate representation of a partial SharedPlan, is composed of alternating act and plan recipe nodes for internal discourse state representation and discourse interpretation.

In addition, Bohus and Rudnicky (2003) have presented a RavenClaw dialog management which is an agenda-based architecture using hierarchical task decomposition and an expectation agenda. For modeling dialog, the domain-specific dialog control is represented in the *Dialog Task Specification* layer

using a tree of dialog agents, with each agent handling a certain subtask of the dialog task.

Recently, the problem of a large state space in POMDP framework has been solved by grouping states into partitions using user goal trees and ontology rules as heuristics (Young et al., 2007).

In this paper, we are interested in exploring algorithms that would integrate this knowledge source for users to achieve domain-specific goals. We used an agenda graph whose hierarchy reflects the natural order of dialog control. This graph is used to both keep track of the dialog state and to select the best example using multiple recognition hypotheses for augmenting previous EBDM framework.

## 3 Example-based Dialog Modeling

Our approach is implemented based on Example-Based Dialog Modeling (EBDM) which is one of generic dialog modelings. We begin with a brief overview of the EBDM framework in this section. EBDM was inspired by Example-Based Machine Translation (EBMT) (Nagao, 1984), a translation system in which the source sentence can be translated using similar example fragments within a large parallel corpus, without knowledge of the language's structure. The idea of EBMT can be extended to determine the next system actions by finding similar dialog examples within the dialog corpus. The system action can be predicted by finding semantically similar user utterances with the dialog state. The dialog state is defined as the set of relevant internal variables that affect the next system action. EBDM needs to automatically construct an example database from the dialog corpus. Dialog Example DataBase (DEDDB) is semantically indexed to generalize the data in which the indexing keys can be determined according to state variables chosen by a system designer for domain-specific applications (Figure 1). Each turn pair (user turn, system turn) in the dialog corpus is mapped to semantic instances in the DEDDB. The index constraints represent the state variables which are domain-independent attributes. To determine the next system action, there are three processes in the EBDM framework as follows:

- **Query Generation:** The dialog manager makes Structured Query Language (SQL)

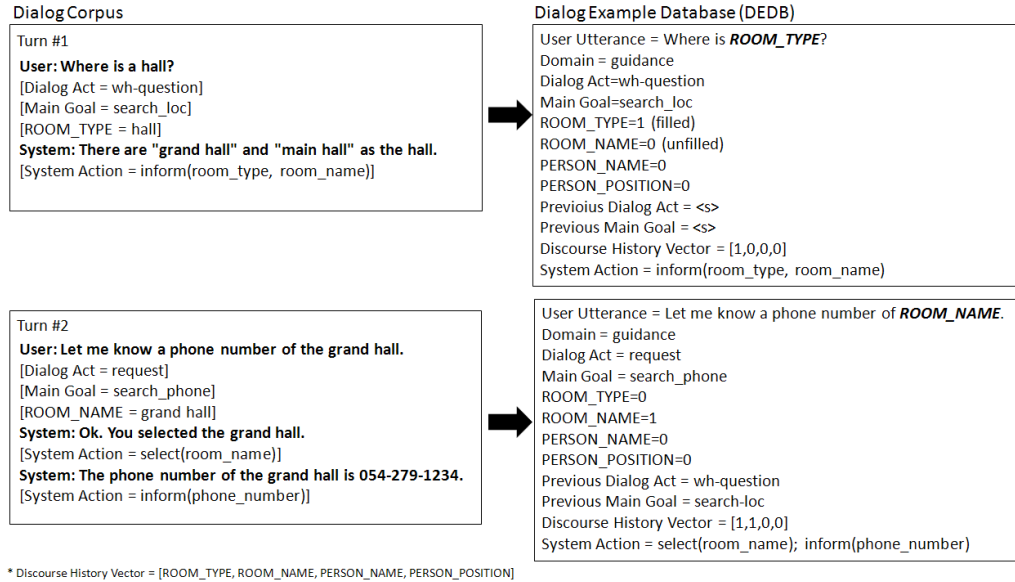


Figure 1: Indexing scheme for dialog example database on building guidance domain

statement using discourse history and NLU results.

- **Example Search:** The dialog manager searches for semantically similar dialog examples in the DEDB given the current dialog state. If no example is retrieved, some state variables can be ignored by relaxing particular variables according to the level of importance given the dialog's genre and domain.
- **Example Selection:** The dialog manager selects the best example to maximize the utterance similarity measure based on lexico-semantic similarity and discourse history similarity.

Figure 2 illustrates the overall strategy of EBDM framework for spoken dialog systems. The EBDM framework is a simple and powerful approach to rapidly develop natural language interfaces for multi-domain dialog processing (Lee et al., 2006b). However, in the context of spoken dialog system for domain-specific tasks, this framework must solve two problems: (1) Keeping track of the dialog state with a view to ensuring steady progress towards task completion, (2) Supporting n-best recognition hypotheses to improve the robustness of dialog manager. Consequently, we sought to solve these prob-

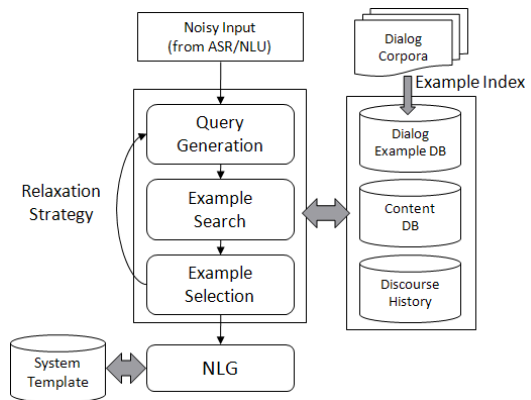


Figure 2: Strategy of the Example-Based Dialog Modeling (EBDM) framework.

lems by integrating the agenda graph as a heuristic which reflects the natural hierarchy and order of subtasks needed to complete the task.

## 4 Agenda Graph

In this paper, agenda graph  $G$  is simply a way of encoding the domain-specific dialog control to complete the task. An agenda is one of the subtask flows, which are possible paths from root node to terminal node.  $G$  is composed of nodes ( $v$ ) which correspond to possible intermediate steps in the process of completing the specified task, and edges ( $e$ ) which con-

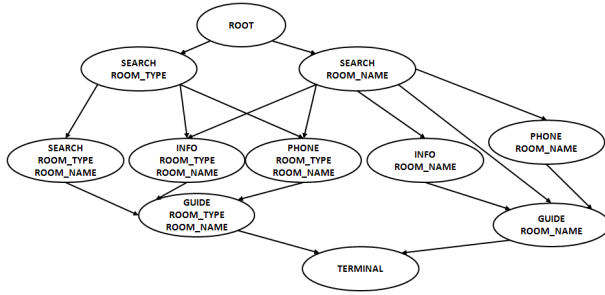


Figure 3: Example of an agenda graph for a building guidance.

nect nodes. In other words,  $v$  corresponds to user goal state to achieve domain-specific subtask in its expected agenda. Each node includes three different components: (1) A precondition that must be true before the subtask is executed; (2) A description of the node that includes its label and identifier; and (3) Links to nodes that will be executed at the subsequent turn. For every edge  $e_{ij} = (v_i, v_j)$ , we defined a transition probability based on prior knowledge of dialog flows. This probability can be assigned based on empirical analysis of human-computer conversations, assuming that the users behave in consistent, goal-directed ways. Alternatively, it can be assigned manually at the discretion of the system developer to control the dialog flow. This heuristic has advantages for practical spoken dialog system because a key condition for successful task-oriented dialog system is that the user and system know which task or subtask is currently being executed. To exemplify, Figure 3 illustrates part of the agenda graph for *PHOPE*, a building guidance robot using the spoken dialog system. In Figure 3,  $G$  is represented by a Directed Acyclic Graph (DAG), where each link in the graph reflects a transition between one user goal state and the next. The set of paths in  $G$  represent an agenda designed by the system developer. We adapted DAG representation because it is more intuitive and flexible than hierarchical tree representation. The syntax for graph representation in our system is described by an XML schema (Figure 4).

#### 4.1 Mapping Examples to Nodes

In the agenda graph  $G$ , each node  $v$  should hold relevant dialog examples corresponding to user goal states. Therefore, the dialog examples in DEDB are

```

<agenda_graph domain="guidance">
  <node node_id="2">
    <property>
      <label>Search Location with Room Type and Room Name</label>
    </property>
    <precondition>
      <main_goal>SEARCH_LOC</main_goal>
      <slot_status name="LOC_ROOM_NAME">1</slot_status>
      <slot_status name="LOC_ROOM_TYPE">1</slot_status>
      <slot_status name="LOC_ROOM_NUMBER">0</slot_status>
    </precondition>
    <next>
      <node_id prob="0.15">4</node_id>
      <node_id prob="0.25">5</node_id>
      <node_id prob="0.60">6</node_id>
    </next>
  </node>
</agenda_graph>
  
```

Figure 4: XML description for the agenda graph

mapped to a user goal state when a precondition of the node is true. Initially, the root node of the DAG is the starting state, where there is no dialog example. Then, the attributes of each dialog example are examined via the preconditions of each user goal node by breadth-first traversal. If the precondition is true, the node holds relevant that may appear in the user's goal state. The method of selecting the best of these examples will be described in 5.

#### 4.2 Discourse Interpretation

Inspired by Collagen (Rich and Sidner, 1998; Lesh et al., 2001), we investigated a discourse interpretation algorithm to consider how the current user's goal can contribute to the current agenda in a focus stack according to Lochbaum's discourse interpretation algorithm (Lochbaum, 1998). The focus stack takes into account the discourse structure by keeping track of discourse states. In our system, the focus stack is a set of user goal nodes which lead to completion of the subtask. The top on the focus stack is the previous node in this set. The focus stack is updated after every utterance. To interpret the type of the discourse state, this breaks down into five main cases of possible current node for an observed user's goal:

- *NEW\_TASK*: Starting a new task to complete a new agenda (Child of the root).
- *NEW\_SUB\_TASK*: Starting a new subtask to partially shift focus (A different child of the parent).

- *NEXT\_TASK*: Working on the next subtask contributing to current agenda (Its child node).
- *CURRENT\_TASK*: Repeating or modifying the observed goal on the current subtask (Current node).
- *PARENT\_TASK*: Modifying the observation on the previous subtask (Parent node).

Nodes in parentheses denote the topological position of the current node relative to the top node on the focus stack. If *NEXT\_TASK* is selected, the current node is pushed to the focus stack. *NEXT\_TASK* covers totally focused behavior, i.e., when there are no unexpected focus shifts. This occurs when the current user utterance is highly correlated to the previous system utterance. The remaining four cases cover various types of discourse state. For example, *NEW\_SUB\_TASK* involves starting a new subtask to partially shift focus, thereby popping the previous goal off the focus stack and pushing a new user goal for the new subtask. *NEW\_TASK*, which is placed on the node linked to root node, involves starting a new task to complete a new agenda. Therefore, a dialog is re-started and the current node is pushed onto the focus stack with the current user goal as its first element.

If none of the above cases holds, the discourse interpretation concludes that the current input should be rejected because we expect user utterances to be correlated to the previous turn in a task-oriented domain. Therefore, this interpretation does not contribute to the current agenda on the focus stack due to ASR and NLU errors that are due to noisy environments and unexpected input. These cases can be handled by using an error recovery strategy in Section 6.

Figure 5 shows some examples of pseudo-codes used in the discourse interpretation algorithm to select the best node among possible next nodes.  $S, H,$  and  $G$  denote the focus stack, hypothesis, and agenda graph, respectively. The **INTERPRET** algorithm is initially called to interpret the current discourse state. Furthermore, the essence of a discourse interpretation algorithm is to find candidate nodes of possible next subtask for an observed user goal, expressed in the definition of **GENERATE**. The **SELECT** algorithm selects the best node to maximize

<pre> INTERPRET(&lt;S,H&gt;,G) ≡ C ← GENERATE(&lt;S,H&gt;,G) if  C  = 1   then return discourse state in C else   c* ← SELECT(&lt;S,H&gt;,C)   return selected discourse state in c* </pre>	<pre> GENERATE(&lt;S,H&gt;,G) ≡ return a union set of : i) NEW_TASK(&lt;S,H&gt;,G) ii) NEW_SUB_TASK(&lt;S,H&gt;,G) iii) NEXT_TASK (&lt;S,H&gt;,G) iv) CURRENT_TASK(&lt;S,H&gt;,G) v) PARENT_TASK(&lt;S,H&gt;,G) </pre>
<pre> NEXT_TASK(&lt;S,H&gt;,G) ≡ C ← ∅ E ← retrieved examples of H foreach e ∈ E   foreach c<sub>s</sub> ∈ {children of top(S) G}     if e is an example of c<sub>s</sub>       then C ← C ∪ {c<sub>s</sub>} return C </pre>	<pre> SELECT(&lt;S,H&gt;,C) ≡ c* = argmax<sub>c</sub> ωS<sub>H</sub>(H) + (1-ω) S<sub>D</sub>(c ∈ C S) return c* </pre>

Figure 5: Pseudo-codes for the discourse interpretation algorithm

the score function based on current input and discourse structure given the focus stack. The details of how the score of candidate nodes are calculated are explained in Section 5.

## 5 Greedy Selection with n-best Hypotheses

Many speech recognizers can generate a list of plausible hypotheses (n-best list) but output only the most probable one. Examination of the n-best list reveals that the best hypothesis, the one with the lowest word error rate, is not always in top-1 position but sometimes in the lower rank of the n-best list. Therefore, we need to select the hypothesis that maximizes the scoring function among a set of n-best hypotheses of each utterance. The role of agenda graph is for a heuristic to score the discourse state to successfully complete the task given the focus stack.

The current system depends on a greedy policy which is based on immediate transitions rather than full transitions from the initial state. The greedy selection with n-best hypotheses is implemented as follows. Firstly, every hypothesis  $h_i$  is scanned and all possible nodes are generated using the discourse interpretation. Secondly, the multi-level score functions are computed for each candidate node  $c_i$  given a hypothesis  $h_i$ . Using the greedy algorithm, the node with the highest score is selected as the user goal state. Finally, the system actions are predicted by the dialog example to maximize the example score in the best node.

The generation of candidate nodes is based on multiple hypotheses from the previous EBDM

framework. This previous EBDM framework chose a dialog example to maximize the utterance similarity measure. However, our system generates a set of multiple dialog examples with each utterance similarity over a threshold given a specific hypothesis. Then, the candidate nodes are generated by matching to each dialog example bound to the node. If the number of matching nodes is exactly one, that node is selected. Otherwise, the best node which would be pushed onto the focus stack must be selected using multi-level score functions.

### 5.1 Node Selection

The node selection is determined by calculating some score functions. We defined multi-level score functions that combine the scores of ASR, SLU, and DM modules, which range from 0.00 to 1.00. The best node is selected by greedy search with multiple hypotheses  $H$  and candidate nodes  $C$  as follows:

$$c^* = \arg \max_{h_i \in H, c_i \in C} \omega S_H(h_i) + (1 - \omega) S_D(c_i|S)$$

where  $H$  is a list of n-best hypotheses and  $C$  is a set of nodes to be generated by the discourse interpretation. For the node selection, we divided the score function into two functions  $S_H(h_i)$ , hypothesis score, and  $S_D(c_i|S)$ , discourse score, where  $c_i$  is the focus node to be generated by single hypothesis  $h_i$ .

We defined the hypothesis score at the utterance level as

$$S_H(h_i) = \alpha S_{rec}(h_i) + \beta S_{cont}(h_i)$$

where  $S_{rec}(h_i)$  denotes the recognition score which is a generalized confidence score over the confidence score of the top-rank hypothesis.  $S_{cont}(h_i)$  is the content score in the view of content management to access domain-specific contents. For example, in the building guidance domain, these contents would be a building knowledge database including room name, room number, and room type. The score is defined as:

$$S_{cont}(h_i) = \begin{cases} \frac{N(C_{h_i})}{N(C_{prev})} & \text{if } C_{h_i} \subseteq C_{prev} \\ \frac{N(C_{h_i})}{N(C_{total})} & \text{if } C_{h_i} \not\subseteq C_{prev} \end{cases}$$

where  $C_{prev}$  is a set of contents at the previous turn and  $C_{total}$  is a set of total contents in the content

database.  $C_{h_i}$  denotes a set of focused contents by hypothesis  $h_i$  at the current turn.  $N(C)$  represents the number of contents  $C$ . This score reflects the degree of content coherence because the number of contents of interest has been gradually reduced without any unexpected focus shift. In the hypothesis score,  $\alpha$  and  $\beta$  denote weights which depend on the accuracy of speech recognition and language understanding, respectively.

In addition to the hypothesis score, we defined the discourse score  $S_D$  at the discourse level to consider the discourse structure between the previous node and current node given the focus stack  $S$ . This score is the degree to which candidate node  $c_i$  is in focus with respect to the previous user goal and system utterance. In the agenda graph  $G$ , each transition has its own probability as prior knowledge. Therefore, when  $c_i$  is *NEXT\_TASK*, the discourse score is computed as

$$S_D(c_i|S) = P(c_i|c = top(S))$$

where  $P(c_i|c = top(S))$  is a transition probability from the top node  $c$  on the focus stack  $S$  to the candidate node  $c_i$ . However, there is a problem for cases other than *NEXT\_TASK* because the graph has no backward probability. To solve this problem, we assume that the transition probability may be lower than that of the *NEXT\_TASK* case because a user utterance is likely to be influenced by the previous turn. Actually, when using the task-oriented dialog system, typical users stay focused most of the time during imperfect communication (Lesh et al., 2001). To assign the backward transition probability, we obtain the minimum transition probability  $P_{min}(S)$  among from the top node on the focus stack  $S$  to its children. Then, the discourse score  $S_D$  can be formalized when the candidate node  $c_i$  does not correspond to *NEXT\_TASK* as follows:

$$S_D(c_i|S) = \max\{P_{min}(S) - \lambda Dist(c_i, c), 0\}$$

where  $\lambda$  is a penalty of distance between candidate node and previous node,  $Dist(c_i, c)$ , according to type of candidate node such as *NEW\_TASK* and *NEW\_SUB\_TASK*. The simplest case is to uniformly assign  $\lambda$  to a specific value.

To select the best node using the node score, we use  $\omega$  ( $0 \leq \omega \leq 1$ ) as an interpolation weight

between the hypothesis score  $S_h$  and the discourse score  $S_D$ . This weight is empirically assigned according to the characteristics of the dialog genre and task. For example,  $\omega$  can set lower to manage the transactional dialog in which the user utterance is highly correlated to the previous system utterance, i.e., a travel reservation task, because this task usually has preference orders to fill slots.

## 5.2 Example Selection

After selecting the best node, we use the example score to select the best dialog example mapped into this node.

$$e^* = \arg \max_{e_j \in E(c^*)} \omega S_{utter}(h^*, e_j) + (1 - \omega) S_{sem}(h^*, e_j)$$

where  $h^*$  is the best hypothesis to maximize the node score and  $e_j$  is a dialog example in the best node  $c^*$ .  $S_{utter}(h, e_j)$  denotes the value of the utterance similarity of the user's utterances between the hypothesis  $h$  and dialog example  $e_j$  in the best node  $c^*$  (Lee et al., 2006a).

To augment the utterance similarity used in the EBDM framework, we also defined the semantic score for example selection,  $S_{sem}(h, e_j)$ :

$$S_{sem}(h, e_j) = \frac{\# \text{ of matching index keys}}{\# \text{ of total index keys}}$$

The semantic score is the ratio of matching index keys to the number of total index keys between hypothesis  $h$  and example record  $e_j$ . This score reflects that a dialog example is semantically closer to the current utterance if the example is selected with more index keys. After processing of the node and example selection, the best example is used to predict the system actions. Therefore, the dialog manager can predict the next actions with the agenda graph and n-best recognition hypotheses.

## 6 Error Recovery Strategy

As noted in Section 4.2, the discourse interpretation sometimes fails to generate candidate nodes. In addition, the dialog manager should confirm the current information when the score falls below some threshold. For these cases, we adapt an example-based error recovery strategy (Lee et al., 2007). In this approach, the system detects that something is

wrong in the user's utterance and takes immediate steps to address the problem using some help messages such as *UtterHelp*, *InfoHelp*, and *UsageHelp* in the example-based error recovery strategies. We also added a new help message, *AgendaHelp*, that uses the agenda graph and the label of each node to tell the user which subtask to perform next such as "SYSTEM: Next, you can do the subtask 1)Search Location with Room Name or 2)Search Location with Room Type".

## 7 Experiment & Result

First we developed the spoken dialog system for *PHOPE* in which an intelligent robot can provide information about buildings (i.e., room number, room location, room name, room type) and people (i.e., name, phone number, e-mail address, cellular phone number). If the user selects a specific room to visit, then the robot takes the user to the desired room. For this system, ten people used the WOZ method to collect a dialog corpus of about 500 utterances from 100 dialogs which were based on a set of pre-defined 10 subjects relating to domain-specific tasks. Then, we designed an agenda graph and integrated it into the EBDM framework.

In an attempt to quantify the impact of our approach, five Korean users participated in a preliminary evaluation. We provided them with pre-defined scenarios and asked them to collect test data from 50 dialogs, including about 150 utterances. After processing each dialog, the participants completed a questionnaire to assess their satisfaction with aspects of the performance evaluation. The speech recognition hypotheses are obtained by using the Hidden Markov model Toolkit (HTK) speech recognizer adapted to our application domain in which the word error rate (WER) is 21.03%. The results of the *Task Completion Rate* (TCR) are shown in Table 1. We explored the effects of our agenda-based approach with n-best hypotheses compared to the previous EBDM framework which has no agenda graph and supports only 1-best hypothesis.

Note that using 10-best hypotheses and the agenda graph increases the TCR from 84.0% to 90.0%, that is, 45 out of 50 dialogs were completed successfully. The average number of turns ( $\#AvgTurn$ ) to completion was also shorter, which

shows 4.35 turns per a dialog using the agenda graph and 10-best hypotheses. From these results, we conclude that the use of the n-best hypotheses with the agenda graph is helpful to improve the robustness of the EBDM framework against noisy inputs.

System	#AvgTurn	TCR (%)
1-best(-AG)	4.65	84.0
10-best(+AG)	4.35	90.0

Table 1: Task completion rate according to using the AG (Agenda Graph) and n-best hypotheses for n=1 and n=10.

## 8 Conclusion & Discussion

This paper has proposed a new agenda-based approach with n-best recognition hypotheses to improve the robustness of the Example-based Dialog Modeling (EBDM) framework. The agenda graph can be thought of as a hidden cost of applying our methodology. However, an explicit agenda is necessary to successfully achieve the purpose of using spoken dialog system. Our preliminary results indicate this fact that the use of agenda graph as heuristics can increase the TCR. In addition, our approach is robust to recognition errors because it maintains multiple hypotheses for each user utterance.

There are several possible subjects for further research on our approach. First, the optimal interpolation weights should be determined. This task will require larger dialog corpora by using user simulation. Second, the cost of designing the agenda graph should be reduced. We have focused on developing a system to construct this graph semi-automatically by applying dialog state clustering and utterance clustering to achieve hierarchical clustering of dialog examples. Finally, future work will include expanding our system to other applications, such as navigation systems for automobiles.

## Acknowledgement

This work was supported by grant No. RTI04-02-06 from the Regional Technology Innovation Program and by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy (MOICE) of Korea.

## References

- Bohus, B. and Rudnicky A. 2003. RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. *Proceedings of the European Conference on Speech, Communication and Technology*, 597–600.
- Grosz, B.J. and Kraus, S. 1996. Collaborative Plans for Complex Group Action. *Artificial Intelligence*, 86(2):269–357.
- Lee, C., Jung, S., Eun, J., Jeong, M., and Lee, G.G. 2006. A Situation-based Dialogue Management using Dialogue Examples. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 69–72.
- Lee, C., Jung, S., Jeong, M., and Lee, G.G. 2006. Chat and Goal-oriented Dialog Together: A Unified Example-based Architecture for Multi-domain Dialog Management. *Proceedings of the IEEE Spoken Language Technology Workshop*, 194–197.
- Lee, C., Jung, S., and Lee, G.G. 2007. Example-based Error Recovery Strategy For Spoken Dialog System. *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 538–543.
- Lesh, N., Rich, C., and Sidner, C. 2001. Collaborating with focused and unfocused users under imperfect communication. *Proceedings of the International Conference on User Modeling*, 63–74.
- Lochbaum, K.E. 1998. A Collaborative Planning Model of Intentional Structure. *Computational Linguistics*, 24(4):525–572.
- McTear, M., O’Neil, I., Hanna, P., and Liu, X. 2005. Handling errors and determining confirmation strategies-An object-based approach. *Speech Communication*, 45(3):249–269.
- Nagao, M. 1984. A Frame Work of a Mechanical Translation between Japanese and English by Analogy Principle. *Proceedings of the international NATO symposium on artificial and human intelligence*, 173–180.
- Rich, C. and Sidner, C.. 1998. Collagen: A Collaboration Agent for Software Interface Agents. *Journal of User Modeling and User-Adapted Interaction*, 8(3):315–350.
- Torres, F., Hurtado, L.F., Garcia, F., Sanchis, E., and Segarra, E. 2005. Error Handling in a Stochastic Dialog System through Confidence Measure. *Speech Communication*, 45(3):211–229.
- Williams, J.D. and Young, S. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech Language*, 21(2):393–422.
- Young, S., Schatzmann, J., Weilhammer, K., and Ye, H.. 2007. The Hidden Information State Approach to Dialog Management. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 149–152.