An Integrated Dialog Simulation Technique for Evaluating Spoken Dialog Systems

Sangkeun Jung, Cheongjae Lee, Kyungduk Kim, Gary Geunbae Lee
Department of Computer Science and Engineering
Pohang University of Computer Science and Technology(POSTECH)
San 31, Hyoja-Dong, Pohang, 790-784, Korea
{hugman, lcj80, getta, gblee}@postech.ac.kr

Abstract

This paper proposes a novel integrated dialog simulation technique for evaluating spoken dialog systems. Many techniques for simulating users and errors have been proposed for use in improving and evaluating spoken dialog systems, but most of them are not easily applied to various dialog systems or domains because some are limited to specific domains or others require heuristic rules. In this paper, we propose a highly-portable technique for simulating user intention, utterance and Automatic Speech Recognition (ASR) channels. This technique can be used to rapidly build a dialog simulation system for evaluating spoken dialog systems. We propose a novel user intention modeling and generating method that uses a linear-chain conditional random field, a data-driven domain specific user utterance simulation method, and a novel ASR channel simulation method with adjustable error recognition rates. Experiments using these techniques were carried out to evaluate the performance and behavior of previously developed dialog systems designed for navigation dialogs, and it turned out that our approach is easy to set up and shows the similar tendencies of real users.

1 Introduction

Evaluation of spoken dialog systems is essential for developing and improving the systems and for assessing their performance. Normally, humans are used to evaluate the systems, but training and employing human evaluators is expensive. Furthermore, qualified human users are not always immediately available. These inevitable difficulties of working with human users can cause huge delay in development and assessment of spoken dialog systems. To avoid the problems that result from using humans to evaluate spoken dialog systems, developers have widely used dialog simulation, in which a simulated user interacts with a spoken dialog system.

Many techniques for user intention, utterance and error simulation have been proposed. However, previously proposed simulation techniques cannot be easily applied to evaluate various dialog systems, because some of these techniques are specially designed to work with their own dialog systems, some require heuristic rules or flowcharts, and others try to build user side dialog management systems using specialized dialog managing methods. These problems motivated us to develop dialog simulation techniques which allow developers to build dialog simulation systems rapidly for use in evaluating various dialog systems.

To be successful, a simulation approach should not depend on specific domains or rules. Also it should not be coupled to a specific dialog management method. Furthermore, successful dialog simulation should fully support both user simulation and environment simulation. In user simulation, it must be capable of simulating both user intentions and user utterances, because user utterances are essential for testing the language understanding component of the dialog system. In addition to user simulation, environment simulation such as ASR channel simulation is desirable because it allows developers to test the dialog system in various acoustic environments.

In this paper, we propose novel dialog simulation techniques which satisfy these requirements. We introduce a new user intention simulation method based on the sequential graphical model, and a user utterance simulator which can generate diverse natural user utterances. The user intention and utterance simulators are both fully data-driven approaches; therefore they have high domain- and language portability. We also propose a novel Automatic Speech Recognizer (ASR) channel simulator which allows the developers to set the desired speech recognition performance level. Through a case study, we showed that our approach is feasible in successful dialog simulation to evaluate spoken dialog

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 9–16
Manchester, August 2008

systems.

This paper is structured as follows. We first provide a brief introduction of other dialog simulation techniques and their differences from our approach in Section 2. We then introduce the overall architecture and the detailed methods of intention, utterance and ASR channel simulation in Section 3. Experiments to test the simulation techniques, and a case study are described in Section 4. We conclude with a brief summary and suggest directions for future work in Section 5.

2 Related Works

Dialog simulation techniques can be classified according to the purpose of the simulation. One of the purposes is to support the refinement of dialog strategies. Some techniques use large amounts of simulated data for a systematic exploration of the dialog state space in the framework of reinforcement learning (Schatzmann et al., 2005; Schatzmann et al., 2007a). Other techniques use simulation techniques to investigate and improve the target dialog strategies by examining the results heuristically or automatically (Chung, 2004; Rieser and Lemon, 2006; Torres et al., 2008). A second purpose of dialog simulation techniques is to evaluate the dialog system itself qualitatively. Eckert et al., (1997) and López-Cózar et., (2003; 2006) used a dialog simulation to evaluate whole dialog systems.

Dialog simulation techniques can also be classified according to the layers of the simulation. Typically, dialog simulation can be divided into three layers: user intention, user surface (utterance) and error simulation.

Some studies have focused only on the intention level simulation (Rieser and Lemon, 2006; Schatzmann et al., 2007b; Cuayahuitl et al., 2005). The main purpose of those approaches was to collect and examine intention level dialog behavior for automatically learning dialog strategies. In this case, surface and error simulations were neglected or simply accessed normally.

Another approach is to simulate both user intention and surface. In this approach, user utterance generation is designed to express a given intention. Chung (2004) tried to use the natural language generation module of (Seneff, 2002) to generate this surface. He used a speech synthesizer to generate user utterances. López-Cózar et., (2003; 2006) collected real human utterances, and selected and played the voice to provide input for the spoken dialog system. Both Chung (2004) and López-Cózar et., (2003; 2006) used rule based intention simulation. They used real ASR to recognize the synthesized or played voice; hence, ASR channel simulation is not needed in their techniques. Scheffler and Young (2000; 2001) used the lattices which are derived from the grammars used by the recognition engine, but generated user utterances by associating the lattice edges with intentions. During utterance generation, they simulated errors in recognition and understanding by probabilistic substitution on the selection of the edge. Schatzmann et al., (2007a; 2007b) proposed a statistical model for user utterance generation and error simulation using agenda based intention simulation.

The existing rule-based techniques for simulating intentions or surfaces are not appropriate in the sense of portability criteria. In addition, specific dialog managing techniques based user simulators (e.g., (Torres et al., 2008)) are not desirable because it is not easy to implement these techniques for other developers. Another important criterion for evaluating dialog simulation techniques for use in evaluating spoken dialog systems is the range of simulation layers. Simulations that are restricted to only the intention level are not sufficient to evaluate the whole dialog system. Domain and language independent techniques for simulating both intentions and utterances are needed, and ASR channel simulation is desirable for evaluating the spoken dialog systems accurately because human-machine dialog is heavily influenced by speech recognition errors.

3 Dialog Simulation Architecture for Dialog System Evaluation

3.1 Overall Architecture

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, Spoken Language Understanding (SLU) and Dialog Manager (DM). Conventionally, ASR has been considered to be a component of dialog systems. However, in this research, we do not include a real ASR module in the dialog system component because a real ASR takes only fixed level of speech as an input. To use real voices, we must either collect real human speech or generate voices using a speech synthesizer. However, both approaches have limitations. When recording and playing real human voices, the cost of data collection is high and the simulator can simulate only the behavior of the humans who were recorded. When using a speech synthesizer, the synthesizer can usually generate the speech of one person, on a limited variety of speech behaviors; this means that the dialog system cannot be evaluated under various conditions. Also, in both approaches, freely adjusting the speech recognition performance level is difficult. In this research, instead of using real speech we simulate the ASR channel and add noises to a clean utterance from the user simulator to mimic the speech recognition result.

The overall architecture of our dialog simulation separates the user simulator into two levels: user intention simulator and utterance simulator (Fig. 1). The user intention simulator accepts the discourse circumstances 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 noises to the utterance. This noisy utterance is passed to a dialog sys-

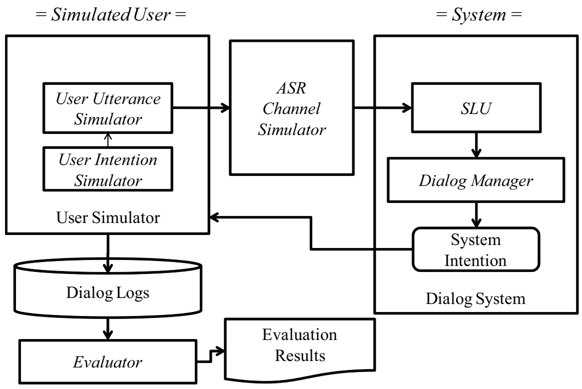


Figure 1: Overall architecture of dialog simulation

tem which consists of a SLU and a DM. The dialog system understands the user utterance, manages dialog and passes the system intention to the user simulator. User simulator, ASR channel simulator and dialog system repeat the conversation until the user simulator generates an end to the dialog.

After finishing simulating one dialog successfully, this dialog is stored in Dialog Logs. If the dialog logs contain enough dialogs, the evaluator uses the logs to evaluate the performance of the dialog system.

3.2 User Intention Simulation

The task of user intention simulation is to generate subsequent user intentions given current discourse circumstances. The intention is usually represented as abstracted user's goals and information on user's utterance (surface). In other words, generating the user's next semantic frame from the current discourse status constitutes the user intention simulation.

Dialog is basically sequential behavior in which participants use language to interact with each other. This means that intentions of the user or the system are naturally embedded in a sequential structure. Therefore, in intention modeling we must consider how to model this sequential property. Also, we must understand that the user's intention depends not only on previous n-gram user and system intentions, but also on diverse discourse circumstances, including dialog goal, the number of items, and the number of filled component slots. Sophisticated user intention modeling should be able to reflect the discourse information.

To satisfy the sequential property and use rich information for user intention modeling, we used linear-chain Conditional Random Field (CRF) model (Lafferty et al., 2001) for user intention modeling. Let Y, X be random vectors, $\Lambda = \{\lambda_k\} \in \mathrm{R}^K$ be a parameter vector, and $\{f_k(y, y', \mathbf{x_t})\}_{k=1}^{K}$ be a set of real-valued feature functions. Then a linear-chain CRF is a distribution of $p(\mathbf{y}|\mathbf{x})$ that takes the form

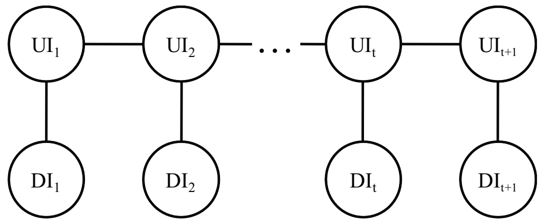


Figure 2: Conditional Random Fields for user intention modeling. UI_t: User Intention ; DI_t: Discourse Information for the tth user turn

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x_t})\right\} \quad (1)$$

where $Z(\mathbf{x})$ is an instance-specific normalization function.

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x_t})\right\}$$

CRF is an undirected graphical model that defines a single log-linear distribution over the joint probability of an entire label sequence given a particular observation sequence. This single distribution removes the per-state normalization requirement and allows entire state sequences to be accounted for at once. This property is well suited to model the entire sequence of intentions in a dialog. Also, CRF is a conditional model, and not a joint model (such as the Hidden Markov Model). Arbitrary facts can be captured to describe the observation in the form of indicator functions. This means that CRF allows us to use rich discourse information to model intentions.

CRF has states and observations in each time line. We represent the user intention as state and discourse information as observations in CRF (Fig. 2). We represent the state as a semantic frame. For example in the semantic frame representing the user intention for the utterance 'I want to go to city hall' (Fig. 3), 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 named entities in the utterance. We use the cartesian product of each slot of semantic frame to represent the state of the utterance in our CRF model. In this example, the state symbol is 'request×search_loc×loc_name'.

For the observation, we can use various discourse events because CRF allows using rich information by interpreting each event as an indicator function. Because we pursue the portable dialog simulation technique, we separated the features of the discourse information into those that are domain independent and those that are domain dependent. Domain independent

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*
Preprocessing Information for User Utterance Simulation	
processed utterance	*I/PRP want/VB to/TO go/VB to/TO [loc_name] /[loc_name]*
Structure Tags	*PRP, VB, TO, VB, TO, [loc_name]*
Word Vocabulary	*I, want, to, go, to, [loc_name]*
Generation Target for User Utterance Simulation	
Structure	*PRP → VB → TO → VB → TO → [loc_name]*
Word Sequence	*I → want → to → go → to → [loc_name]*

Figure 3: Example of semantic frame for user intention, and preprocessing and generation target for user utterance simulation.

Domain Independent Features	
PREV_1_SYS_ACT	previous system action. Ex) PREV_1_SYS_ACT=confirm
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_fav_rest)=gajokjung

Figure 4: Example feature design for navigation domain

UI_t ← user intention at t turn
S ← user intentions set ($UI_t \in S$)
$UI_1, UI_2, \ldots, UI_{t-1}$ ← already simulated user intention sequence
$DI_1, DI_2, \ldots, DI_t$ ← discourse information from 1 to t turn

For each UI_t in S
    Calculate P($UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t$)
UI_t ← random user intention from P($UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t$)

Figure 5: User intention generation algorithm

features include discourse information which is not relevant to the specific dialog domain and system. For example, previous system acts in Fig. 4 are not dependent on specific dialog domain. The actual values of previous system acts could be dependent on each dialog domain and system, but the label itself is independent because every dialog system has system parts and corresponding system acts. In contrast, domain specific discourse information exists for each dialog system. For example, in the navigation domain (Fig. 4), the current position of the user or the user's favorite restaurant could be very important for generating the user's intention. This information is dependent on the specific domain and system. We handle these features as 'OTHER_INFO'.

We trained the user intention model using dialog examples of human-machine. One training example consists of a sequence of user intentions and discourse information features in a given dialog. We collected training examples and trained the intention model using a typical CRF training method, a limited-memory quasi-Newton code for unconstrained optimization (L-BFGS) of (Liu and Nocedal, 1989).

To generate user intentions given specific discourse circumstances, we calculate the probability of a sequence of user intentions from the beginning of the dialog to the corresponding turn. For example, suppose that we need to generate user intention at the third turn (UI_3) (Fig. 2). We have previously simulated user intentions UI_1 and UI_2 using DI_1 and DI_2. In this case, we calculate the probability of $UI_1 \rightarrow UI_2 \rightarrow UI_3$ given DI_1, DI_2 and DI_3. Notice that DI_3 contains discourse information at the third turn: it includes previous system intention, attributes and other useful information. Using the algorithm (Fig. 5) we generate the user intention at turn t. The probability of $P(UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t)$ is calculated using the equation (1). In the generation of user intention at t turn, we do not select the UI_t which has higher probability. Instead, we select UI_t randomly based on the probability distribution $P(UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t)$ because we want to generate diverse user intention sequence given the same discourse context. If we select UI_t which has highest probability, user intention simulator always returns the same user intention sequence.

3.3 User Utterance Simulation

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]'). We approach the task of user utterance simulation by assuming that the types of structures and the vocabulary are limited when we make utterances to express certain context and intention in a specific domain, and that humans express their intentions by recombining and re-aligning these structures and vocabularies.

To model this process, we need to collect the types of structures and vocabularies. For this, we need to define the context space. We define the structure and vocabulary space as a production of dialog act and main goal. In an example of semantic frame for the utterance "I want to go to city hall" (Fig. 3), the structure and vocabulary (SV) space ID is 'request_#_search_loc', which is produced by the dialog act and the main goal. We collect structure tags, which consist of a part of speech tag, a component slot tag, and a vocabulary that corresponds to SV space. For example (Fig. 3), structure

First Phase – Generating Structures and Words given SV space

1. Repeat generate S_t based on $P_{SV}(S_{t+1}|S_t)$, until S_T = <*setence_end*> , where $S_t \in \boldsymbol{S}$, $t=1,2,3,....T$.
2. Generate W_t based on $P_{SV}(W_t|S_t)$, where $t=1,2,3,..,T$, $W_t \in \boldsymbol{V}$
3. The generation word sequence $\boldsymbol{W}=\{W_1,W_2,..,W_T\}$ is inserted into the set of generated utterance $\boldsymbol{U}$
4. Repeat *1* to *3* for *Max_Generation_Number* times, *Max_Generation_Number* is given by developers

Second Phase – Selection by measure

1.Rescore the utterance U_k in the set of $\boldsymbol{U}$ by the measure
2.Select top n-best

Figure 6: Algorithm of user utterance simulation

tags include PRP, VB, TO, VB as a part of speech tag and [loc_name] as a component slot tag. The vocabulary includes I, want, to, go, and [loc_name]. In the vocabulary, every named-entity word is replaced with its category name.

In this way, we can collect the structure tags and vocabulary for each SV space from the dialog logs. For the given SV space, we estimate probability distributions for statistical user utterance simulation using a training process. For each space, we estimate tag transition probability $P_{SV}(S_{t+1}|S_t)$ and collect structure tags set S_{SV} and vocabularies V_{SV}.

We devised a two-phase user utterance generation algorithm (Fig. 6). Symbols are as follows. The detail explanation of Fig. 6 will be followed in the next subsections.

- $\boldsymbol{S_{SV}}$: structure tag set for given SV
- $\boldsymbol{V_{SV}}$: vocabularies for given SV
- S_i : structure tag, $i = 0, ..., T$, $S_i \in \boldsymbol{S_{SV}}$
- W_i : word, $i = 0, ..., T$, $W_i \in \boldsymbol{V_{SV}}$
- W_{seq} : generated word sequence. $W_{seq} = (W_1, W_2, ..., W_T)$
- U_k : *k*-th sampled utterance, $k = 1, ...,$ *Max_Sampling_Number*, $U_k \in \boldsymbol{U}$

3.3.1 First Phase - Generating Structure and Word Sequence

We generate the structure tag S_1 based on the probability of $P_{SV}(S_1| < sentence\_start >)$ and then S_1 influences the generating of S_2 after $P_{SV}(S_2|S_1)$. In this way, a structure tag chain is generated sequentially based on the structure tag transition probability $P_{SV}(S_{t+1}|S_t)$ until the last generated structure tag S_T is $< sentence\_end >$. We assume that the current structure tag has a first order Markov property, which means that the structure tag is only influenced by the previous structure tag. After the structure tags are generated, the emission probability $P_{SV}(W_t|S_t)(w = 1, \ldots, T)$ is used to generate the word sequence given the tag sequence. We iterate the process of generating structures and word sequences sufficient times to generate many different structure tags and word sequences which may occur in real human expressions. Selecting natural utterances from the generated utterances requires an automatic evaluation metric.

3.3.2 Second Phase - Selection by the BLEU measure

To measure the naturalness of the generated utterances, we use the BLEU (Bilingual Evaluation Understudy) score (Papineni et al., 2001) which is widely used for automatic evaluation in Statistical Machine Translation (SMT). In SMT, translated candidate sentences are evaluated by comparing semantically equivalent reference sentences which have been translated by a human. Evaluation of the user utterance generation shares the same task of evaluation in SMT. We can evaluate the naturalness of generated utterances by comparing semantically equivalent reference utterances collected by humans. Therefore, the BLEU score can be adopted successfully to measure the naturalness of the utterances.

The BLEU score is the geometric mean of the n-gram precisions with a brevity penalty. The original BLEU metric is used to evaluate translated sentences by comparing them to several reference sentences. We modified the BLEU metric to compare one generated utterance with several reference utterances. To rescore the generated utterances, we used the Structure and Word interpolated BLEU score (SWB). After the first phase, we obtain generated utterances which have both structure and word sequence. To measure the naturalness of a generated utterance, we check both structural and lexical naturalness. We calculated Structure_Sequence_BLEU score using the generated structure tags sequences instead of words sequences with the reference structure tag sequences of the SV space in the BLEU calculation process. The Word_Sequence_BLEU is calculated by measuring BLEU score using the generated words sequence with the reference word sequences of the SV space. SWB is calculated as:

$$\begin{aligned} SWB &= \alpha * \text{Structure\_Sequence\_BLEU} \\ &+ (1-\alpha) * \text{Word\_Sequence\_BLEU} \end{aligned}$$

In this study, we set $\alpha = 0.5$. Using SWB, we select the top 20-best generated utterances and return a corresponding generated utterance by selecting one of them randomly.

3.4 ASR channel Simulation

ASR channel simulation generates speech recognition errors which might occur in the real speech recognition process. In this study, we simulate the ASR channel and modify the generated clean utterance to a speech recognized erroneous utterance. Successful ASR channel simulation techniques should have the following properties: the developer should be able to set the simulated word error rate (WER) between 0% ~ 100%; the simulated errors should be generated based on realistic phone-level and word-level confusions; and the technique should be easily adapted to new tasks, at low cost.

Our ASR channel simulation approach is designed to satisfy these properties. The proposed ASR channel simulation method involved 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 (Fig. 7 for Korean language example).

In the first step, we used the WER to determine the positions of erroneous words. For each word, we randomly generate a number between 0 and 1. If this number is between 0 and WER, we mark the word Error Word (1); otherwise we mark the word Clean Word (0). In the second step, we generate ASR error types for the error marked words based on the error type distribution. In the third step, we generate various types of ASR error. In the case of deletion error, we simply delete the error marked word from the utterance. In the case of insertion error, we select one word from the pronunciation dictionary randomly, and insert it before the error marked word. In the case of substitution error, we use a more complex process to select a substitutable word.

To select a substitutable word, we compare the marked error word with the words from pronunciation dictionary which are similar in syllable sequence and phoneme sequence. First, we convert the final word sequence from the user simulator into a phoneme sequence using a Grapheme-to-Phoneme (G2P) module (Lee et al., 2006). Then, we extract a part of the phoneme sequence which is similar to the error marked word from the entire phoneme sequence of the utterance. The reason for extracting a target phoneme sequence corresponding to one word from the entire phoneme sequence is that the G2P results vary between the boundaries of words. Then, we separate the marked word into syllables and compare their syllable-level similarity to other words in the pronunciation dictionary. We calculate a similarity score which interpolates syllable and phoneme level similarity using following equations.

$$\begin{aligned} Similarity &= \beta * \text{Syllable\_Alignment\_Score} \\ &+ (1-\beta) * \text{Phone\_Alignment\_Score} \end{aligned}$$

We used the dynamic global alignment algorithm of (Needleman and Wunsch, 1970) for both syllable and phoneme sequence alignment. This alignment algorithm requires a weight matrix. As a weight matrix, we used a vowel confusion matrix which is based on the manner of articulation. We consider the position (back/front, high/mid/low) of the tongue and the shape (round/flat) of the lips. We select candidate words which have higher similarity than an arbitrary threshold θ and replace the error marked word with a random word from this set. We repeat steps 1 to 3 many times (usually 100) to collect error added utterances.

In the fourth step, we rescore the error added utterances using the language model (LM) score. This LM is trained using a domain corpus which is usually used in ASR. We select top n-best erroneous utterances (we set n=10) and choose one of them randomly. This utterance is the final result of ASR channel simulator, and is fed into the dialog system.

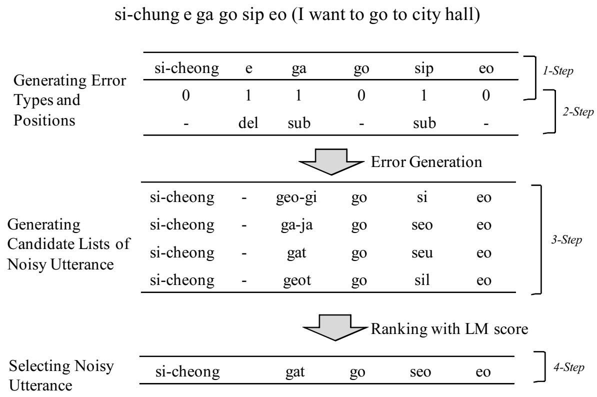


Figure 7: Example of ASR channel simulation

4 Experiments

We proposed a method that user intention, utterance and ASR channel simulation to rapidly assemble a simulation system to evaluate dialog systems. We conducted a case study for the navigation domain Korean spoken dialog system to test our simulation method and examine the dialog behaviors using the simulator. We used 100 dialog examples from real user and dialog system to train user intention and utterance simulator. We used the SLU method of (Jeong and Lee, 2006), and dialog management method of (Kim et al., 2008) to build the dialog system. After trained user simulator, we perform simulation to collect 5000 dialog samples for each WER settings (WER = 0 ~ 40 %).

To verify the user intention and utterance simulation quality, we let two human judges to evaluate 200 randomly chosen dialogs and 1031 utterances from the simulated dialog examples (WER=0%). At first, they evaluate a dialog with three scale (1: Unnatural, 2: Possible, 3: Natural), then evaluate the utterances of a dialog with three scale (1: Unclear, 2: Understandable, 3: Natural).

The inter evaluator agreement (kappa) is 0.45 and 0.58 for dialog and utterance evaluation respectively, which show the moderate agreement (Fig. 8). Both judges show the positive reactions for the quality of user intention and utterance, the simulated dialogs can be possibly occurred, and the quality of utterance is close to natural human utterance.

We also did regression analysis with the results of human evaluation and the SWB score to find out the relationship between SWB and human judgment. Fig. 9 shows the result of polynomial regression (order 3) result. It shows that 'Unclear' utterance might have 0.5

	Human 1	Human 2	Average	Kappa
Dialog	2.38	2.22	2.30	0.45
Utterance	2.74	2.67	2.71	0.58

Figure 8: Human evaluation results on dialog and utterance

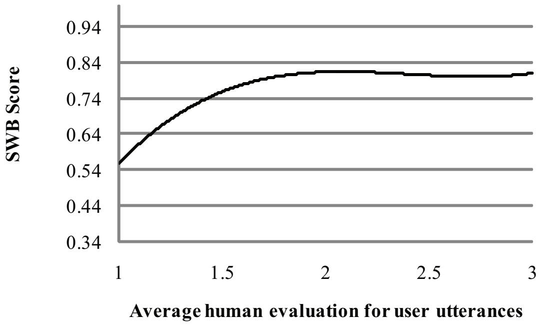
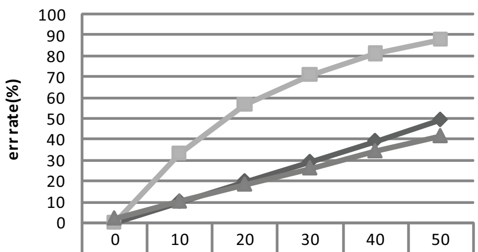


Figure 9: Relationship between SWB score and human judgment

˜ 0.7 SWB score, 'Possible' and 'Natural' simulated utterance might have over 0.75. It means that we can simulate good user utterance if we constrain the user simulator with the threshold around 0.75 SWB score.

To assess the ASR channel simulation quality, we compared how SLU of utterances was affected by WER. SLU was quantified according to sentence error rate (SER) and concept error rate (CER). Compared to WER set by the developer, measured WER was the same, SER increased more rapidly, and CER increased more slowly (Fig. 10). This means that our simulation framework models SLU errors effective as well as speech recognition errors.

Fig. 11 shows the overall dialog system behaviors using the user simulator and ASR channel simulator. As the WER rate increased, dialog system performance decreased and dialog length increased. This result is similar as observed to the dialog behaviors in real human-machine dialog.



	0	10	20	30	40	50
WER measured	0	10	19.71	29	39.06	49.21
SER	0	33.28	56.6	70.91	81.29	88
CER	1.9087	10.1069	18.3183	26.1619	34.4322	41.755

Figure 10: Relationship between given WER and measured other error rates. X-axis = WER fixed by ASR channel(%)

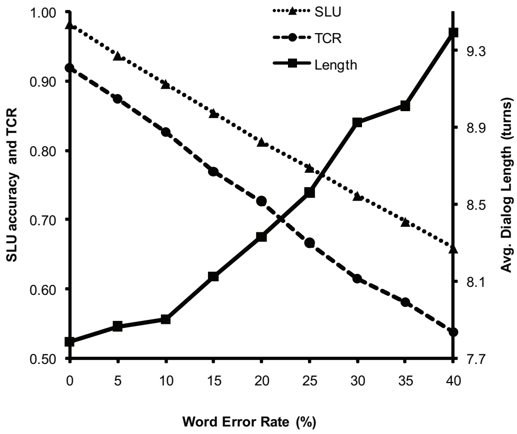


Figure 11: Dialog simulation result on navigation domain

5 Conclusion

This paper presented novel and easy to build dialog simulation methods for use in evaluation of spoken dialog systems. We proposed methods of simulating utterances and user intentions to replace real human users, and introduced an ASR channel simulation method that acts as a real speech recognizer. We introduce a method of simulating user intentions which is based on the CRF sequential graphical model, and an utterance simulator that generates user utterances. Both user intention and utterance simulators use a fully data-driven approach; therefore, they have high domain- and language portability. We also proposed a novel ASR channel simulator which allows the developers to set the speech recognition performance level. We applied our methods to evaluate a navigation domain dialog system; experimental results show that the simulators successfully evaluated the dialog system, and that simulated intention, utterance and errors closely match to those observed in real human-computer dialogs. We will apply our approach to other dialog systems and bootstrap new dialog system strategy for the future works.

6 Acknowledgement

This research was supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy of Korea.

References

Chung, G. 2004. Developing a flexible spoken dialog system using simulation. *Proc. ACL*, pages 63–70.

Cuayahuitl, H., S. Renals, O. Lemon, and H. Shimodaira. 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. *Automatic*

Speech Recognition and Understanding, 2005 IEEE Workshop on, pages 100–105.

Eckert, W., E. Levin, and R. Pieraccini. 1997. User modeling for spoken dialogue system evaluation. *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 80–87.

Jeong, M. and G. Lee. 2006. Jointly Predicting Dialog Act and Named Entity for Statistical Spoken Language Understanding. *Proceedings of the IEEE/ACL 2006 workshop on spoken language technology (SLT)*.

Kim, K., C. Lee, S Jung, and G. Lee. 2008. A frame-based probabilistic framework for spoken dialog management using dialog examples. In *the 9th sigdial workshop on discourse and dialog (sigdial 2008), To appear*.

Lafferty, J.D., A. McCallum, and F.C.N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning table of contents*, pages 282–289.

Lee, J., S. Kim, and G.G. Lee. 2006. Grapheme-to-Phoneme Conversion Using Automatically Extracted Associative Rules for Korean TTS System. In *Ninth International Conference on Spoken Language Processing*. ISCA.

Liu, D.C. and J. Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528.

López-Cózar, R., A. De la Torre, JC Segura, and AJ Rubio. 2003. Assessment of dialogue systems by means of a new simulation technique. *Speech Communication*, 40(3):387–407.

López-Cózar, Ramón, Zoraida Callejas, and Michael Mctear. 2006. Testing the performance of spoken dialogue systems by means of an artificially simulated user. *Artif. Intell. Rev.*, 26(4):291–323.

Needleman, SB and CD Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53.

Papineni, K., S. Roukos, T. Ward, and WJ Zhu. 2001. BLEU: a method for automatic evaluation of MT. *Research Report, Computer Science RC22176 (W0109-022), IBM Research Division, TJ Watson Research Center*, 17.

Rieser, V. and O. Lemon. 2006. Cluster-Based User Simulations for Learning Dialogue Strategies. In *Ninth International Conference on Spoken Language Processing*. ISCA.

Schatzmann, J., K. Georgila, and S. Young. 2005. Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems. In *6th SIGdial Workshop on Discourse and Dialogue*. ISCA.

Schatzmann, J., B. Thomson, and S. Young. 2007a. Error simulation for training statistical dialogue systems. *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 526–531.

Schatzmann, J., B. Thomson, and S. Young. 2007b. Statistical User Simulation with a Hidden Agenda. *Proc. SIGDial, Antwerp, Belgium*.

Scheffler, K. and S. Young. 2000. Probabilistic simulation of human-machine dialogues. *Proc. of ICASSP*, 2:1217–1220.

Scheffler, K. and S. Young. 2001. Corpus-based dialogue simulation for automatic strategy learning and evaluation. *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, pages 64–70.

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

Torres, Francisco, Emilio Sanchis, and Encarna Segarra. 2008. User simulation in a stochastic dialog system. *Comput. Speech Lang.*, 22(3):230–255.