Recent Approaches to Dialog Management for Spoken Dialog Systems

Cheongjae Lee, Sangkeun Jung, Kyungduk Kim,
Donghyeon Lee, and Gary Geunbae Lee
Department of Computer Science and Engineering
Pohang University of Science and Technology (POSTECH)
Pohang, Republic of Korea
{lcj80,hugman,getta,semko,gblee}@postech.ac.kr

Received 4 February 2010; Revised 15 March 2010; Accepted 18 March 2010

A field of spoken dialog systems is a rapidly growing research area because the performance improvement of speech technologies motivates the possibility of building systems that a human can easily operate in order to access useful information via spoken languages. Among the components in a spoken dialog system, the dialog management plays major roles such as discourse analysis, database access, error handling, and system action prediction. This survey covers design issues and recent approaches to the dialog management techniques for modeling the dialogs. We also explain the user simulation techniques for automatic evaluation of spoken dialog systems.

Categories and Subject Descriptors: Software & Applications [**Human-Computer Interaction**]:

General Terms: Spoken Language Processing, Spoken Dialog System, User Interface

Additional Key Words and Phrases: Dialog Management, Dialog Modeling, User Simulation

1. INTRODUCTION

1.1 Overview of Spoken Dialog Systems

Spoken dialog systems can be viewed as an advanced application of spoken language technology. The objective in developing spoken dialog systems is to provide a human-centric interface for any user to access and manage information[1]. These systems are

[1] In this paper, we focus on task-based dialog systems which are designed to accomplish a well-defined task such as making a flight booking.

becoming ubiquitous due to their rapid improvement in performance and decrease in cost. The spoken dialog systems receive speech inputs from the user, and the system responds with the required action and the information. For example, a user might use a spoken dialog system to reserve a flight over the phone, to direct a robot to guide him to a specific room, or to control in-car devices such as a music player or a navigator. Since the early 1990s, many spoken dialog systems have been developed in the commercial domain to support a variety of applications in telephone-based services. For example, early spoken dialog systems functioned in restricted domains such as telephone-based call routing systems (HMIHY) [Gorin et al. 1997], weather information systems (JUPITER) [Zue et al. 2000], and travel planning (DARPA communicator) [Walker et al. 2001]. More recently developed systems are used in in-car navigation, entertainment, and communications [Minker et al. 2004; Lemon et al. 2006; Weng et al. 2006]. For example, the EU project TALK[2] focused on the development of new technologies for adaptive dialog systems using speech, graphics, or a combination of the two in the car. More recently, multi-domain dialog systems have been employed in real life situations [Allen et al. 2000; Larsson and Ericsson 2002; Lemon et al. 2002; Pakucs 2003; Komatani et al. 2006]. Such multi-domain dialog systems are now able to provide services for telematics, smart home, or intelligent robots. These systems have gradually become capable of supporting multiple tasks and of accessing information from a broad variety of sources and services.

1.2 Components of Spoken Dialog Systems

The general spoken dialog systems typically consist of the main components shown in Figure 1.

- User Input: User input is usually speech signal with noises.
- Automatic Speech Recognition (ASR): The speech signal processing first transforms a speech waveform into a sequence of parameter vectors. The speech recognition converts the sequence of parameter vectors into a textual input (e.g., a sequence of words).
- Spoken Language Understanding (SLU): The textual input of user utterance is analyzed by natural language processing (NLP) modules (e.g., morphological analysis, part-of-speech tagging, and shallow parsing). The SLU module maps the pre-processed utterance to a meaning representation (e.g., semantic frame) in

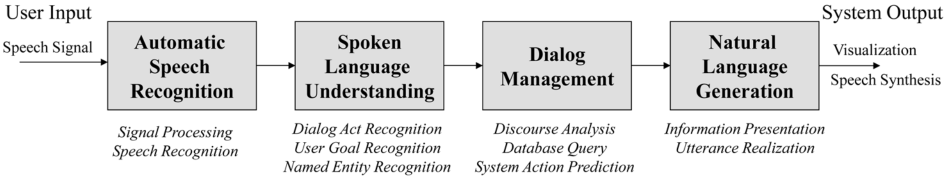


Figure 1. Traditional architecture of spoken dialog systems.

[2] TALK project site: http://www.talk-project.org

which the dialog act, user goal, and named entities are extracted by semantic parser or statistical model.

- Dialog Management (DM): This is the heart of the spoken dialog system because it coordinates the activity of all components, controls the dialog flow, and communicates with external applications. The DM should play many roles which include discourse analysis, knowledge database query, and system action prediction based on the discourse context.
- Natural Language Generation (NLG): The system responses are typically generated as natural language with a list of content items from a part of the external knowledge database (e.g., restaurant database) that answers the specific user query or request.
- System Output: The system output can be visualized on a display if available and synthesized by a text-to-speech (TTS) module or pre-recorded audio.

Among them, the DM is one of the central components within the spoken dialog systems. The major role of the DM is to select correct system actions based on observed evidences and inferred dialog states from the results of SLU (e.g., dialog act, user goal, and discourse history). In addition, the DM should be able to handle errors when the user input has ASR and SLU errors occurred by noises or unexpected inputs.

The remainder of this article is concerned with the design issues of DM and the variety of techniques and approaches developed to model dialogs. We start with a brief introduction to the role and the design issues of the dialog manager to develop the spoken dialog system in Section 2. Next, the recent approaches to model the dialogs in the dialog management are summarized in Section 3. We also explain the recent work of user simulation techniques to automatically evaluate the spoken dialog systems (Section 4). Finally, we conclude with a brief summary in Section 5.

2. DIALOG MANAGEMENT

2.1 Role of Dialog Management

In general, the DM accepts the user's intention which is represented as a semantic frame of SLU results, and outputs the system responses at a concept level. The system responses have to reflect the discourse context by maintaining the discourse history[3]. Although the roles of the DM may depend to some extent on the type of task that is involved, the key roles include (Figure 2):

- Searching and providing query results by connecting to an external knowledge database based on the current input and the discourse context
- Asking further slot information to submit an appropriate query
- Requesting to confirm unclear slot information and to rephrase if the user's input is out-of-coverage
- Predicting the next system action at the concept level to output the system's

[3] The discourse history is usually stored in many different structures depending on the DM design.

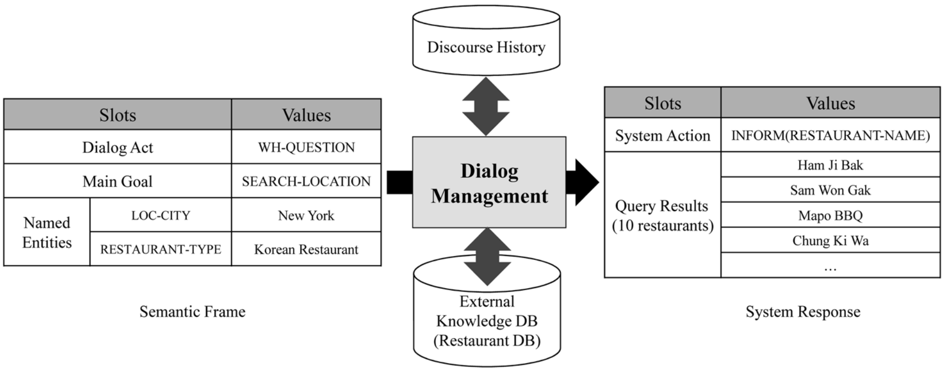


Figure 2. Role of Dialog Management.

utterance in NLG and TTS modules
- Controlling generic conversational mechanisms (e.g., barge-ins, backchannels, multi-party dialogs) in human-human dialogs to implement more human-like dialog systems

2.2 Degrees of Initiative

A dialog consists of a sequence of user and system turns which usually depend on the discourse context. The process of dialog can be viewed as an exchange of information in which the initiative may shift between the user and the system. The initiative concerns who directs the progression of the dialog. In general, the degrees of initiative in the spoken dialog system fall into one of the following strategies (Table I):

- System-initiative: The system has the initiative to guide the dialog at each step.
- User-initiative: The users takes a control of the dialogs, and the system responds to whatever the user directs.
- Mixed-initiative: The system has overall control of the dialogs. However, the users can barge in and change the dialog direction.

In a system-initiative dialog, the system usually asks one or more questions to extract some slots from the user step by step. After the slots are sufficiently filled, it can submit an appropriate query to the external knowledge database. These dialogs are generally constructed in such a way that the user's input is restricted to single words or phrases that answer the previous system prompts. A major advantage of this form of dialog control is that the user's inputs can be determined in advance because a set of vocabulary and grammar for each response can be restricted. In this way, such constraints of the search space in ASR and SLU models can significantly improve speech recognition and understanding performance. For this reason, most commercial systems are designed as system-initiative. However, the disadvantage is that it takes much time to complete more complex tasks where many slots are required to access information. In addition, the user's input is not natural because the dialog flows are predetermined with a set of limited words and phrases.

In a user-initiative dialog, the user takes control of the dialog although the system

Table I. Examples of Dialog Initiative.

Initiative Types	Example
System-initiative	System: Please state the name of the city which you are visiting. User: New York. System: Okay New York. What sort of cuisine would you like? User: Korean.
User-initiative	User: I want to go to Korean restaurant in New York. System: There are 10 Korean restaurants in New York. User: Where is Mapo BBQ?
Mixed-initiative	System: Where are you going? User: Korean restaurant in New York. System: There are 10 Korean restaurants in New York. What is the restaurant's name? User: Are there any cheap ones?

may sometimes ask confirmation questions if some slots are unclear. The user can determine the questions to be asked and the role of the system is to answer the questions. The advantage of this strategy is that it allows the user to converse with the system freely and naturally. Actually, developing a user-initiative system is more complicated than the system-initiative system because ASR and SLU should cover the relatively larger size of vocabulary and grammar, and the DM should handle more flexible dialog flows.

In a mixed-initiative dialog, the system is supposed to control dialog, but the user can have some flexibility at times to provide more information or to change the task. Therefore, mixed-initiative systems involve complex turn-taking mechanisms such as handling barge-in utterances. For example, the user can take the initiative away from the system when the user wants to ask new items which are different from the focused items. In this case, the system follows the user request, or tries to direct the user back to the original course. Recently, most advanced spoken dialog systems have tried to address this type because it looks like human-human communication. However, the certain problems should be solved to deploy this type of the dialog system in the real world. For example, barge-in utterances are more difficult to detect their boundaries and to recognize them correctly.

2.3 Error Handling

Since the performance in spoken language technologies such as ASR and SLU have been improved, spoken dialog systems can be developed now for many different application domains. Nevertheless, there are major problems for practical spoken dialog systems. One of them which must be considered by the DM is the error propagation from ASR and SLU modules. In general, errors in spoken dialog systems are prevalent due to errors in speech recognition and language understanding. The user's input may be unclear or incomplete because some or all of the words are incorrectly recognized or even though all the words are correctly recognized, the SLU module does not capture all the correct meanings due to data sparseness or ambiguity. 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 SLU modules because of noisy environments and unexpected inputs. Therefore, error handling is also an active research topic in the dialog management problems to improve the performance of the spoken dialog systems against ASR and SLU errors.

Error handling approaches in traditional DMs typically deal with these errors by adopting dialog mechanisms for detecting and repairing potential errors at the conversational level [McTear et al. 2005; Torres et al. 2005; Lee et al. 2007; Walker et al. 2000; Bohus and Rudnicky 2005]. The most commonly used measure for error detection is a confidence score computing in recognition and understanding processes [Koo et al. 2001; Hazen et al. 2002; Lo and Soong 2005]. The decision to engage this method is typically based on comparing the confidence score against the manually preset threshold. However, confidence scores are not entirely reliable and are dependent on noisy environments and user types. In addition, false acceptance, may not be easy for the user to correct the system and put the dialog back on track. Thus, it can bring some problems at the level of the DM. To address these problems, the DM can adopt some error recovery strategies (e.g., explicit/implicit confirmation and rephrasing) to repair these errors [Skantze 2005]. An explicit confirmation takes the form of a question that asks explicitly for confirmation of the target slots of the task (e.g., departure date, departure time, departure city in the flight reservation system). This may be accompanied by a request to answer with “yes”' or “no”. The DM can also use an implicit confirmation in which the system embeds in its next question a repetition of its understanding of what the user said in the response to the previous question. Explicit and implicit confirmation strategies are good to repair unreliable information by computing confidence scores. In these cases, the user says a partial phrase or a short utterance to acknowledge and confirm the target slot values. However, the deficiency of context may lead to new recognition and understanding errors. In addition, the distribution of user behaviors in coping with errors shows that users who achieve successful error recoveries use significantly more rephrasals than attempts to repair a chain of errors [Shin et al. 2002]. For these reasons, the rephrase strategies are also used to repair errors by accepting a similar utterance to the previous one. Recently, some groups have also investigated help generation mechanisms because just repeating the previous utterance cannot always correct the recognition and understanding errors [Hockey et al. 2003; Fukubayashi et al. 2006; Lee et al. 2007]. For example, dynamic help generation was developed by estimating the gap between user's mental model and the system [Fukubayashi et al. 2006]. They represent the user's mental model on the domain concept tree in which each node has known degrees that denote the degree of how much a user understands the concepts corresponding to the nodes. The grammar and the domain concept trees are manually designed by system developers to recover ASR and SLU errors. Examples of error recovery strategies are shown in Table II.

Table II. Examples of Error Recovery Strategies.

Strategy Types	Example
Explicit Confirmation	User: I want to go to Korean restaurant in New York. System: So you want to go to Korean restaurant in New York? Please answer yes or no. User: Yes.
Implicit Confirmation	User: I want to go to Korean restaurant in New York. System: What is the name of Korean restaurant in New York? User: Mapo BBQ.
Static Rephrasal	User: I want to go to Korean restaurant in New York. System: Sorry. Would you please rephrase? User: I'd like to go to Korean restaurant in New York.
Dynamic Rephrasal	User: I want to go to Korean restaurant in New York. System: Sorry. Would you please rephrase using expression “I'd like to go to [RES-TYPE] in [LOC-ADDRESS]”. User: I'd like to go to Korean restaurant in New York.

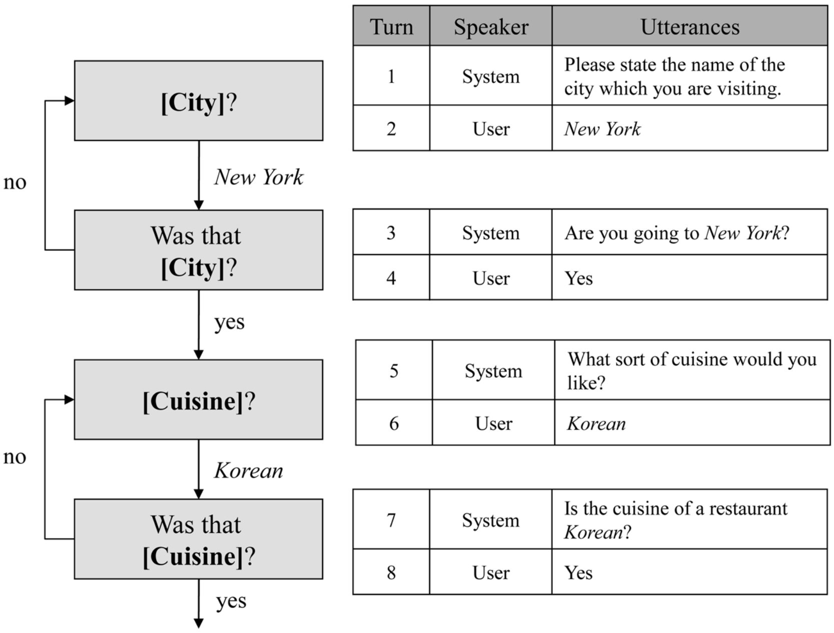
3. RECENT APPROACHES TO DIALOG MANAGEMENT

As mentioned in Section 2, the DM plays a key control role in the spoken dialog systems given the current dialog state. In practice, the DM must find the best system action to be done at the next turn by maintaining the discourse history. A number of different approaches to the DM problem have been developed to date in the community. The DM can be divided into three main approaches: (1) knowledge-based dialog management, (2) data-driven dialog management, and (3) hybrid dialog management.

3.1 Knowledge-based Dialog Management

Early dialog systems such as SUNDIAL [Peckham 1993] and ARISE [Lamel et al. 1999] were designed by application developers who have domain-specific knowledge. These systems are usually confined to both highly structured tasks and system-initiative dialogs, where a restricted and regularized language set can be expected. This knowledge-based approach generally uses finite-state automata which often involve handcrafted rules. These are dictated by the knowledge of the application, and by continuous experiments with real users. It has been used for rapid prototyping of dialog systems for strong-typed interactions with clearly-defined structures and goals [McTear 1998] (Figure 3). This approach has also been deployed in many practical applications because of its simplicity. However, hand-crafting rules in advance is difficult, and its flow is inflexible. For example, if the users provide more information that was requested by the system's question (over-informative), the system cannot manage the dialog flow because it was not designed in such a case. It also suffers from poor domain portability: when the designers develop a new application for a different domain, the entire design process must be restarted from the beginning.

To overcome these limitations, several groups [Rich and Sidner 1998; Bohus and Rudnicky 2003; Bui et al. 2004; Larsson and Traum 2006] have explored generic



Turn	Speaker	Utterances
1	System	Please state the name of the city which you are visiting.
2	User	*New York*
3	System	Are you going to *New York*?
4	User	Yes
5	System	What sort of cuisine would you like?
6	User	*Korean*
7	System	Is the cuisine of a restaurant *Korean*?
8	User	Yes

Figure 3. Example of dialog graph for restaurant information system.

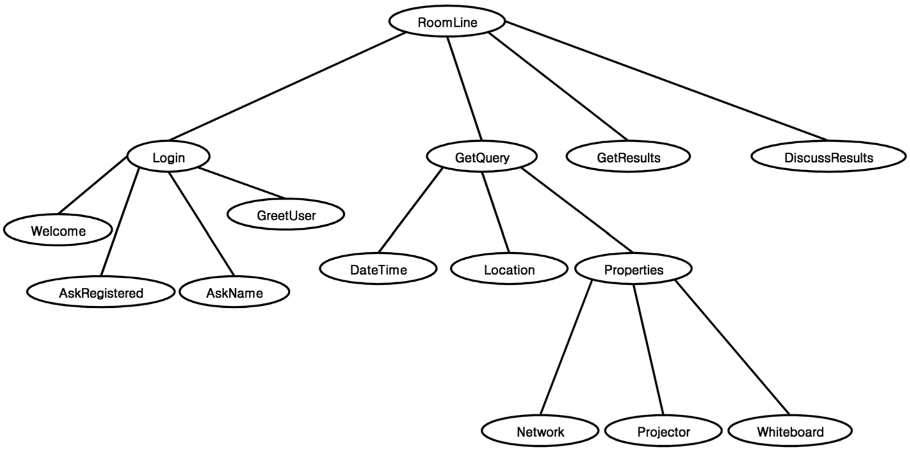


Figure 4. A portion of dialog task specification in RoomLine system.

dialog modeling approaches based on agenda or task models, which are powerful representations for segmenting large tasks into smaller and more easily handled subtasks. Several extensions are being investigated by using this approach. For example, RavenClaw is one of the most popular dialog management frameworks based on the agenda-based approach [Bohus and Rudnicky 2009]. The RavenClaw is a two-tier dialog management framework that enforces a clear separation between the domain-dependent and the domain-independent aspects of the dialog control logic.

The domain-dependent aspects are captured by the dialog task specification, essentially a hierarchical tree structure for the interaction (Figure 4), provided by the system developer. A reusable, domain-independent dialog engine manages the conversation by executing the given dialog task specification. However, the design process is still time-consuming and expensive because the knowledge sources (e.g., hierarchical task structure and plan recipes) are usually designed by human experts. To address this problem, there are some approaches to automatically model prior knowledge from dialog corpus [Roy and Subramaniam 2006; Bangalore et al. 2006; Lee et al. 2009a; Griol et al. 2009]. For example, an unsupervised clustering technique was used to automatically build a domain model (or topic structure) from call transcriptions in a call routing domain [Roy and Subramaniam 2006].

3.2 Data-driven Dialog Management

More recently, the research community for DM has exploited the benefits of data-driven approaches to ASR and SLU. Although a data-driven approach requires time-consuming data annotation, the training is done automatically and requires little human supervision. In addition, new systems can be developed at only the cost of collecting new data for moving to a new domain; this requires less time and effort than the knowledge-based approach.

These advantages have motivated the development of stochastic dialog modeling using reinforcement learning (RL) based on Markov decision processes (MDPs) [Levin et al. 2000] or partially observable MDPs (POMDPs) [Williams and Young 2007][4]. These frameworks apply statistically data-driven and theoretically-principled dialog modeling to dynamically allow changes to the dialog strategy. They accomplish this by optimizing some reward or cost functions given the current dialog state using some RL algorithms. In addition, POMDP-based DMs have the robustness to ASR and SLU errors by supporting *n*-best recognition hypotheses to estimate the belief state. The belief state is a distribution over the dialog states in the absence of knowing its state exactly because the system makes only observations (e.g., ASR and SLU results) about the real world which give incomplete information about the true current state. The error handling can be easily implemented by maintaining the belief distribution with no need of special mechanism. The variables of user context (e.g., expertise and emotion) can be also naturally incorporated into the state space by the factored model. However, practical deployment of RL-based dialog systems has encountered several obstacles [Paek 2006]. For example, the optimized policy may remove control from application developers and the refinement of the dialog control is difficult. These are serious problems because the developers should have the opportunity to easily control the dialog flow in practical systems. Many researchers are solving these problems in their ongoing work [Williams and Young 2005; Lemon et al. 2006; Young et al. 2007; Thomson et al. 2008; Williams 2008b]. For example, there are some studies on how to mix traditional knowledge-based DM design with RL-based DM to reflect domain-dependent business rules and to reduce the large policy space [Lemon et al. 2006;

[4] POMDP extends MDP by removing the requirement that the system knows its current state precisely.

Williams 2008b]. However, this approach still needs improvement before it can be applied to developing practical dialog systems.

A supervised approach to DM has been developed which uses maximum likelihood estimation of a stochastic model from human-human dialog corpus [Hurtado et al. 2005]. To avoid the data sparseness problem, this approach uses dialog register (DR) representation, which is a data structure for keeping track of discourse history as dialog state sequences. The DR contains the information about slot names and slot values provided by the user throughout the previous history of the dialog.

Some example-based approaches have been presented that perform dialog modeling using prepared dialog examples [Murao et al. 2003; Inui et al. 2001; Lee et al. 2009c]. These approaches assume that the same system actions are triggered in a similar dialog state. Therefore, the next system action can be predicted when the DM finds the dialog examples having a similar dialog state to the current dialog state. Although the keyword spotting technique was commonly used to find the similar examples, the dialog examples can be semantically indexed to generalize the data using semantic constraints (e.g., dialog act, main goal, and slot-filling vector) to represent the dialog state [Lee et al. 2009c]. The best example is then selected among the candidate examples by calculating heuristic similarity measures between the current input and the example.

3.3 Hybrid Approaches to Dialog Management

Traditional RL-based DMs require a large number of dialog corpora to learn an optimal policy because of a very large state space and a very large policy space. To address this problem, user simulation techniques have been widely used to generate a large number of simulated dialogs which are generalized from limited real corpora based on specific user models [Pietquin and Dutoit 2006; Schatzmann et al. 2007]. In a recent research, a hybrid approach to integrate reinforcement and supervised learning has been also proposed to optimize dialog policies with a fixed dialog corpus [Henderson et al. 2008]. This approach can eliminate the need for a large number of dialog corpora to optimize the dialog policies in traditional RL-based dialog systems. In this approach, RL is used to optimize a measure of dialog reward, while supervised learning is used to restrict the learnt policy to the portion of the space for which data are available.

In the classical POMDP formulation, the optimization process is free to choose any action at any time. As a result, there is no obvious way to incorporate domain knowledge or constraints such as business rules. For example, it is obvious that the system should never prescribe any medicine before it has asked for the patient's symptoms in a medical domain. However, there is no direct way to consider this to the optimization process. The unifying method was proposed to constrain the set of possible actions based on conventional rules in the POMDP framework [Lemon et al. 2006; Williams 2008b]. In this approach, the optimization process runs faster and more reliably than in a classical POMDP because spurious action choices are pruned by the conventional rules.

In addition, traditional example-based DMs have critical problems in deploying practical spoken dialog systems such as both lack of prior knowledge and weakness to ASR and SLU errors. To solve these problems, a novel hybrid approach in which both dialog examples and a prior knowledge are used has been presented to improve the robustness of example-based DM [Lee et al. 2009b]. In this approach, an agenda graph as prior knowledge is one of the subtask flows to encode the domain-dependent dialog control to complete the task. This knowledge is used to predict the next system action and to handle an unexpected focus shift by keeping track of the dialog state using the agenda graph. In addition, n-best recognition hypotheses are re-ranked by reflecting heuristics for computing both utterance-level and discourse-level scores.

4. EVALUATION OF SPOKEN DIALOG SYSTEMS

Evaluation of spoken dialog systems is essential for developing and improving the systems and for assessing their performance. The quantitative evaluation metrics have been used for assessing their performance such as task completion rate for measuring dialog success (e.g., dialog success rate, actual/perceived task completion, task success rate), dialog length of measuring dialog costs (e.g., average turn length and dialog turns), and heuristic score function of measuring weighted sum of the dialog success and the dialog costs (e.g., reward, dialog score, and average score) [Walker et al. 1997; Lemon et al. 2006b].

In general, humans are employed to evaluate the systems, but employing and training human evaluators are expensive. Furthermore, qualified human users are not always immediately available. These inevitable difficulties of working with human users can cause huge delays in development and assessment of spoken dialog systems. To avoid the problems that result from using humans to evaluate systems, developers have widely used dialog simulation, in which a simulated user interacts with a system [Schatzmann et al. 2005; López-Cózar et al. 2003; Jung et al. 2009]. User simulation for spoken dialog systems involves following essential problems: 1) user intention simulation, 2) user surface simulation, and 3) error simulation.

Typical spoken dialog systems deal with the dialog between a human user and a machine. Human users utter spoken language to express their intention, which is recognized, understood and managed by ASR, SLU and DM modules. The general architecture of a user simulator is separated into two levels: user intention and utterance simulators (Figure 5). The user intention simulator accepts the discourse contexts with system intention as input and generates the next user intention. The user utterance simulator constructs a corresponding user sentence to express the given user intention. The simulated user sentence is fed to the ASR channel simulator, which then adds noise to the utterance.

This noisy utterance is passed to a dialog system which consists of SLU and DM modules. The dialog system understands the user utterance, manages the dialog and passes the system intention to the user simulator. The user simulator, ASR channel simulator and dialog system continue the conversation until the user simulator generates an end to the dialog.

4.1 User Intention Simulation

The task of user intention simulation is to generate subsequent user intentions given current discourse contexts. The intention is usually represented as abstracted user's

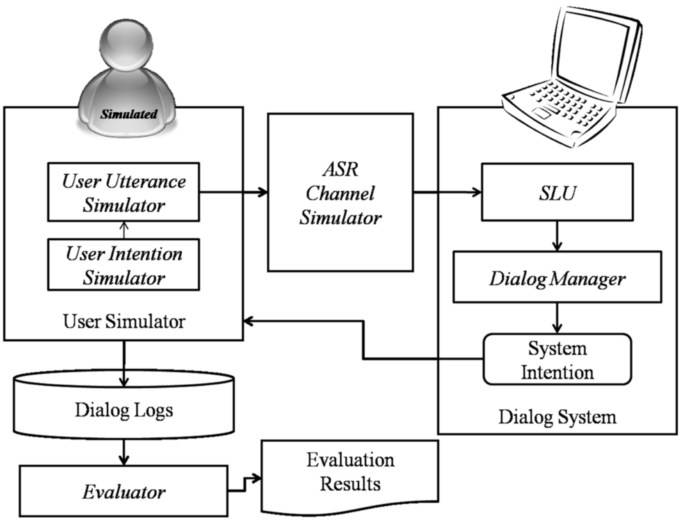


Figure 5. Overall architecture of dialog simulation.

Semantic Frame for User Inention Simulation	
raw user utterance	I want to go to city hall.
dialog_act	request
main_goal	search_loc
component.[loc_name]	cityhall

Figure 6. Example of semantic frame on car navigation domain.

goals and information on the user's utterance (surface).

Therefore general user intention simulation in spoken dialog systems takes the following probabilistic formula.

$$P(user\ intention \mid discourse\ context)$$

For example, in the case of a user simulation and spoken dialog system of [Lee et al. 2009c; Jung et al. 2009], they defined the *user intention* state S = [dialog_act, main_goal, component_slot], where dialog_act is a domain-independent label of an utterance at the level of illocutionary force (e.g. statement, request, wh_question) and main_goal is the domain-specific user goal of an utterance (e.g. give_something, tell_purpose). Component slots represent domain-specific named-entities in the utterance. For example, in the user intention state for the utterance “I want to go to City Hall” (Figure 6), the combination of each slot of semantic frame represents the state. In this example, the state symbol is ‘request+search_loc+[loc_name]’. The discourse context can be varied according to intention modeling methods.

There are two main approaches in the user intention simulator implementation: knowledge-based and data-driven approaches.

A knowledge-based intention model is built up from human discourse knowledge. It

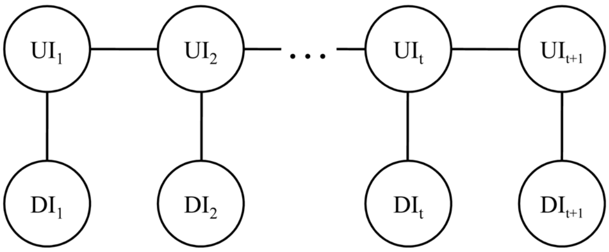


Figure 7. Conditional Random Fields for User Intention Modeling. UI_t: User Intention; DI_t: Discourse Information for the *t*th user turn.

generates correct and reasonable user response under the coverage of designed rules. In a knowledge based intention simulation approach, the developer can create different rules that determine the behavior of the simulated user given the discourse information [Chung 2004; López-Cózar et al. 2003; López-Cózar et al. 2006]. Schatzmann et al. proposed an agenda-based user simulation technique for bootstrapping a statistical dialog manager without access to training data [Schatzmann et al. 2007]. It simulates user behavior based on a compact representation of the *user goal* and a stack-like *user agenda*. It is usually hard to design all possible rules which cover the diverse circumstances caused by ASR, SLU, and DM errors. Also, designing rules for a user model from the bottom is as difficult as designing a new dialog managing model.

On the other hand, data-driven user intention modeling is relatively easy to build up if we have enough data since most of the probabilistic methods are domain- and language independent. There are several domain-independent data-driven user intention modeling methods. The pioneering work of Eckert et al. introduced a simple n-gram model for predicting the user intention [Eckert et al. 1997]. In later work of the same group, they describe how the pure bigram model can be modified to account for a more realistic degree of structure in dialog [Levin et al. 2000]. Pietquin's model [Pietquin 2004] extended Levin's model with simple representations of user goal, memory and satisfaction.

Georgila et al. proposed a linear feature combination to map from an intention state to a vector of real-valued features [Georgila et al. 2005]. Graphical model-based user intention simulation was also proposed. Cuayahuitl et al. presented a method for intention simulation based on Hidden Markov Models and Input-Output Hidden Markov Models [Cuayahuitl et al. 2005]. Another graphical model based user intention simulation method is Conditional Random Fields (CRF) [Lafferty et al. 2001] based intention model [Jung et al. 2009]. In their work, sequential behaviors of dialog participants are modeled with a linear chain between state nodes. Also arbitrary facts are captured to describe the discourse context in the form of indicator functions. In their work, the user intentions are represented as states, and the discourse information is represented as observations (Figure 7). The example of discourse context for a linear-chain CRF based intention model is illustrated in Figure 8.

Another direction of user intention simulation is taking hybrid approach. Recently

Domain Independent Features	
PREV_1_SYS_ACT	previous system action. Ex) PREV_1_SYS_ACT=inform
PREV_1_SYS_ACT_ATTRIBUTES	previous system mentioned attributes. Ex) PREV_1_SYS_ACT_attributes=city_name
PREV_2_SYS_ACT	previous system action. Ex) PREV_2_SYS_ACT=confirm
PREV_2_SYS_ACT_ATTRIBUTES	previous system mentioned attributes. Ex) PREV_2_SYS_ACT_attributes=city_name
SYSTEM_HOLDING_COMP_SLOT	system recognized component slot. Ex) SYSTEM_HOLDING_COMP_SLOT=loc_name
Domain Dependent Features	
OTHER_INFO	other useful domain dependent information Ex) OTHER_INFO(user_favorite_restaurant)=gajokjung Ex) OTHER_INFO(user_current_position)=daeidong

Figure 8. Example feature design for navigation domain.

[Jung et al. 2009] introduced a hybrid user intention modeling using Markov logic [Richardson and Domingos 2006]. They proposed a data-driven user intention modeling technique which can be diversified or personalized by integrating human discourse knowledge which is represented in first-order logic. The framework can easily and selectively integrate diverse types of user knowledge into a data-driven user intention simulation. They implemented a cooperative, corrective and self-directing users, respectively, in their framework.

Although there are no generally accepted evaluation metrics as to what constitutes a good intention model in dialog simulators, the evaluation metrics has been recently investigated to automatically determine the quality of the simulated dialogs [Schatzmann et al. 2005; Williams 2008a; Ai and Litman 2008]. For example, Schatzmann et al. provided a broad set of tests to evaluate user simulations such as bigram, Levin's, and Pietquin's models by comparing simulated and real dialogs [Schatzmann et al. 2005]. The simulated user responses were compared to real user responses in an unseen test set to assess how the generated dialogs are as natural as a real user's behavior, and the corpora of simulated dialogs were also compared to real corpora because this indicates how well the simulation covers the variety of user behavior in the training data. They presented the precision and recall measures of simulated vs. real user responses, and the distribution of turn lengths, dialog lengths, a ratio of system and user actions per dialog. In addition, some researchers proposed the rank-ordered evaluation metric based on the Cramér-von Mises divergence between the distribution of dialog scores in the real and simulated dialogs [Williams 2008a] and human judgments as the gold standard [Ai and Litman 2008].

4.2 User Utterance Simulation

A user utterance simulation technique is needed to investigate the performance of the dialog system since the simulations that are restricted to only the intention level are not sufficient to evaluate the performance of all dialog system components because the spoken dialog system is heavily influenced by the performance of the SLU as well as the DM.

User utterance simulation generates surface level utterances which express a given user intention. For example, if users want to go somewhere and provide place name information, we need to generate corresponding utterances (e.g. “I want to go to [place_name]” or “Let's go to [place_name]”). There can be many semantically equivalent sentences which express the corresponding user intention. This is formulated as follows:

$$P(\mathbf{W} \mid \textit{user intention}),$$

where $\mathbf{W}=\{w_1, w_2,...,w_N\}$ is a word (w_n) sequence.

Chung tried to use the natural language generation module of [Seneff 2002l to generate this surface [Chung 2004]. López-Cózar et al. collected real human utterances, and selected and played the voice to provide input for the spoken dialog system [López-Cózar et al. 2003]. Schatzmann et al. presented an utterance generation model based on co-occurring frequency [Schatzmann et al. 2007]. A generative maximum-likelihood model for predicting user utterance for a given user act is built by obtaining the appropriate relative frequency statistics from transcribed and annotated dialog corpus.

Recently, Jung et al. proposed a two-phase user utterance simulation method [Jung et al. 2009]. In the first phase, the process of generating structures and word sequences is iterated sufficient times to generate many different structure tags and word sequences which may occur in real human expressions. In the second phase, good user utterances are selected by the naturalness measure. In their work, to measure the naturalness of a generated utterance, the Structure and Word interpolated BLEU score (SWB) is calculated from the structural sequence BLEU score and lexical sequence BLEU score. BLEU (Bilingual Evaluation Understudy) score is widely used for automatic evaluation in statistical machine translation [Papineni et al. 2002].

4.3 ASR Channel Simulation

ASR channel simulation generates speech recognition errors which might occur in the real speech recognition process. Furthermore, the ASR channel simulator which can allow a developer to set the simulated word error rate (WER) between 0 and 1 is desirable. Therefore, the ASR channel simulation problem is generating noise-added user utterance $\mathbf{W_{noisy}}$ from a noise-free user utterance $\mathbf{W_{clean}}$ reflecting the error degree of WER.

The goal of error simulation is generating appropriate automatic speech recognition (ASR) errors or spoken language understanding errors on generated user intentions and utterances. Previous work on ASR channel simulation has investigated a number of different techniques. Some of the approaches directly set the error rate on the type of task [Pietquin and Renals 2002] and the individual speaker [Prommer et al. 2006]. The simulated word error rate can also be set to approximate the distribution found in the speech data [Georgila et al. 2005; Lemon et al. 2006b]. The ASR channel simulation based on phonetic confusions has been explored. Word sequences are mapped to phone or syllable sequences using a pronunciation dictionary and confusions are then generated using a set of probabilistic phoneme conversion rules [Deng et al. 2003], a handcrafted phone confusion matrix [Pietquin 2004] or a weighted finite state transducer [Fosler-Lussier et al. 2002; Stuttle et al. 2004].

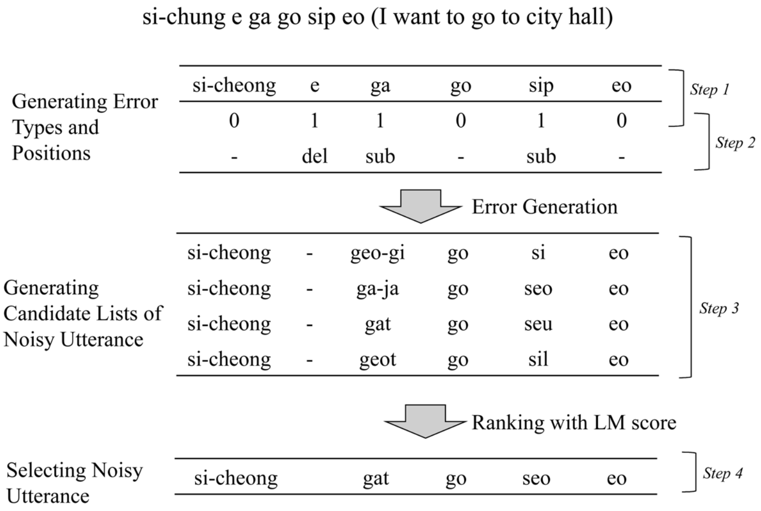

si-chung e ga go sip eo (I want to go to city hall)
Generating Error Types and Positions
si-cheong e ga go sip eo
0 1 1 0 1 0
- del sub - sub -
Step 1
Step 2
Error Generation
Generating Candidate Lists of Noisy Utterance
si-cheong - geo-gi go si eo
si-cheong - ga-ja go seo eo
si-cheong - gat go seu eo
si-cheong - geot go sil eo
Step 3
Ranking with LM score
Selecting Noisy Utterance
si-cheong gat go seo eo
Step 4

Figure 9. Example of ASR channel simulation.

Schatzmann et al. proposed ASR-confusions methods [Schatzmann et al. 2007]. In their work, erroneous utterances are generated based on word fragment-to-fragment alignment. A computationally less expensive word-level error simulation method has been suggested by [Pietquin and Dutoit 2006]. Jung et al. proposed another acoustic-linguistic knowledge based ASR channel simulation [Jung et al. 2009]. In their research, phone confusion models are used to generate ASR errors. However, an actual training corpus is not used to build confusion models. Instead the phone confusion models are built based on linguistic knowledge to implement a simple ASR channel simulator. The problem of finding an acoustic distance can be considered as an alignment problem between two sequences of symbols. In this work, the dynamic global alignment algorithm of [Needleman and Wunsch 1970] is used for both syllable and phoneme sequence alignment.

Figure 9 shows the example of an ASR channel simulation which is based on [Jung et al. 2009]'s method. The method involves four steps: 1) Determining error position, 2) Generating error types on error marked words, 3) Generating ASR errors such as substitution, deletion and insertion errors, and 4) Rescoring and selecting simulated erroneous utterances.

4.4 User Simulation Criteria

Since simulated users are developed to replace a real user, the fundamental criterion of a user simulation is how the simulated patterns are as *natural* as a real user's behavior. In user intention simulation, the simulated intention should be natural to the corresponding discourse context. In the user surface simulation, the generated user utterance should be as natural as a real user's utterance. In an ASR channel simulation, the noise-added sentences are similar to the recognized sentences by a real ASR.

Unlike the other natural language problems such as text classification and spoken language understanding where naturalness and accuracy are important, *variety* is another important criterion in user simulation. Since one of the main goals of user simulation is evaluating spoken dialog systems with various environments, generating diverse intention, surface and error patterns are desirable.

Another important criterion of user simulation is *controllability*. Developers want to test their dialog systems in a specific environment they want. Therefore successful user simulation should allow controllability by developers to manipulate the characteristics of simulated users. It might be a user type control in intention level. Grammar level or fluency level control might be necessary in surface simulation. Word error rate control is a basic controllability for ASR channel simulation.

5. SUMMARY

This paper has provided a review for recent approaches to dialog management in spoken dialog systems. The dialog management techniques are very important to control the interaction with the user and to communicate with external knowledge sources. Error handling is also closely related to the dialog management problem to improve the robustness of the spoken dialog systems in a noisy environment. Recently, a number of approaches to dialog management have been developed to deploy a practical spoken dialog system in the real world. Data-driven approaches can be used to easily build the dialog management strategies and to overcome the problems arising from traditional knowledge-based approaches. In addition, hybrid approaches can solve the complexity problem of RL-based dialog systems and improve the usability of spoken dialog systems. The user simulation techniques have also been focused on automatically evaluating the spoken dialog systems.

Looking to the future, it can be expected that spoken dialog systems will become more widely used and accepted in the real world. However, there are still some challenges that need to be overcome. First, it is important to develop methods that compensate for poor speech recognition rate, such as the error handling and the robust dialog modeling techniques. In addition, the spoken dialog systems should be developed to new types of applications and new areas of deployment that go beyond the telephone-based systems for simple information-access dialogs. For example, a broader acceptance of language learning systems using spoken dialog technology could reduce education expenses for learning foreign languages. Finally, enabling spoken dialog systems to automatically learn from experiences (e.g., dialog logs) is an important new research topic.

ACKNOWLEDGMENT

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-C1090-0902-0045).

REFERENCES

AI, H. AND LITMAN, D. J. 2008. Assessing Dialog System User Simulation Evaluation Measures Using Human Judges. In *Proc. of the Association for Computational Linguistics*. 622–629.

ALLEN, J., BYRON, D., DZIKOVSKA, M., FERGUSON, G., GALESCU, L., AND STENT, A. 2000. An Architecture for a Generic Dialogue Shell. *Natural Language Engineering 6*, 3, 1–16.

BANGALORE, S., FABBRIZIO, G. D., AND STENT, A. 2006. Learning the structure of task-driven human-human dialogs. In *Proc. of the Association for Computational Linguistics*. 201–208.

BOHUS, D. AND RUDNICKY, A. 2003. RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. In *Proc. of the European Conference on Speech, Communication and Technology*. 597–600.

BOHUS, D. AND RUDNICKY, A. I. 2005. Error Handling in the RavenClaw Dialog Management Framework. In *Proc. of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 225–232.

BOHUS, D. AND RUDNICKY, A. I. 2009. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech and Language 23*, 332–361.

BUI, T., RAJMAN, M., AND MELICHAR, M. 2004. Rapid Dialogue Prototyping Methodology. In *Proc. of the international conference on Text, Speech, and Dialogue.* 579–586.

CHUNG, G. 2004. Developing a flexible spoken dialog system using simulation. In *Proc. of the Association for Computational Linguistics.* 63–70.

CUAYAHUITL, H., RENALS, S., LEMON, O., AND SHIMODAIRA, H. 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding.* 100–105.

DENG, Y., MAHAJAN, M., AND ACERO, A. 2003. Estimating Speech Recognition Error Rate Without Acoustic Test Data. In *Proc. of the European Conference on Speech, Communication and Technology.*

ECKERT, W., LEVIN, E., AND PIERACCINI, R. 1997. User modeling for spoken dialogue system evaluation. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding.* 80–87.

FOSLER-LUSSIER, E., AMDAL, I., AND KUO, H. K. J. 2002. On the Road to Improved Lexical Confusability Metrics. In *Proc. of the ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology.*

FUKUBAYASHI, Y., KOMATANI, K., OGATA, T., AND OKUNO, H. 2006. Dynamic Help Generation by Estimating User°Øs Mental Model in Spoken Dialogue Systems. In P*roc. of the International Conference on Spoken Language Processing.* 1946–1949.

GEORGILA, K., HENDERSON, J., AND LEMON, O. 2005. Learning User Simulations for Information State Update Dialogue Systems. In *Proc. of the European Conference on Speech, Communication and Technology.* 893–896.

GORIN, A., RICCARDI, G., AND WRIGHT, J. 1997. How May I Help You? *Speech Communication 23*, 1-2, 113–127.

GRIOL, D., RICCARDI, G., AND SANCHIS, E. 2009. Learning the Structure of Human-Computer and Human-Human Dialogs. In *Proc. of the Annual Conference of the International Speech Communication Association*. 2775–2778.

HAZEN, T. J., SENEFF, S., AND POLIFRONI, J. 2002. Recognition confidence scoring and its use in speech understanding systems. *Computer Speech and Language 16*, 1, 49–67.

HENDERSON, J., LEMON, O., AND GEORGILA, K. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics 34*, 4, 487–511.

HOCKEY, B., LEMON, O., CAMPANA, E., HIATT, L., HIERONYMUS, J., GRUENSTEIN, A., AND DOWDING, J. 2003. Targeted help for spoken dialogue systems: intelligent feedback improves naive users' performance. In *Proc. of the European Chapter of Association of Computational Linguistics.* 147–154.

HURTADO, L. F., GRIOL, D., SANCHIS, E., AND SEGARRA, E. 2005. A Stochastic Approach to Dialog Management. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding.* 226–231.

INUI, M., EBE, T., INDURKHYA, B., AND KOTANI, Y. 2001. A Case-Based Natural Language Dialogue System using Dialogue Act. In *Proc. of the IEEE International Conference on Systems, Man,* *and Cybernetics.* 193–198.

JUNG, S., LEE, C., KIM, K., JEONG, M., AND LEE, G. G. 2009. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer Speech and Language 23*, 4, 479–509.

JUNG, S., LEE, C., KIM, K., AND LEE, G. G. 2009. Hybrid Approach to User Intention Modeling for Dialog Simulation. In *Proc. of the Association for Computational Linguistics.* 17–20.

KOMATANI, K., KANDA, N., NAKANO, M., NAKADAI, K., TSUJINO, H., OGATA, T., AND OKUNO, H. G. 2006. Multi-Domain Spoken Dialogue System with Extensibility and Robustness against Speech Recognition Errors. In *Proc. of the SIGDIAL Workshop on Discourse and Dialogue.* 9–17.

KOO, M. W., LEE, C. H., AND JUANG, B. H. 2001. Speech Recognition and Utterance Verification Based on a Generalized Confidence Score. *IEEE Trans. on Speech and Audio Processing 9*, 8, 821–832.

LAFFERTY, J. D., MCCALLUM, A., AND PEREIRA, F. C. N. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proc. of the International Conference on Machine Learning*, 282–289.

LAMEL, L., ROSSET, S., GAUVAIN, J., AND NENNACEF, S. 1999. The LIMSI ARISE system for train travel information. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing.* 501–504.

LARSSON, S. AND ERICSSON, S. 2002. GoDiS - issue-based dialogue management in a multidomain, multi-language dialogue system. In *Demo abstract in the Association of Computational Linguistics.* 104–105.

LARSSON, S. AND TRAUM, D. R. 2006. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering 6*, 323–340.

LEE, C., JUNG, S., KIM, K., AND LEE, G. G. 2009a. Automatic Agenda Graph Construction from Human-Human Dialogs using Clustering Method. In *Proc. of the North American Chapter of the Association for Computational Linguistics/Human Language Technology.* 17–20.

LEE, C., JUNG, S., KIM, K., AND LEE, G. G. 2009b. Hybrid Approach to Robust Dialog Management using Agenda and Dialog Examples. *Computer Speech and Language.*

LEE, C., JUNG, S., KIM, S., AND LEE, G. G. 2009c. Example-based Dialog Modeling for Practical Multi-domain Dialog System. *Speech Communication 51*, 5, 466–484.

LEE, C., JUNG, S., LEE, D., AND LEE, G. G. 2007. Example-based Error Recovery Strategy for Spoken Dialog System. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding.* 538–543.

LEMON, O., GEORGILA, K., AND HENDERSON, J. 2006a. Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with Real Users: The TALK TownInfo Evaluation. In *Proc. of the IEEE Spoken Language Technology Workshop.*

LEMON, O., GEORGILA, K., AND HENDERSON, J. 2006b. Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: the talk towninfo evaluation. In *Proc. of the European Conference on Speech, Communication and Technology.* 178–181.

LEMON, O., GEORGILA, K., HENDERSON, J., AND STUTTLE, M. 2006. An ISU Dialogue System Exhibiting Reinforcement Learning of Dialogue Policies: Generic Slot-Filling in the TALK Incar System. In *Proc. of the European Chapter of Association of Computational Linguistics.* 119–122.

LEMON, O., GRUENSTEIN, A., AND PETERS, S. 2002. Multi-tasking and Collaborative Activities in Dialogue Systems. In *Proc. of the SIGDIAL Workshop on Discourse and Dialogue.* 113–124.

LEMON, O., LIU, X., SHAPIRO, D., AND TOLLANDER, C. 2006. Hierarchical Reinforcement Learning of Dialogue Policies in a development environment for dialogue systems: REALLDUDE. In *Proc. of Brandial, the 10th SemDial Workshop on the Semantics and Pragmatics of Dialogue (demonstration systems).*

LEVIN, E., PIERACCININ, R., AND ECKERT, W. 2000. A stochastic model of computer-human interaction for learning dialog strategies. *IEEE Trans. on Speech and Audio Processing 8*, 1, 11–23.

LO, W. AND SOONG, F. 2005. Generalized Posterior Probability for Minimum Error Verification of Recognized Sentences. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 85–88.

LÓPEZ-CÓZAR, R., CALLEJAS, Z., AND MCTEAR, M. 2006. Testing the performance of spoken dialogue systems by means of an artificially simulated user. *Artificial Intelligence Review 26*, 4, 291–323.

LÓPEZ-CÓZAR, R., LA TORRE, A. D., SEGURA, J. C., AND RUBIO, A. J. 2003. Assessment of dialogue systems by means of a new simulation technique. *Speech Communication 40*, 3, 387–407.

MCTEAR, M. 1998. Modelling spoken dialogues with state transition diagrams: Experience of the CSLU toolkit. In *Proc. of the International Conference on Spoken Language Processing*. 1223–1226.

MCTEAR, M., O'NEILL, I., HANNA, P., AND LIU, X. 2005. Handling errors and determining confirmation strategies-An object-based approach. *Speech Communication 45*, 3, 249–269.

MINKER, W., HAIBER, U., HEISTERKAML, P., AND SCHEIBLE, S. 2004. SENECA spoken language dialogue system. *Speech Communication 43*, 89–102.

MURAO, H., KAWAGUCHI, N., MATSUBARA, S., YMAGUCHI, Y., AND INAGAKI, Y. 2003. Examplebased Spoken Dialogue System using WOZ System Log. In *Proc. of the SIGDIAL Workshop on Discourse and Dialogue*. 140–148.

NEEDLEMAN, S. AND WUNSCH, C. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology 48*, 3, 443–453.

PAEK, T. 2006. Reinforcement learning for spoken dialogue systems: comparing strengths and weaknesses for practical deployment. In *Proc. of Workshop on Dialogue on Dialogues, International Conference of Spoken Language Processing.*

PAKUCS, B. 2003. Towards dynamic multi-domain dialogue processing. In *Proc. of the European Conference on Speech, Communication and Technology*. 741–744.

PAPINENI, K., ROUKOS, S., WARD, T., AND ZHU, W.-J. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of the Association for Computational Linguistics*. 311–318.

PECKHAM, J. 1993. A new generation of spoken dialog systems: results and lessons from the SUNDIAL project. In *Proc. of the European Conference on Speech, Communication and Technology*. 33–40.

PIETQUIN, O. 2004. A Framework for Unsupervised Learning of Dialogue Strategies, Ph.D. Thesis, Faculty of Engineering, Mons.

PIETQUIN, O. AND DUTOIT, T. 2006. A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning. *IEEE Trans. on Audio, Speech and Language Processing 14*, 2, 589–599.

PIETQUIN, O. AND RENALS, S. 2002. ASR System Modeling for Automatic Evaluation and Optimization of Dialogue Systems. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing.*

PROMMER, T., HOLZAPFEL, H., AND WAIBEL, A. 2006. Rapid Simulation-Driven Reinforcement Learning of Multimodal Dialog Strategies in Human-Robot Interaction. In *Proc. of the International Conference on Spoken Language Processing.*

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.

RICHARDSON, M. AND DOMINGOS, P. 2006. Markov logic networks. *Machine Learning 62*, 1, 107–136.

ROY, S. AND SUBRAMANIAM, L. V. 2006. Automatic generation of domain models for call centers from noisy transcriptions. In *Proc. of the Association for Computational Linguistics*. 737–744.

SCHATZMANN, J., GEORGILA, K., AND YOUNG, S. 2005. Quantitative Evaluation of User Simulation Technique for Spoken Dialogue Systems. In *Proc. of the SIGDIAL Workshop on Discourse and Dialogue*. 45–54.

SCHATZMANN, J., THOMSON, B., WEILHAMMER, K., YE, H., AND YOUNG, S. 2007. Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proc. of the Human*

Language Technology/North American Chapter of the Association for Computational Linguistics. 149–152.

SCHATZMANN, J., THOMSON, B., AND YOUNG, S. 2007. Error simulation for training statistical dialogue systems. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding*. 526–531.

SENEFF, S. 2002. Response planning and generation in the Mercury flight reservation system. *Computer Speech and Language 16*, 3, 283–312.

SHIN, J., NARAYANAN, S., GERBER, L., KAZEMZADEH, A., AND BYRD, D. 2002. Analysis of User Behavior under Error Conditions in Spoken Dialogs. In *Proc. of the International Conference on Spoken Language Processing*. 2069–2072.

SKANTZE, G. 2005. Exploring human error recovery strategies: Implications for spoken dialogue systems. *Speech Communication 45*, 3, 325–341.

STUTTLE, M., WILLIAMS, J., AND YOUNG, S. 2004. A framework for dialog data collection using a simulated ASR channel. In *Proc. of the International Conference on Spoken Language Processing*.

THOMSON, B., SCHATZMANN, J., AND YOUNG, S. 2008. Bayesian Update of Dialogue State For Robust Dialogue Systems. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 4937–4940.

TORRES, F., HURTADO, L., GARCIA, F., SANCHIS, E., AND SEGARRA, E. 2005. Error handling in a stochastic dialog system through confidence measures. *Speech Communication 45*, 3, 211–229.

WALKER, M., ABERDEEN, J., BOLAND, J., BRATT, E., GAROFOLO, J., HIRSCHMAN, L., LE, A., LEE, S., NARAYANAN, S., PAPINENI, K., PELLOM, B., POLIFRONI, J., POTAMIANOS, A., PRABHU, P., RUDNICKY, A., SANDERS, G., SENEFF, S., STALLARD, D., AND WHITTAKER, S. 2001. DARPA communicator dialog travel planning systems: the June 2000 data collection. In *Proc. of the European Conference on Speech, Communication and Technology*. 1371–1374.

WALKER, M., LANGKILDE, I., WRIGHT, J., GORIN, A., AND LITMAN, D. 2000. Learning to predict problematic situations in a spoken dialogue system: experiments with How May I Help You? In *Proc. of the North American Chapter of the Association for Computational Linguistics*. 210–217.

WALKER, M. A., LITMAN, D. J., KAMM, C. A., AND ABELLA, A. 1997. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In *Proc. of the European Chapter of Association of Computational Linguistics*. 271–280.

WENG, F., VARGES, S., RAGHUNATHAN, B., RATIU, F., PON-BARRY, H., LATHROP, B., ZHANG, Q., BRATT, H., SCHEIDECK, T., XU, K., PURVER, M., MISHRA, R., LIEN, A., RAYA, M., PETERS, S., MENG, Y., RUSSELL, J., CAVEDON, L., SHRIBERG, E., SCHMIDT, H., AND PRIETO, R. 2006. CHAT: A Conversational Helper for Automotive Tasks. In *Proc. of the International Conference on Spoken Language Processing*. 1061–1064.

WILLIAMS, J. D. 2008a. Evaluating User Simulations with the Cramer-von Mises Divergence. *Speech Communication 50*, 829–846.

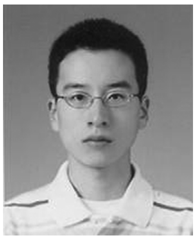
WILLIAMS, J. D. 2008b. The best of both worlds: Unifying conventional dialog systems and POMDPs. In *Proc. of the International Conference on Spoken Language Processing*.

WILLIAMS, J. D. AND YOUNG, S. 2005. Scaling Up POMDPs for Dialog Management: The "Summary POMDP" Method. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding*. 250–255.

WILLIAMS, J. D. AND YOUNG, S. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language 21*, 393–422.

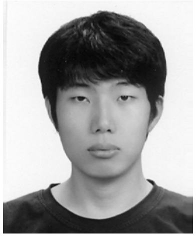
YOUNG, S., SCHATZMANN, J., WEILHAMMER, K., AND YE, H. 2007. The Hidden Information State Approach to Dialog Management. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 149–152.

ZUE, V., SENEFF, S., GLASS, J., POLIFRONI, J., PAO, C., HAZEN, T., AND HETHERINGTON, L. 2000. JUPITER: a telephone-based conversatioinal interface for weather information. *IEEE Trans. on Speech and Audio Processing 8*, 85–96.



Cheongjae Lee is a Ph.D. student at the Department of Computer Science and Engineering at POSTECH, Pohang, South Korea. He holds B.S. degrees of both Computer Science and Engineering, and Life Science from POSTECH. His research interests include spoken dialog system, robust dialog management, and dialog-based computer assisted language learning.

Sangkeun Jung received his B.S./M.S./Ph.D. degrees at the Department of Computer Science and Engineering from POSTECH. His research interests are spoken dialog systems and user simulation.



Kyungduk Kim is a Ph.D. student at the Department of Computer Science and Engineering at POSTECH, Pohang, South Korea. He received his B.S./M.S. degrees at the Department of Computer Science and Engineering from POSTECH. His research interests include multi-modal dialog system and robust dialog management.

Donghyeon Lee is a Ph.D. student at the Department of Computer Science and Engineering at POSTECH, Pohang, South Korea. He received his B.S. degree of the Department of Computer Science and Engineering at Sung Kyun Kwan University, Suwon, South Korea. His research interests are spoken dialog systems, automatic speech recognition, and bootstrapping dialog systems.

Gary Geunbae Lee received his B.S. and M.S. degrees in Computer Engineering from Seoul National University in 1984 and 1986 respectively. He received Ph.D. degree in Computer Science from UCLA in 1991 and was a research scientist in UCLA from 1991.3 to 1991.9. He has been a professor at CSE department, POSTECH in Korea since 1991. He is a director of Intelligent Software laboratory which focuses on human language technology researches including natural language processing, speech recognition/synthesis, speech translation. He authored more than 100 papers in international journals and conferences, and has served as a technical committee member and reviewer for several international conferences such as ACL, COLING, IJCAI, ACM SIGIR, AIRS, ACM IUI, Interspeech-ICSLP/EUROSPEECH, EMNLP and IJCNLP. He is currently leading several national and industry projects for robust spoken dialog systems, computer assisted language learning, and expressive TTS.