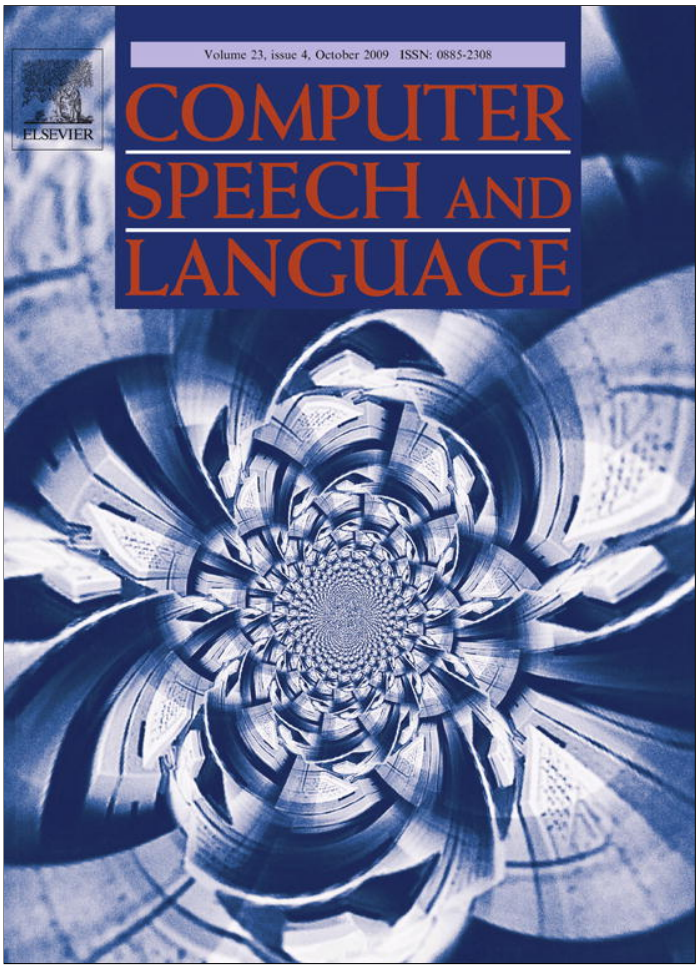

Available online at www.sciencedirect.com
ScienceDirect
Computer Speech and Language 23 (2009) 479–509
COMPUTER SPEECH AND LANGUAGE
www.elsevier.com/locate/csl

Data-driven user simulation for automated evaluation of spoken dialog systems

Sangkeun Jung *, Cheongjae Lee, Kyungduk Kim, Minwoo Jeong, Gary Geunbae Lee

Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), San 31, Hyoja-Dong, Pohang, 790-784, Republic of Korea

Received 15 August 2008; received in revised form 24 February 2009; accepted 6 March 2009
Available online 19 March 2009

Abstract

This paper proposes a novel integrated dialog simulation technique for evaluating spoken dialog systems. A data-driven user simulation technique for simulating user intention and utterance is introduced. A novel user intention modeling and generating method is proposed that uses a linear-chain conditional random field, and a two-phase data-driven domain-specific user utterance simulation method and a linguistic knowledge-based ASR channel simulation method are also presented. Evaluation metrics are introduced to measure the quality of user simulation at intention and utterance. Experiments using these techniques were carried out to evaluate the performance and behavior of dialog systems designed for car navigation dialogs and a building guide robot, and it turned out that our approach was easy to set up and showed similar tendencies to real human users.

Keywords: User simulation; Dialog simulation; Evaluation; Data-driven; Spoken dialog system; Dialog system

1. Introduction

Automated 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 employing and training human evaluators is 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.

Successful dialog simulation should fully support both user simulation and ASR (Automatic Speech Recognition) channel simulation. User simulation must be capable of simulating both user intentions and user utterances because user utterances are essential for testing the language understanding component of the

* Corresponding author. Tel.: +82 54 279 5581; fax: +82 54 279 2299.
E-mail address: hugman@postech.ac.kr (S. Jung).

0885-2308/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2009.03.002

dialog system. ASR channel simulation is also essential because it allows developers to test the dialog system in various acoustic environments.

Many techniques for user intention and utterance simulation have been proposed (Chung, 2004; Cuayahuitl et al., 2005; Eckert et al., 1997; Georgila et al., 2006; López-Cózar et al., 2006; Levin et al., 2000; Pietquin, 2004; Schatzmann et al., 2007c; Scheffler and Young, 2001; Torres et al., 2008). However, previously proposed simulation techniques cannot be easily applied to evaluate multiple 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 management methods. These problems motivated us to develop a new data-driven user simulation technique which allows developers to build user simulation systems rapidly for use in evaluating multiple various dialog systems.

This paper proposes a novel data-driven user simulation technique which supports intention and utterance level simulation as well as simple linguistic knowledge-based ASR channel simulation. We introduce a new user intention simulation method based on a 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. Furthermore novel evaluation metrics are introduced to measure the qualities of user simulation in the level of intention and utterances. A case study showed that our approach is feasible for successful dialog simulation to evaluate spoken dialog systems.

This paper is structured as follows. We first provide a brief introduction of other user simulation techniques and their differences from our approach in Section 2. We then introduce the overall simulation architecture and describe in detail the simulation methods for intention and utterance as well as the simulation method for the ASR channel simulation in Section 3. Novel evaluation metrics for intention and utterance simulation are proposed in Section 4. Experiments to test the simulation techniques and two case studies are described in Section 5. We conclude with a brief summary and suggest directions for future works in Section 6.

2. Related works

User simulation techniques can be classified according to the layers of the simulation. Typically, dialog simulation can be divided into user simulation and error simulation. User simulation can be layered into the user intention level and user surface (utterance) level. Error simulation techniques include automatic speech recognition error simulation and spoken language understanding error simulation.

In recent years, simulation on the intention level has been most popular. This approach was first taken by Eckert et al. (1997) and has been adopted in later work on user simulation by most other research. We can distinguish two main approaches in the user intention simulator implementation: rule-based and data-driven approaches.

In a rule-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., 2006, 2003). Schatzmann et al. proposed an agenda-based user simulation technique for bootstrapping a statistical dialog manager without access to training data (Schatzmann et al., 2007a). It simulates user behavior based on a compact representation of the *user goal* and a stack-like *user agenda*.

Data-driven approach uses statistical methods to generate the user intention given discourse information (history). Earlier studies have involved the use of the 'bigram' model of dialog in which the simulated user input is dependent only on the previous system utterance (Eckert et al., 1997). The advantage of this approach lies in its simplicity and in that it is totally domain- and language independent. Levin et al. modified the bigram model to account for a more realistic degree of conventional structure in dialog (Levin et al., 2000). Scheffler and Young introduced the use of a graph-based model to overcome the lack of goal consistency that the Levin model suffers from while maintaining variability in user behavior (Scheffler and Young, 2000, 2001). It is a goal directed model in which a goal is defined as a specification of the dialog transaction that the user wants to accomplish. The arcs of the network represent actions and the nodes represent choice points. In their work, all of the possible paths that a user may take during a dialog need to be mapped out in advance in the form of a network. Pietquin et al. combines features from Scheffler and Young's work with the Levin model ensuring the actions are consistent with the user's goal without designing the user action networks in advance (Pietquin and Dutoit, 2006; Pietquin, 2004).

Graphical model based user intention simulation techniques have been also proposed. Cuayahuitl et al. combine the goal oriented approach with the bigram model using Hidden Markov Models (HMMs) and Input–Output Hidden Markov Models (IOHMM) to predict not only user intentions but also system intentions (Cuayahuitl et al., 2005). To reflect rich dialog state descriptions and a longer dialog history, Georgila et al. proposed an advanced *n*-gram based user intention simulation model and linear-feature combination model (Georgila et al., 2005). They have extended *n*-gram models with *n* ranging from 2 to 5 in order to cover longer spans of dialog history. The linear-feature combination method maps from a dialog state to a vector of real-valued features. Most of these features are binary, indicating the presence or absence of a piece of information. Supervised learning is used to estimate a set of weights for each indicating function. This technique allows considering richer discourse information to predict next user intention.

Schatzmann et al. extended their work on an agenda-based user model for training statistical dialog managers and presented a method for estimating the model parameters on human–computer dialog data (Schatzmann et al., 2007c). The approach models the observable dialog acts in terms of a sequence of hidden user states and uses an EM-based algorithm to iteratively estimate (locally) optimal parameter values.

A user utterance simulation technique is designed to express a given intention in natural language. Usually, 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 spoken dialog understanding (SLU) as well as the dialog manager (DM).

Chung tried to use the natural language generation module of Seneff (2002) 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., 2007b). A generative maximum-likelihood model for predicting a user utterance for a given user act is built by obtaining the appropriate relative frequency statistics from a transcribed and annotated dialog corpus.

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., 2006). 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). Schatzmann et al. proposed ASR-confusions methods (Schatzmann et al., 2007b). 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). The acoustic distances of their work are based on acoustic-linguistic knowledge.

Natural language understanding error simulation techniques has been explored by using the confidence level of simulated ASR results. Confidences score generation techniques based on distributions of understood user intention matched with original user intention were presented by Pietquin (2004), Williams and Young (2007). In these works, the distributions are handcrafted. In Schatzmann et al. (2007b), the confidence scores for correct and incorrect hypotheses are generated by sampling from the distributions found in the training data. In Pietquin and Dutoit's work, the confidence score (probability) can be calculated in the probabilistic framework of dialog simulation (Pietquin and Dutoit, 2006).

In this research, we developed user intention, surface simulation and ASR channel simulation methods for automated evaluation of spoken dialog systems. Both user intention and utterance simulators are fully data-driven to be domain- and language portable. In the user intention level simulation, a conditional random field based simulation method is proposed. The model captures a longer dialog history and rich discourse descriptions reflecting the information as observation vector. The user surface level simulation was performed by devising a data-driven user utterance synthesis approach rather than by using a corpus-based approach. A two-phase

user utterance generation method is developed. Possible utterance candidates are generated from the structure and words transition distribution and selected using naturalness scores based on BLEU scores. An acoustic and linguistic knowledge-based automatic speech recognition channel simulation technique for Korean is developed for varying the simulation condition. We also introduce novel simulation quality evaluation metrics which measure the naturalness of simulated utterance and intention sequences.

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, SLU and DM modules. The overall architecture of our user simulator is separated into two levels: user intention and utterance simulators (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 noise to the utterance.

Conventionally, ASR has been considered to be a component of dialog systems. However, this research does not include a real ASR module in the dialog system component because a real ASR takes only a fixed level of speech as an input. Using real voices requires either collecting real human speech or generating 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. Instead of using real speech data, an ASR channel simulator is implemented simply based on the linguistic knowledge and connected to the user simulator. The ASR channel simulator adds noise to a clean utterance from the user simulator to mimic the speech recognition result.

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.

After a dialog is simulated successfully, it is stored in the *Dialog Logs*. When the dialog logs contain enough dialogs, the evaluator uses the logs to evaluate the performance of the dialog system. The dialog examples are formatted as XML (eXtensible Markup Language) for communication between simulation components and for storage.

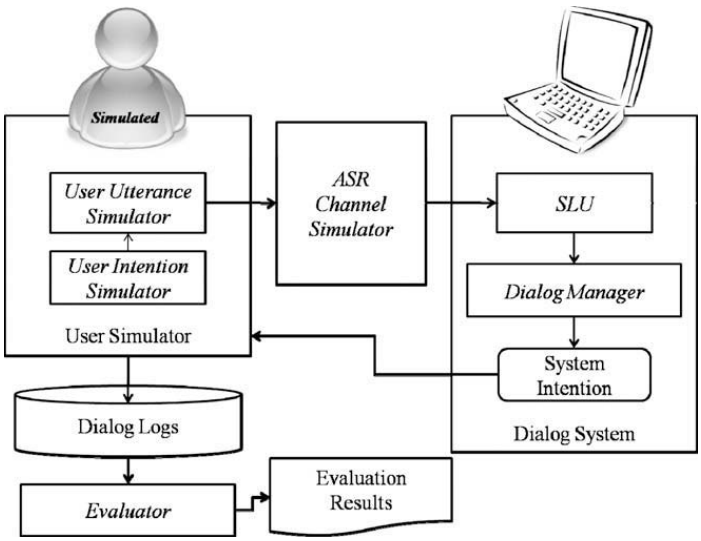

Fig. 1. Overall architecture of dialog simulation.

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 the user's utterance (surface). In other words, generating the user's next semantic frame from the current discourse status constitutes the user intention simulation.

There is 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. Recently, 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. presents a method for intention simulation based on Hidden Markov Models and Input–Output Hidden Markov Models (Cuayahuitl et al., 2005). This paper proposes a novel graphical model based user intention modeling method taking advantage of sequential data modeling with various feature combinations.

A 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, intention modeling must consider this sequential property. Also, the user's intention depends not only on the 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 give rich user information for user intention modeling, we used the linear-chain Conditional Random Fields (CRF) model (Lafferty et al., 2001) for user intention modeling. Let $\mathbf{Y}$, $\mathbf{X}$ 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

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

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

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

A linear-chain 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, a CRF is a conditional model rather than 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 modeling allows us to use rich discourse information to model intentions.

A linear-chain CRF has states and observations in each time line. We represent the user intentions as states, and the discourse information as observations (Fig. 2). We define 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" (Fig. 3), the Cartesian product of each slot of semantic frame represents the state in our CRF model. In this example, the state symbol is 'request + search_loc + [loc_name]'.

The observation can be various discourse events because CRF allows the use of rich information by interpreting each event as an indicator function. The features of the discourse information were separated into those that are domain-independent and domain dependent. Domain-independent features include discourse

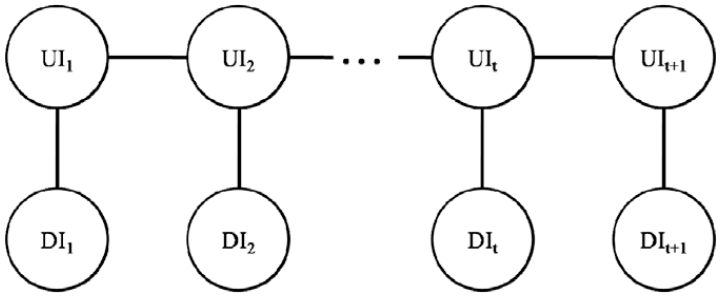

Fig. 2. Conditional Random Fields for user intention modeling. UI_t: User Intention; DI_t: Discourse Information for the tth user turn.

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, [loc_name]*
Word Vocabulary	*I, want, to, go, [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]*

Fig. 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_favorite_restaurant)=gajokjung Ex) OTHER_INFO(user_current_position)=daeidong

Fig. 4. Example feature design for navigation domain.

information which is not relevant to the specific dialog domain and system. For example, previous system acts in Fig. 4 are not dependent on the specific dialog domain. The actual values of previous system acts could depend 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. These features are handled as OTHER_INFO.

function *SimulateNextUserIntention* (Simulated_User_Intentions, Discourse_Information)
{Simulate next user intention U_t given previously simulated user intentions and discourse information}

$U_1, U_2, \ldots, U_{t-1} \leftarrow$ previously simulated user intention from turn 0 to t-1,
$D_1, D_2, \ldots, D_t \leftarrow$ the discourse information from turn 0 to t
$UI'_t \leftarrow$ user intention at t turn
$S \leftarrow$ user intention set ($UI_t \in S$)

{Compute the probability distribution over S}
foreach UI_t in S **do**
  calculate P ($UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t$)

{Randomly select UI_t based on the calculated probability distribution}
$UI'_t \leftarrow$ random user intention from P ($UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t$)

Return UI'_t

Fig. 5. User intention generation algorithm.

The user intention model was trained using human–machine dialog examples. The examples were refined by human annotators, so they consist of good dialog examples. One training example consists of a sequence of user intentions and discourse information features in a given dialog. Training examples were collected and the intention model trained using a typical CRF training method, a limited-memory quasi-Newton code for unconstrained optimization (L-BFGS) (Liu and Nocedal, 1989).

User intentions given specific discourse circumstances were generated by calculating the probability of a sequence of user intentions from the beginning of the dialog to the corresponding turn. For example, user intention at the third turn (UI_3) is generated using previously simulated user intentions UI_1 and UI_2 (Fig. 2). In this case, the probability of $UI_1 \rightarrow UI_2 \rightarrow UI_3$ is calculated 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. The algorithm in Fig. 5 is used to generate the user intention at turn t. The probability of $P(UI_1, UI_2, \ldots, UI_t \mid DI_1, DI_2, \ldots, DI_t)$ is calculated using Eq. (1). User intention at turn t is not generated by selecting the UI_t which has the highest probability. Instead, UI_t is selected randomly based on the probability distribution $P(UI_1, UI_2, \ldots, UI_t \mid DI_1, DI_2, \ldots, DI_t)$ to generate diverse user intention sequences given the same discourse context. If the UI_t with the highest probability is selected, the user intention simulator always returns the same user intention sequence. For example, let the user intention set $S = \{A, B, C\}$ and try to simulate UI_3. Suggest that the probability distribution are $P(UI_1, UI_2, A \mid DI_1, DI_2, DI_3) = 1/2$, $P(UI_1, UI_2, B \mid DI_1, DI_2, DI_3) = 1/4$ and $P(UI_1, UI_2, C \mid DI_1, DI_2, DI_3) = 1/4$, then the user intention simulator randomly selects the next user intention to be A, B and C based on probabilities 1/2, 1/4 and 1/4, respectively.

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]"). The task of user utterance simulation assumes that the types of structures and the vocabulary are limited when we make utterances to express certain contexts and intentions in a specific domain, and that humans express their intentions by re-combining and re-aligning these structures and vocabularies.

To model this process, the types of structures and vocabularies should be collected. Structure tags consist of part-of-speech tags and component slot tags of user utterances, and vocabulary corresponds to the words in the user utterances. The structure tags and vocabulary are grouped by defining the structure and vocabulary space as a product 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 generated by the dialog act and the main goal. Structure tags and vocabulary were collected corresponding to the SV space. For example (Fig. 3), structure tags include PRP, VB and VB as part of speech tags and [loc_name]

$\boldsymbol{S}_{SV}$: 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 $W_{seq} = (W_1, W_2, \ldots, W_T)$

U_k : k-th generated utterance,

$k=1,\ldots,$ *Max_Generating_Number*, $U_k \in \boldsymbol{U}$

1st Phase – Generating Structures and Words given SV space

1. Repeat generating S_t based on $P_{SV}(S_{t+1}|S_t)$, until $S_T = <setence\_end>$, where $S_t \in \boldsymbol{S}_{sv}$ $t=1,2,3,\ldots,T$
2. Repeat generating W_t based on $P_{SV}(W_t|S_t)$, where $t=1,2,3,..,T$, $W_t \in \boldsymbol{V}_{sv}$
3. The generated word sequence $\boldsymbol{W}_{seq}$ is inserted into the set of generated utterance $\boldsymbol{U}$
4. Repeat *1* to *3* for *Max_Generating_Number* times, *Max_Generating_Number* is given by developers

2nd Phase – Selection by measure

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

Fig. 6. Algorithm of user utterance simulation.

as a component slot tag. The vocabulary includes I, want, to, go, and [loc_name]. Every named-entity word in the vocabulary is replaced with its category name. In this way, the structure tags and vocabulary can be collected for each SV space from the dialog logs. For the given SV space, probability distributions were estimated for statistical user utterance simulation using a training process. For each space, the tag transition probability $P_{SV}(S_{t+1}|S_t)$ and emission probability $P_{SV}(W_t|S_t)$ were estimated for each space, and the structure tags set $\boldsymbol{S}_{SV}$ and vocabularies $\boldsymbol{V}_{SV}$ were collected. Back-off bigram estimation is used for the transition and emission probability, and smoothed by the Kneser–Ney method (Kneser and Ney, 1995).

A two-phase user utterance generation algorithm was devised (Fig. 6). A detailed explanation of Fig. 6 is given in the following subsections.

3.3.1. First phase – Generating structure and word sequence

The structure tag S_1 is generated based on the probability of $P_{SV}(S_1|<$sentence start$>)$ and then S_1 influences the generation 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>. The structure tag transition is assumed to have 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. 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 this research, a thousand utterances were generated for a given intention, and then the appropriate utterances selected from them. Selecting natural utterances from the generated utterances requires an automatic evaluation metric.

3.3.2. Second phase – Selection by the BLEU measure

The naturalness of the generated utterances was measured with 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

	utterance	intention
U1	Where are Chinese restaurants?	DA=wh_question, MG=search_loc, CS=loc_keyword[chinese]
S1	There are Buchunsung and Idongbanjum in Daeidong, and Hongunbanjum and Wangson in Hyojadong.	SA=inform(target, address)
U2	Daeidong	DA=say, MG=none, CS=loc_address[Daeidong]
S2	There are Buchunsung and Idongbanjum in Daeidong. Which one do you want?	SA=inform(target, address), specify(loc_name)

Fig. 7. A fragment of simulated dialog from the car navigation domain. DA = dialog act, MG = main goal, CS = component slot and SA = system act.

reference sentences which have been translated by a human. Evaluation of the user utterance generation is the same task as 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. 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. The details of BLEU metrics and evaluating simulated user utterances are discussed in Section 4.1. SWB was used to select the top 20-best generated utterances and the utterance simulator returns a final corresponding generated utterance by selecting one of them randomly.

3.4. Illustration of user simulation process

This section illustrates the detailed processes of user simulation which were described in previous sections with a fragment of simulated dialog. Fig. 7 shows the fragment of simulated dialog example, and Fig. 8 shows the detail process conducted for generating the dialog. We translated a Korean user simulation process to English for clear understanding.

First, the user simulator analyzes the current discourse context and makes it a feature vector. Since no discourse was conducted until the time of first user turn (U_1), most discourse context features take 'none' values except the OTHER_INFO feature. User-related information such as the user's favorite food genre and favorite area were provided to the intention simulator. The intention simulator constructs feature vector DI_1 using the features and calculates $P(UI_1|DI_1)$ over all user intention sets. The intention simulator selects the user intention UI_1 based on the probability distribution; in Fig. 8, wh_question + search_loc + [loc_keyword] was selected.

Since the intention simulator generates the next user intention, which has wh_question as dialog act, search_loc as main goal and loc_keyword as component slot, the utterance simulator now tries to generate proper utterances to express the given intention. The SV space is wh_question + search_loc. The structure to structure transition probability and structure to word emission probability are iteratively used to generate structures and word sequences in the first phase. In the second phase, the generated utterances are rescored with SWB scores, and random utterances among the top *n*-best are returned. In Figs. 7 and 8, the utterance for UI_1 is "Where are Chinese restaurants?"

Like this, the intention simulator analyzes the discourse contexts, calculates the user intention probabilities and returns the next user turn; then the utterance simulator generates an appropriate surface for the given user intention. The generated user utterance is fed to the ASR channel simulator to add speech recognition errors.

3.5. ASR channel simulation

To test our user simulator with the previously developed dialog system, we developed a simple ASR channel simulator based on linguistic knowledge. ASR channel simulation generates speech recognition errors which might occur in the real speech recognition process. The ASR channel simulator allows the developer to set the simulated word error rate (WER) between 0 and 1.

To avoid the poor assumption of a globally fixed error rate, ASR error simulation based on phonetic confusions has been explored by other researchers. In this approach, word sequences are converted to phone

1-1) Simulate UI_1 - *Intention Simulator*
Feature for feature vector DI_1 PREV_1_SYS_ACT=none PREV_1_SYS_ACT_ATTRIBUTES=none PREV_2_SYS_ACT=none PREV_2_SYS_ACT_ATTRIBUTEs=none SYSTEM_HOLDING_COMP_SLOT=none OTHER_INFO(user_favorite_food_genre)=ChineseFood OTHER_INFO(user_favorite_area)=Daeidong Feature vector DI_1 is constructed by activating the above feature Calculate $P(UI_1 \mid DI_1)$ P(request+guide_loc+[loc_keyword] $\mid DI_1$) = 0.54 P(wh_question+search_phone_numer+[loc_ref] $\mid DI_1$) = 0.13 P(wh_question+search_loc+[loc_keyword] $\mid DI_1$) = 0.15 {*selected} P(wh_question+search_phone_numer+[loc_address,loc_name] $\mid DI_1$)= 0.02 … Randomly select user intention based on the probability distribution
1-2) Simulate user utterance given UI_1 - *Utterance Simulator*
A: generate structure sequence in SV[wh_question+search_loc] space. generate word sequence in SV[wh_question+search_loc] space. Repeat A-process many times to get candidate utterances Rescore the candidate utterances with SWB score and return a random utterance among the top n-best
2-1) Simulate UI_2 - *Intention Simulator*
Feature for feature vector DI_2 PREV_1_SYS_ACT=inform PREV_1_SYS_ACT_ATTRIBUTES=target,address PREV_2_SYS_ACT=none PREV_2_SYS_ACT_ATTRIBUTEs=none SYSTEM_HOLDING_COMP_SLOT=[loc_keyword] OTHER_INFO(user_favorite_food_genre)=ChineseFood OTHER_INFO(user_favorite_area)=Daeidong Feature vector DI_2 is constructed by activating the above feature Calculate $P(UI_1, UI_2 \mid DI_1, DI_2)$, where UI_1=wh_question+search_loc+[loc_keyword] $P(UI_1$, wh_question+search_loc+[loc_addrses] $\mid DI_1, DI_2$) = 0.09 $P(UI_1$, wh_question+search_loc+[none] $\mid DI_1, DI_2$) = 0.21 $P(UI_1$, say+none+[loc_address] $\mid DI_1, DI_2$) = 0.43 {*selected} $P(UI_1$, statement+guide_loc+[route_type] $\mid DI_1, DI_2$) = 0.02 … Randomly select user intention based on the probabilty distribution
2-2) Simulate user utterance given UI_2 - *Utterance Simulator*
A: generate structure sequence in SV[say+none] space. generate word sequence in SV[say+none] space. Repeat A-process many times to get candidate utterances Rescore the candidate utterances with SWB score and return a random utterance among the top n-best

Fig. 8. Detail intention and utterances simulation process for the fragment dialog example in Fig. 7.

sequences using a pronunciation dictionary or Grapheme-to-Phoneme module and confusions are then generated using a set of probabilistic phoneme conversion rules (Deng et al., 2003), a handcrafted phone confusion matrix (Pietquin, 2004), a weighted finite state transducer (Fosler-Lussier et al., 2002; Stuttle et al., 2004), or fragment-to-fragment alignment-based confusion models (Schatzmann et al., 2007b). These studies have shown promising results, but they require the collection of large amounts of training data consisting of

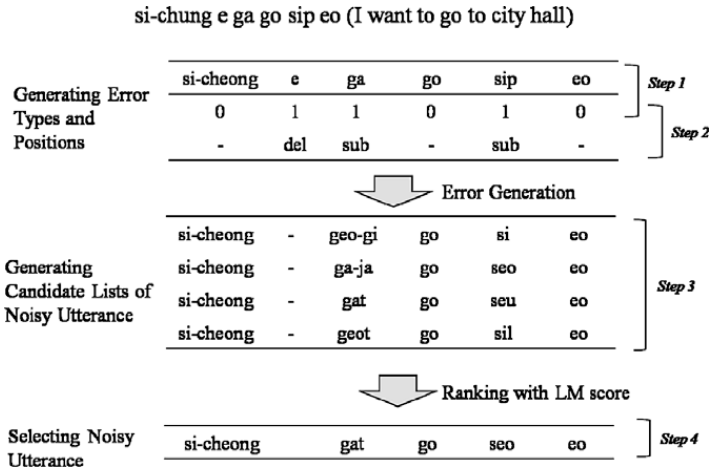

Fig. 9. Example of ASR channel simulation.

recognized results and the corresponding transcribed results for modeling context-dependent phone confusions; this is often expensive.

A computationally less expensive word-level error simulation method has been suggested by Pietquin and Dutoit (2006). They calculated acoustic distance using linguistic heuristic knowledge. In this research, we also use phone confusion models 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 similar to Pietquin and Dutoit (2006). The problem of finding an acoustic distance can be considered as an alignment problem between two sequences of symbols. In this work, we used the dynamic local alignment algorithm of Needleman and Wunsch (1970) for both syllable and phoneme sequence alignment. The phone confusion models are used as weight for finding similar phoneme and syllable sequences using alignment rather than for directly producing errors.

The developed 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 (see Fig. 9 for a Korean language example.).

In the first step, the WER is used to determine the positions of erroneous words. Each word is randomly assigned a number between 0 and 1. If this number is between 0 and the WER, the word is an Error Word (1); otherwise it is a Clean Word (0).

In the second step, ASR error types (substitution, deletion, insertion) are generated for the error-marked words based on the error type distribution. Greenberg et al. reported the speech recognition error types for eight English speech recognizers (Greenberg et al., 2000). According to the results, the average error proportion of substitution, deletion and insertions for eight participants was about 6.25:1.98:1. This research assumes that the Korean speech recognition error type distributions are similar to those of English.

In the third step, various types of ASR errors are generated. Words corresponding to the deletion errors are simply removed from the utterance. In the case of insertion error, one word is randomly selected from the pronunciation dictionary and inserted before the error-marked word. Substitution errors, however, require a more complex process.

To select a substitutable word, the marked error word is compared with the words from the pronunciation dictionary which are similar in syllable sequence and phoneme sequence. First, the final word sequence from the user simulator is converted into a phoneme sequence using a Grapheme-to-Phoneme (G2P) module (Lee et al., 2006). Then, the part of the phoneme sequence which corresponds to the error-marked word is extracted from the entire phoneme sequence. The reason for extracting the target phoneme sequence is that the G2P results vary between the boundaries of words.

Then, we separate the marked word into syllables and compare the syllable and phoneme level similarity to other words in the pronunciation dictionary. We calculate a similarity score which interpolates syllable and phoneme level similarity using following equations.

$$\text{Similarity} = \alpha^* \text{ Syllable\_Alignment\_Score} + (1-\alpha)^* \text{ Phoneme\_Alignment\_Score}, \quad \text{where } 0 \leqslant \alpha \leqslant 1$$

We used the dynamic local alignment algorithm of Needleman and Wunsch (1970) for both syllable and phoneme sequence alignment. In this research, α was set to 0.5 heuristically. This alignment algorithm requires weight matrices. The weight matrices used were vowel and consonant confusion matrices which were calculated based on the Korean manner of articulation. Each cell of the given consonant and vowel confusion matrices represents the similarity between consonants and vowels, respectively. The matrices are defined by heuristics learned from linguistics such as the following:

– Consonant confusions
  - According to the place of articulation, consonants are classified as labial, coronal, palatal, velar, and glottal; we assume that the closer the place of articulation is, the greater the confusion value is too. For example, the consonants k[k^h](ㅋ), t[t^h](ㅌ), and p[p^h](ㅍ) are articulated in the same manner (stop and aspirated), but at different places (velar, coronal, and labial). In our assumption, p[p^h] (labial) is closer to t[t^h] (coronal) than [k^h] (velar).
  - According to the manner of articulation, consonants are classified into two groups: (1) stop, plosive, fricative, nasal, and liquid, (2) aspirated, hardened, or otherwise; lower confusion values were assigned to the consonants with the different types of articulation. For example, the consonants n[n](ㄴ), s[s](ㅅ), and ss[s'](ㅆ) are articulated on the same place (coronal), but in different manners (nasal/not aspirated, fricative/not aspirated, and fricative/aspirated). Therefore n[n] and s[s] were assigned high confusion values as well as s[s] and ss[s'], but n[n] and ss[s'] had low confusion values.

– Vowel confusions
  - According to the place of tongue, vowels are classified into front, back, high and low; we assume that the closer the tongue location is, the greater the confusion value is.
  - According to the shape of lips, vowels are classified as rounded, semi-rounded, and unrounded; we assigned different confusion values based on their degree of similarity.
  - Confusion values are defined based on the relative distance of the vowels in the vowel chart.[1]

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

4. User simulation evaluation metrics

Evaluating user simulation quality helps developers judge whether the simulated dialogs are useful for evaluating a spoken dialog system. This research introduces several automatic evaluation metrics for measuring the qualities of intention simulation and utterance simulation.

4.1. Evaluation metrics for utterance simulation

Section 3.3.2 briefly introduced the BLEU metrics for evaluating simulated user utterances. Evaluation of the user utterance generation is similar to the evaluation in SMT. In SMT, translated candidate sentences are evaluated by comparing them to semantically equivalent human-translated reference sentences, while simulated utterances should be evaluated by comparing them to semantically equivalent human utterances in utterance simulation. Therefore, the BLEU score can be adopted 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

[1] The International Phonetic Association – Vowel Chart (http://www.arts.gla.ac.uk/IPa/vowels.html).

BLEU metric to compare one generated utterance with several reference utterances. The brief computation of BLEU is as follows (see (Papineni et al., 2001) for more detail):

$$p_n = \frac{\sum_{ngram \in G} Count_{clip}(ngram)}{\sum_{ngram \in G} Count(ngram)}$$

where $\boldsymbol{G}$ is a generated utterance, and

$$Count_{clip}(ngram) = \min(count, maximum\_reference\_count)$$

where count is the number of matched n-gram tokens between the generated utterance and the reference utterances, and maximum reference count is the maximum total count in the reference utterances. Next, let g be length of the generated utterance and r be of the effective reference utterance. The brevity penalty BP is computed as follows:

$$BP = \begin{cases} 1 & \text{if } g > r \\ e^{(1-r/g)} & \text{if } g \leqslant r \end{cases}, \quad \text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} \frac{1}{N} \log p_n\right).$$

BP is used to penalize unusually short utterances compared with reference utterances since short generated utterances tend to have higher n-gram precision. In this research, we use $N = 4$.

The generated utterances were rescored with the SWB score. After the first phase, the generated utterances have both structure and word sequence. The naturalness of a generated utterance is measured through both structural and lexical naturalness. The gram in the BLEU calculation of SMT is usually 'word'; however, the grams in the BLEU calculation of utterance simulation are both structural tags and words. Structure_Sequence_BLEU was calculated with the generated structure tag sequences instead of word sequences with the reference structure tag sequences of the SV space in the BLEU calculation process. Likewise, the Word_Sequence_BLEU is calculated by measuring the BLEU score using the generated words sequence with the reference word sequences of the SV space. SWB is calculated as:

$$\text{SWB} = \beta^* \text{ Structure\_Sequence\_BLEU} + (1-\beta)^* \text{ Word\_Sequence\_BLEU}, \quad \text{where } 0 \leqslant \beta \leqslant 1$$

In this study, β was empirically set to 0.2 since the Korean language is an inflected language which is relatively free in terms of the structural grammar. SWB score indicates the structural and lexical naturalness of the simulated utterance compared to semantically equivalent human utterances. This score can be used to select a desirable simulated utterance or to assess the overall utterance simulation quality by averaging the scores.

The SWB score can be also used for evaluating the performance of the utterance simulator. The average of the SWB score of simulated utterances in the simulated dialogs can indicate the overall quality of the utterance simulator.

4.2. Evaluation metrics for intention simulation

There are no generally accepted criteria as to what constitutes a good user simulation model in dialog systems. In our view, a good user model should be able to generate "human-like" behavior and dialog that make sense while interacting with a system policy. Several automatic metrics such as precision, recall and accuracy are introduced for evaluating the performance of intention modeling. Precision and recall are a common measure of 'goodness' in user modeling (Zukerman and Albrecht, 2001) and were first used in evaluating user simulation models in dialog systems by Schatzmann et al. (2005). To compare simulated intention with real responses given by users in the same context, expected accuracy, expected precision and expected recall were proposed by Georgila et al. (2005).

4.2.1. Discourse BLEU

The basic approaches of the above mentioned metrics examine how close simulated user intention is to real user intention in the same context. In other words, the target of the metrics is a simulated turn itself rather than a sequence of simulated turns. This research, however, evaluates intention simulation by measuring the overall intention sequence's goodness at the dialog level rather than focusing on the individual turn level.

D-BLEU (Discourse-BLEU) score is introduced as an intention simulation metric which can capture the similarity of simulated dialogs and real human–machine dialogs. D-BLEU is a BLEU score modified to capture the dialog level naturalness. It is the geometric mean of the *n*-gram precisions with a brevity penalty. Here, gram is the symbol of both user intention and system intention. For example, a simulated dialog has a UI1 → SI1 → UI2 → SI2 (UI: user intention, SI: system intention) intention sequence. In this example, 1-gram corresponds to {UI1, SI1, UI2, SI2}, 2-gram corresponds to {<dialog_start>-UI1, UI1-SI1, SI1-UI2, UI2-SI2, SI2-<dialog_end>} and 3-gram corresponds to {<dialog_start>-UI1-SI1, UI1-SI1-UI2, SI1-UI2-SI2, UI2-SI2-<dialog_end>} and so on. The reference intention sequences are collected from the reference human–machine dialogs extracted from the same dialog goal. Notice that the reference in the utterance simulation metric was the semantically equivalent utterances which have same intention, while the reference in the intention simulation metric was the intention sequences extracted from the dialogs which have the same dialog goal.

D-BLEU is calculated using the BLEU process by handling the user and system intention symbol as words. Another good point of using D-BLEU is that it considers not only *n*-gram precision but also the length of dialog naturally through the brevity penalty. During the simulation unusually short or long dialogs are simulated occasionally since there can be malfunctions both in the user simulator and dialog system. D-BLEU penalizes unusually short dialogs with the brevity penalty. In the case of an unusually long dialog, the *n*-gram precision decreases as the dialog turns increase. D-BLEU scores range from 0.0 to 1.0 with higher values indicating more similar dialogs.

4.2.2. Kullback-Leibler divergence

Cuayahuitl et al. (2005) proposed KL-divergence as a dialog similarity. They used an Input–Output Hidden Markov Model (IOHMM) for intention modeling. They build two IOHMM's from real dialogs and simulated dialogs, and then measured the distance between the two IOHMMs.

Since CRF in our intention modeling is a probabilistic model, we can use KL-divergence as a dialog similarity. The difference between D-BLEU and KL-divergence is that D-BLEU describes the similarity of a dialog to corresponding real dialogs, and KL-divergence tells the similarity or distance of all simulated dialogs from real dialogs in probability distributions.

This research uses KL-divergence as an indicator to tell how the simulated dialogs vary from real dialogs. Since the CRF is the conditional model, conditional KL-divergence is calculated as follows:

$$D_{KL}(p\|q) = \sum_{\mathbf{x}} p(\mathbf{y}' \mid \mathbf{x}; \Lambda_p) \log \frac{p(\mathbf{y}' \mid \mathbf{x}; \Lambda_p)}{q(\mathbf{y}' \mid \mathbf{x}; \Lambda_q)} \tag{2}$$

where $\mathbf{y}' = \{y'_1, y'_2, \ldots, y'_T\}$, y'_i = true label, and Λ is a parameter vector.

Cuayahuitl et al. used the KL-divergence values directly to calculate dialog similarity (Cuayahuitl et al., 2005). However, in this research the KL-divergence values were normalized with the number of data, since the KL-divergence tends to be affected by the number of data ($\mathbf{x}$ in Eq. (2). The average of normalized $D_{KL}(p\|q)$ and $D_{KL}(q\|p)$ give the dialog similarity. Here, p is the CRF trained from human–machine dialogs, and q is the CRF trained from simulator-machine dialogs.

$$\text{Average of normalized KL-divergence} = \frac{D_{KL}(p\|q) + D_{KL}(q\|p)}{2^* \text{ (num of data)}}$$

If the average of normalized KL-divergence is close to zero, two CRF's have similar probability distributions, indicating that all the simulated dialog examples are actually similar to the real human–machine dialogs.

5. Experiments

Our user simulation method was evaluated through a case study of the Korean spoken dialog system for the car navigation domain to verify its feasibility as well as the evaluation metrics. In addition to the car navigation domain, we also applied the user simulation methods to a building guide robot domain to show that the proposed methods are actually portable to other domains.

5.1. Case study on car navigation domain

One hundred dialog examples from a real user and dialog system in the car navigation domain were used to train the user intention and utterance simulator. The SLU method of Jeong and Lee (2006), and dialog management method of Lee et al. (2005) were used to build the dialog system. After training, simulations collected 5000 dialog samples at each WER setting (WER = 0.0–0.4).

User intention and utterance simulation quality was verified by three groups of two judges, each of which evaluated 200 randomly chosen simulated dialogs and utterances (WER = 0.0). Ai and Diane observed that it is difficult for human judges to rate on the 5-point scale and the agreements among the judges are fairly low, so they tried to rescale to a 3-point scale (Ai and Diane, 2008). Similarly this study used a 3-point scale for both dialog and utterance level evaluation. First, judges evaluated a dialog with the 3-point scale and then evaluated the utterances of the dialog with a 3-point scale.

5.1.1. Experiments on user intention simulation

The human judgments and corresponding D-BLEU score of each judge group were analyzed to check the relationship between D-BLEU score and human judgment and verify the simulated dialog's quality. Fig. 10 shows the question asked for each simulated dialog.

Fig. 11 shows the average of human judgment, average of D-BLEU score and kappa values for each group. The inter evaluator agreement (kappa) is 0.49, 0.36 and 0.6 for each group respectively, showing moderate agreement. The overall average of human evaluation of 600 dialog examples was over 2.3, which means that judges had positive reactions to the quality of simulated user intention sequences.

Fig. 12 also shows that the D-BLEU score and human judgment have similar tendencies. For example, the average of D-BLEU scores for the dialog examples which were judged as 1 by human judges in A-group is 0.78, and the average of D-BLEU scores for the dialog examples judged as 2 and 3 increases to 0.82 and 0.91 respectively. In other words, there is a clear linear relationship. The linear relationship between human judgments and D-BLEU scores was confirmed through correlation analysis of the dialog examples. The correlation coefficient is 0.44, which is significant at $p < 0.01$. The results suggest that the dialog examples which have D-BLEU scores over 0.85 can be treated as natural dialog.

Fig. 13 shows another proof of D-BLEU metrics. We separated 5000 simulated dialog examples into seven groups according to the D-BLEU score, and then trained the user intention CRF models for each dialog group. For example, CRF 0.4 means the CRF model which is trained using the dialog examples where $0.4 \leqslant$ D-BLEU < 0.5. The reason the starting point of the x-axis is CRF 0.4 is that the minimum D-BLEU of the simulated examples was 0.47. The average of normalized KL-divergence was calculated for each CRF model with the CRF trained from human/machine dialogs. Fig. 13 shows these averages decreased as

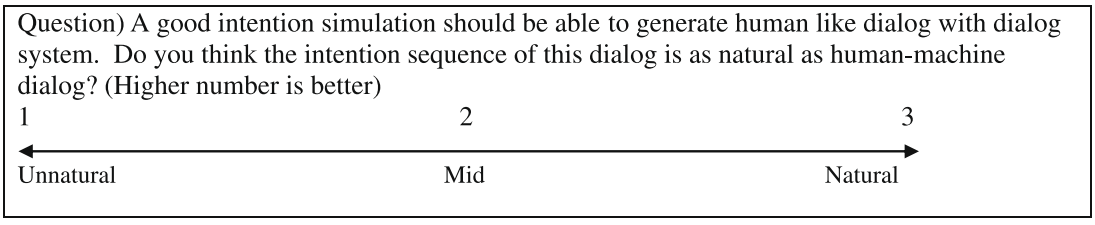

Fig. 10. Intention sequence level question.

	A-group	B-group	C-group	overall
average of human evaluation	2.38	2.31	2.29	2.32
average of D-BLEU	0.86	0.87	0.85	0.86
kappa	0.49	0.36	0.6	-

Fig. 11. Average score of human evaluation and D-BLEU score for each judge group.

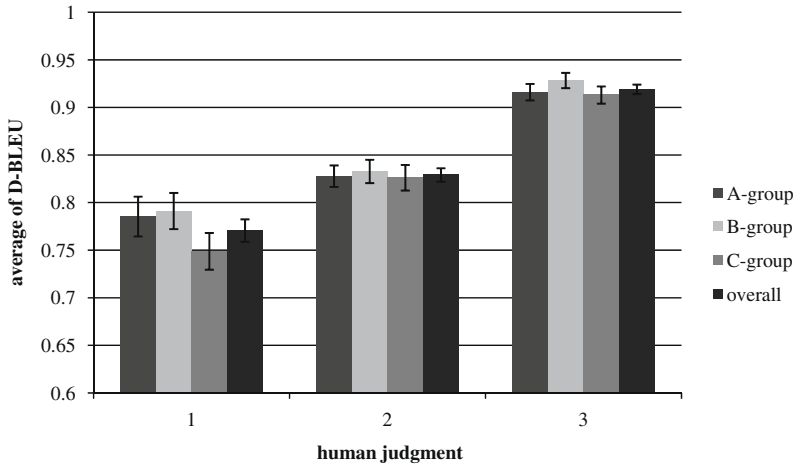

Fig. 12. Relationship between D-BLEU score and human judgments on intention simulation. (Error-bars indicate the standard errors.)

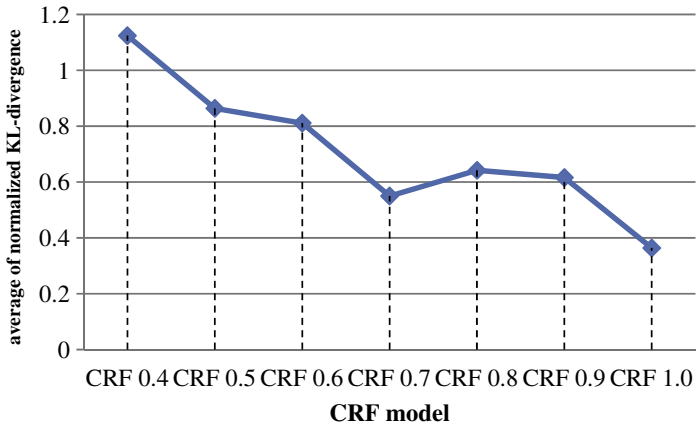

Fig. 13. Relationship between average of normalized KL-Divergence and CRF model. CRF 0.4 means the CRF model trained from the dialog examples where $0.4 \leqslant$ D-BLEU < 0.5.

the D-BLEU increased to one. It means that the dialog examples which have higher D-BLEU scores are more similar to the human–machine dialogs in terms of relative entropy.

We also examined whether the user intention simulator correctly reproduces the statistical properties of real user dialog acts and main goals. Fig. 14a and b shows the relative frequency of the user dialog acts and main goals in real and simulated data respectively. As shown, the distributions over both dialog acts and main goals generated by the user intention simulator are similar to those of real users.

To compare the proposed approach with another intention simulation method, we implemented *n*-gram model and linear-feature combination model (Georgila et al., 2006). The *n*-gram based user intention simulation method reuses the *n*-gram model suggested by Eckert et al. (1997), but with *n* ranging from 2 to 5 in order to cover longer spans of dialog history. Unlike the advanced *n*-gram model of Georgila et al. (2006) which is based on the *Information States* framework (Bos et al. 2003), this research used previous user intentions and system intentions as history information.

The linear-feature combination method maps a state s to a vector of real-valued features $f(s)$. Most of the features are binary indicating the presence or absence of a piece of information. Supervised learning is used to estimate a set of weights w_a for each action a which describe how "useful" each vector element of $f(s)$ is for predicting a. This research used a limited-memory quasi-Newton code for optimization (L-BFGS) (Liu and Nocedal, 1989). The probability distribution can be represented as follows:

$$\widehat{P}(a \mid s) = \frac{\exp(f(s)^T w_a)}{\sum_a \exp(f(s)^T w_a)} \tag{3}$$

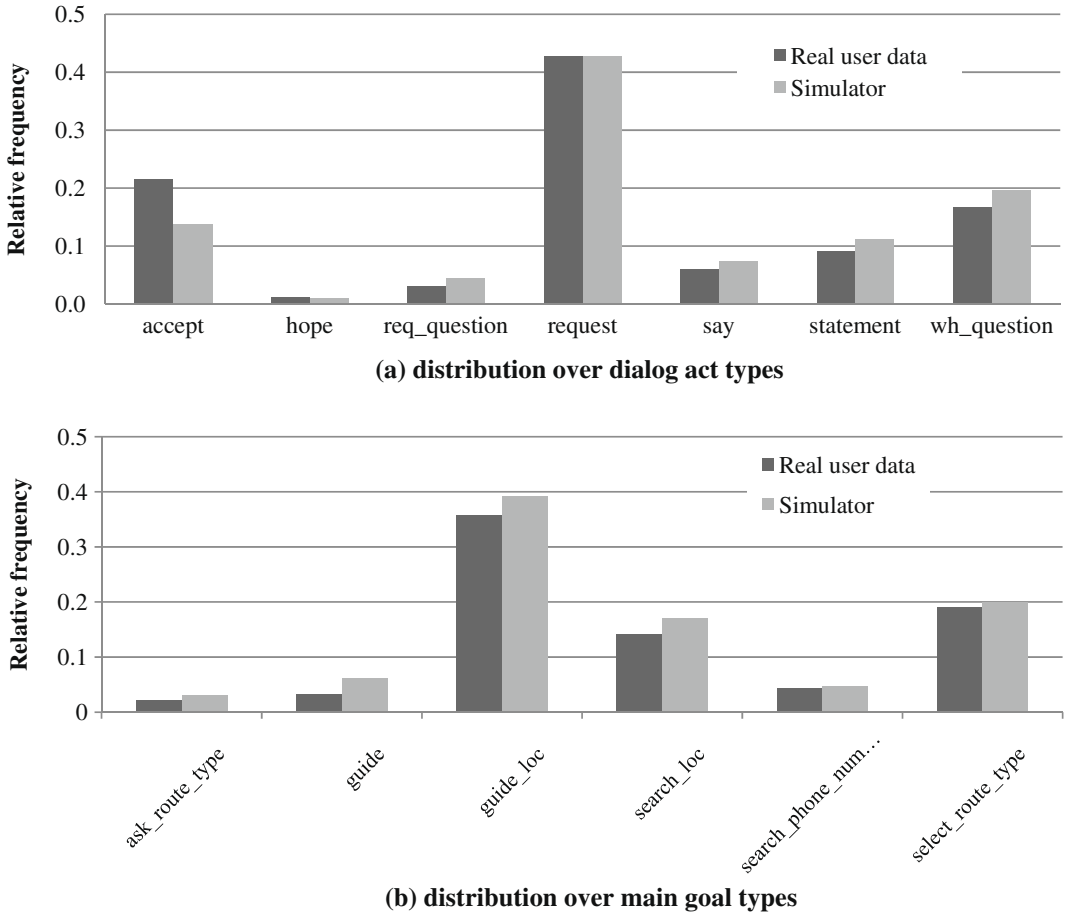

Fig. 14. Distribution over both dialog act and main goal.

We used same features that we used for CRF based intention simulation (Fig. 4) for the linear-feature combination method rather than using the features of Georgila et al. for the pair comparison.

Fig. 15 shows the comparison of the proposed CRF based user intention simulator with the *n*-gram and linear-feature combination intention simulator. We collected a thousand simulated dialog examples from each model. Fig. 15a shows the discourse BLEU between models. The linear-feature combination and proposed approach have much higher D-BLEU values than the *n*-gram model. This means these methods generate more natural intention sequences. We found the same tendency in both average turn length (Fig. 15b) and task completion rate (Fig. 15c).

The *n*-gram model generates the next user intention based on only previous user and system action symbols regardless of other useful discourse information. So it does not make sense in the larger context of the dialog since all state information is neglected. This is the reason that the linear-feature combination and CRF based user intention simulator generate more natural intention sequences because both approaches consider the dialog circumstance by capturing useful evidence as features. Fig. 15 shows that the 4-gram intention model has the best result in the *n*-gram models.

Fig. 15a and c shows that the proposed CRF based intention simulation method has higher D-BLEU and TCR values than the linear-feature combination method. Both the linear-feature combination method and CRF based user intention method are similar in that they capture useful discourse information using indicating functions, and that supervised learning techniques are used. However, the target of the model is different. The linear-feature combination method's target is a user intention given current dialog information (Eq. (3)), while the proposed CRF based user intention model is trained to optimize the modeling of user intention sequences (Eq. (1)). This might be the main reason that the CRF based method generates more natural simulated dialog examples.

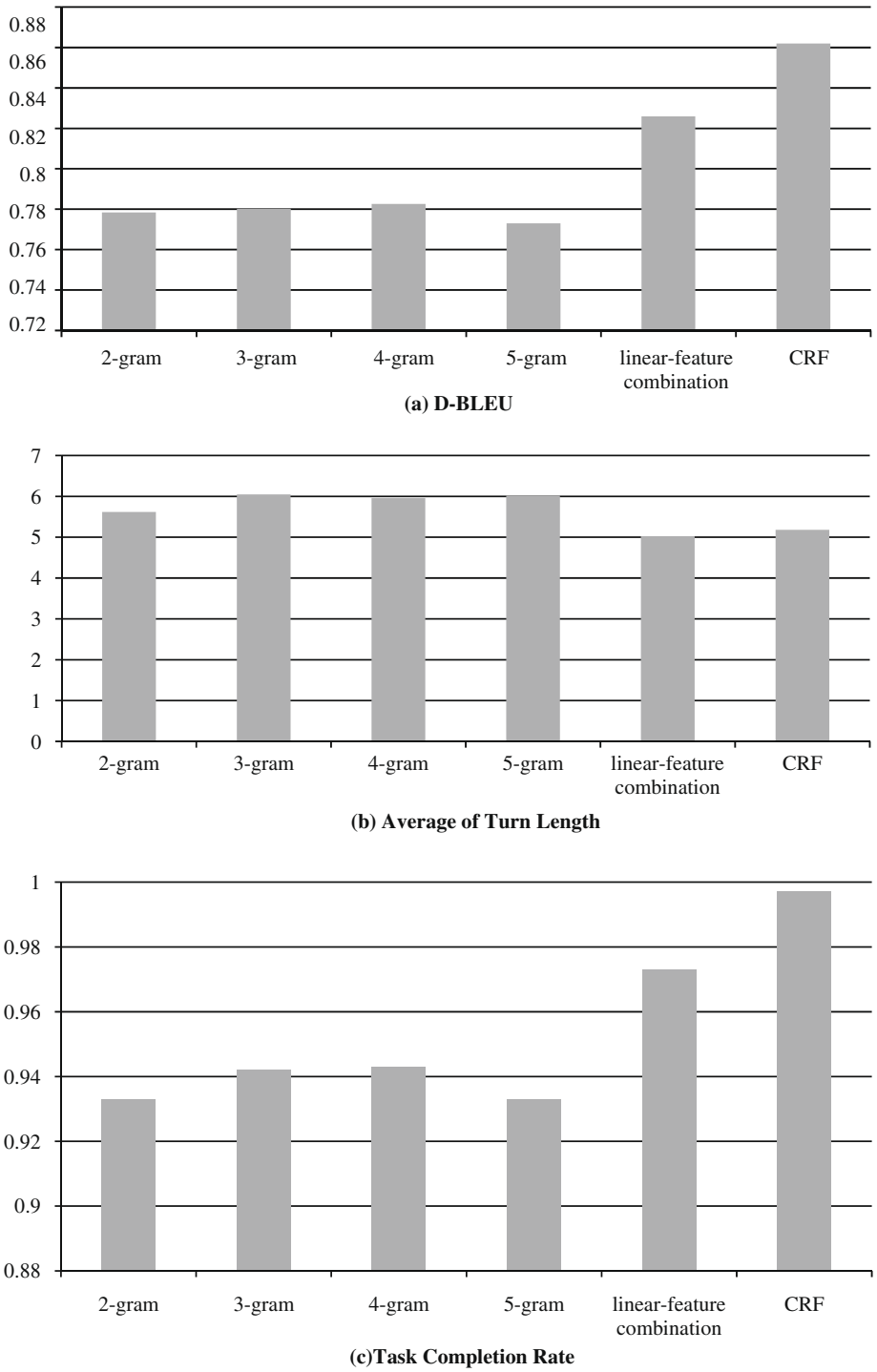

Fig. 15. Comparison CRF based intention simulator with *n*-gram and linear-feature combination intention simulator.

5.1.2. Experiments on user utterance simulation

The human judgments and corresponding SWB score of each judge group were analyzed to check the relationship between SWB score and human judgment and verify the simulated utterance's quality. Fig. 16 shows the question asked for each simulated utterance.

Fig. 17 shows the average of human judgment and kappa values for each group. The inter evaluator agreement (kappa) is 0.49, 0.23 and 0.54 for each group respectively, showing moderate agreement except for B-group. The overall average of human evaluations of 3042 simulated utterances was over 2.6 which mean that judges had positive reactions to the quality of simulated user utterances.

Question) A good utterance simulation in a dialog should express the given intention as naturally as a human. Do you think this utterance is natural? (higher number is better)

1 2 3

Unnatural Mid Natural

Fig. 16. Utterance level question.

	A-group	B-group	C-group	overall
average of human evaluation	2.66	2.57	2.64	2.62
average of SWB	0.81	0.81	0.81	0.81
kappa	0.49	0.23	0.54	-

Fig. 17. Average score of human evaluation and SWB score for each judge group.

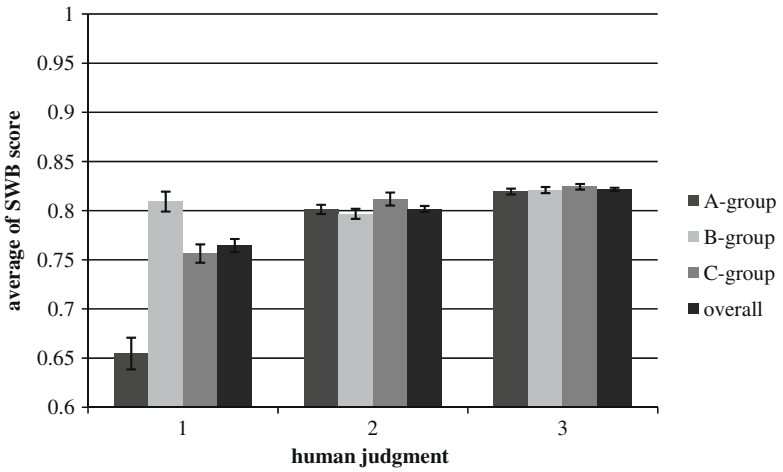

Fig. 18. Relationship between human judgments and SWB scores for each judge group. (Error bars indicate the standard errors).

Fig. 18 shows the relationship between average of SWB score and human judgments. As in intention simulation verification, SWB scores were averaged according to the utterances which were evaluated as 1–3. Even though the average of SWB score of B-group corresponding to human judgment '1' is abnormally high, we can see that the overall trends have a certain linear relationship. The linear relationship between human judgments and SWB scores was confirmed through correlation analysis of the utterance examples. The correlation coefficient is 0.18, which is significant at $p < 0.01$. The results suggest that the utterance examples which have SWB scores over 0.8 can be treated as natural utterances.

The user utterance simulator can also be evaluated by comparing statistical properties of the simulated utterances with those of real user utterances. Fig. 19 shows the distribution over the number of words per utterance, showing that the length of the simulated utterances is similarly distributed to the length of real utterances.

5.1.3. Unseen rate of words and intention sequence simulation

Basically, data-driven user simulation generates the situations which are present in the training data. For successful user simulation, unseen situation should be also generated. To generate unseen events, this research used a probability based random selection method to simulate user utterance and intention sequences.

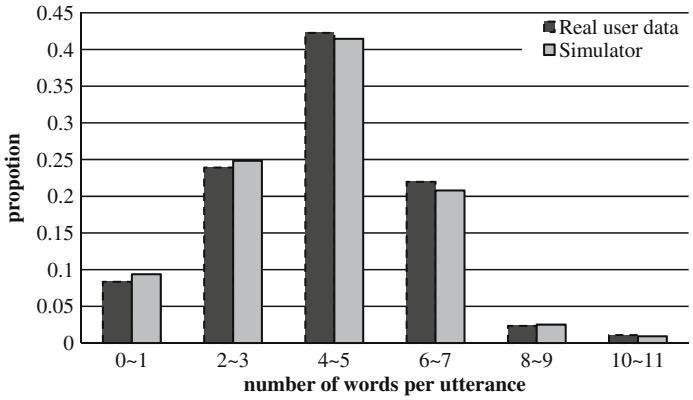

Fig. 19. Distribution over number of words per utterance.

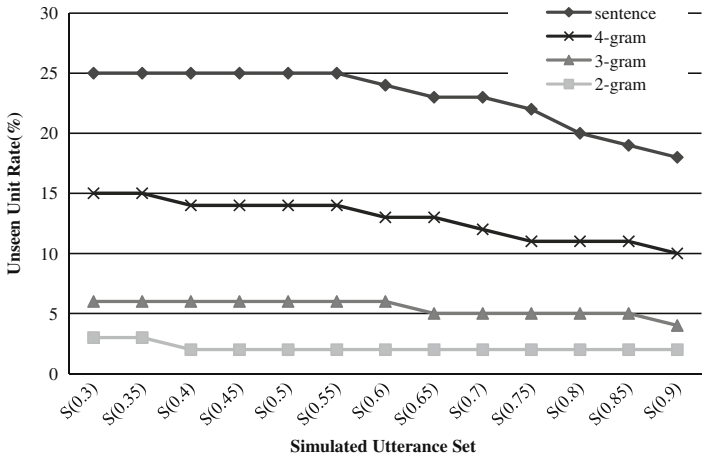

Fig. 20. Unseen unit rates in the simulated utterance. (WER = 0.0, SLU Accuracy = 0.98) S(0.7) means the set of simulated utterances which have SWB score > = 0.7.

To verify that the proposed user simulation method can simulate the unseen events, the unseen rates of units were calculated. Fig. 20 shows the unseen unit rates of utterance. 2-gram, 3-gram, 4-gram and sentence units are collected for each simulated utterance set from the 15,098 simulated utterances (WER = 0.0). For example, the simulated data set S(0.7) is the set of simulated utterances which have SWB scores greater than or equal to 0.7. Fig. 20 shows that the unseen rates of higher *n*-gram units are higher than that of lower *n*-gram units. It also shows that the unseen unit rates decrease as SWB score increase. About 25% of sentences are unseen utterances around SWB score 0.3, and about 18% of sentences are unseen utterances around SWB score 0.9.

Notice that, about 18% of sentences of S(0.9) can be considered as a natural utterance according to the SWB score and are actually unseen sentences to the training data. Also the overall spoken language understanding accuracy is over 0.9. It means that the utterances generated by our simulation method can be thought to be as not only natural but also diverse utterances.

Similar to the unseen utterance check, the unseen unit rates of intention sequence were calculated. 2-gram, 3-gram, 4-gram, over 5-gram and dialog intention sequence were collected for each simulated intention sequence from the 5000 simulated dialogs. For example, the simulated data set S(0.7) of Fig. 21 is the set of simulated intention sequence which have D-BLEU scores greater than or equal to 0.7. Here, intention sequence is the sequence of symbols of user intention and system intention such as UI1 → SI1 → UI2 → SI2 (UI: user intention symbol, SI: system intention symbol).

Fig. 21 shows the similar tendency of utterance level unseen rates. The unseen rates of higher *n*-gram units are higher than those of lower *n*-gram units. It also shows that the unseen unit rates decrease as D-BLEU score increase.

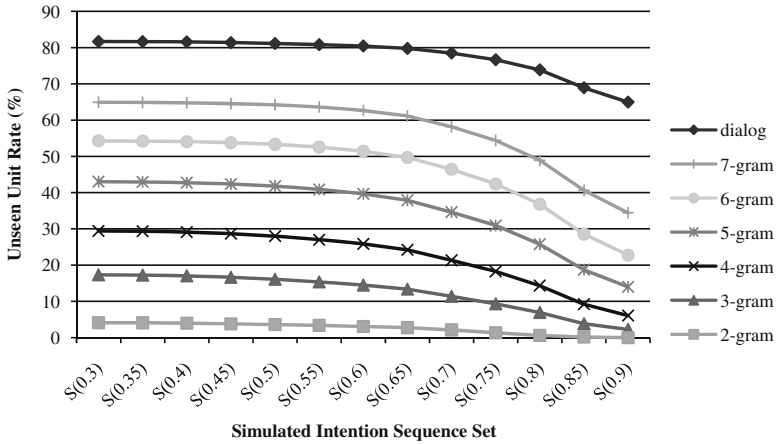

Fig. 21. Unseen unit rates in the simulated intention sequence. (WER = 0.0, TCR = 0.95) S(0.7) means the set of intention sequence which have D-BLEU score > = 0.7.

Notice the unseen rate of 2-gram along the *x*-axis. The unseen rates of 2-gram are almost zero. This means that one turn of simulated interaction such as UI → SI or SI → UI is present in training corpus. However, much higher *n*-gram units tend to be absent in the training corpus. About 65% of simulated dialogs are unseen around D-BLEU score 0.9, when 80% of dialogs are unseen around D-BLEU score 0.3. The unseen rate of intention sequence is much higher than that of utterance. Both user simulator and dialog system might malfunction during the interaction where the reference human–machine dialog data is the well-refined dialogs. The malfunctioning *n*-gram units are not present in the corpus which is the reason for high unseen rates.

Despite the high unseen rate, the interaction between simulator user and dialog system generates successful dialogs since the overall Task Completion Rate (TCR) is over 0.9. It means that the intention sequences generated by our simulation method can be thought to be not only natural but also diverse intention sequences.

5.1.4. Experiments on ASR channel simulation

The task of comparing the simulated ASR channel simulator with a real ASR system is similar to comparison between two different ASR systems. Since there is no widely accepted standard measure to compare ASR systems, we analyzed and compared the simulated utterances with recognized utterances of real ASR systems in various ways.

The ASR system that we used had substitution, deletion, insertion and word error rates as 8.7%, 9.34%, 1.23% and 19.25% respectively. The ASR is based on the HTK toolkit (Young et al., 1997) and uses an *n*-gram language model for decoding. The total number of recognized utterances is 994. As we described in section 3.5, the developer can set the word error rate and the distribution of error type (substitution, deletion, insertion) in the proposed ASR channel simulator. For the pair comparison, we set the error rates of each error type of the ASR channel simulator to the same value as those of the real ASR.

Fig. 22 shows the comparison between the real ASR and simulated ASR in terms of error rates. As we can see, the simulated ASR actually generates a similar error distribution that we wanted. Fig. 23 shows the distribution over number of syllables per substituted word by both real ASR and simulated ASR. Both systems have many substitution errors which have one or two syllables in the corresponding words. The figure shows that if the number of syllables is longer than three, then the corresponding words are usually not substituted in either system. The reason might be there is more acoustic and phoneme level information for the words which have many syllables.

To find out how similar the errors generated in the real and simulated ASR are, we compared 10 words which are frequently substituted in the real ASR system with the corresponding words generated by the simulated ASR. Fig. 24 shows the substituted words of the real and simulated ASR. Like Fig. 23, it shows that most of the frequent substation errors occurred in short words which have one or two syllables. We can find that the proposed simulated ASR actually generates similar errors. When the original words in the reference are substituted by the words which has syllable- and phoneme level similarity, both real ASR and simulated

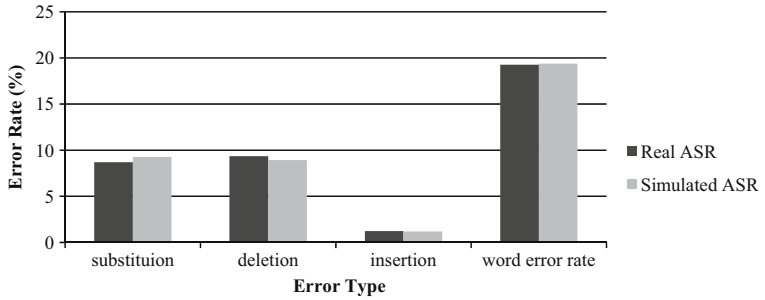

Fig. 22. Error rates on each error type between real ASR and simulated ASR.

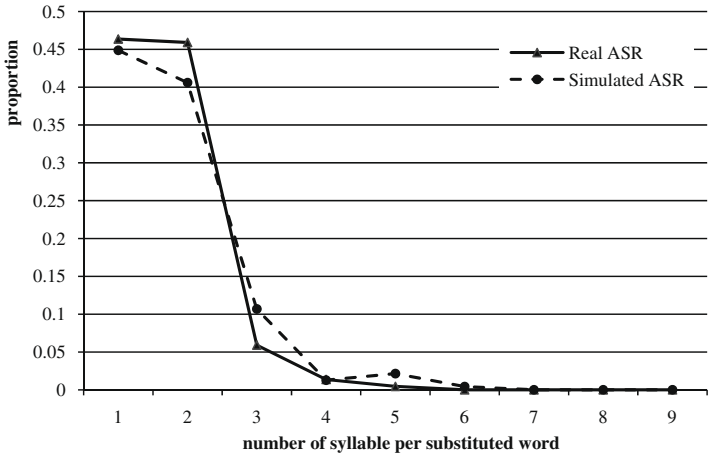

Fig. 23. Distribution over number of syllables per substituted word by real and simulated ASR.

word in reference	substituted word by real ASR	substituted by simulated ASR
i	**e**, gunggeul-lae, **iss**, sa-ga-i, jal	**e**, **iss**, ya, i-ya, ui
ni	ge, **de**	e, i, **de**, ji
peu	**peun**	pa, ha, **peun**
ui	**e**	**e,** i
si-gan	**gi-leum**, i	**gi-leum,** ji-geum
i-ya	ya	yeo-yu, eo-ya, eo-yo
bwa	**bwa-la**	jwo **bwa-la**
geo-gi	**go-gi**, ui	kko-chi geo-li **go-gi** keo-pi geu-li
mu-eos	mwo-ji	mwo-yeoss
jung-sig	**tongsin**	junggug-jib, han-sig, **tongsin**

Fig. 24. Comparison the substituted words generated by real and simulated ASR. The bold terms denote the words generated by both real and simulated ASR.

ASR generate similar errors. However for some words the simulated ASR couldn't generate the same substitution error as the real ASR, such as "gunggeul-lae" in the first column in Fig. 24. When the original word "i" is substituted by real ASR to "gunggeul-lae" which is not related to the original word, the simulated ASR generates substitution errors such as "e", "iss", "ya", "i-ya" and "ui" which are similar to original word "i" in terms of syllable and phoneme. This may be because the proposed knowledge-based ASR channel simulation method does not use the actual sound information, while the real ASR uses the signal level information fully.

5.1.5. Simulator and dialog system behavior

To examine the overall behavior between the simulator and dialog system, the simulated dialog examples were analyzed in terms of dialog system evaluation. We tried to observe the dialog system's performance in

various circumstances by adjusting the WER from 0.0 to 0.5 to examine how SLU of utterances was affected by WER. Notice that since the dialog system works on the simulated noisy utterances, SLU performance can be measured by comparing the simulated user intention with the SLU result, and the WER and Sentence Error Rate (SER) measured by comparing the simulated user utterance with the noise-added utterance. SLU was quantified according to Concept Error Rate (CER). The commonly used WER, SER and CER can be defined as

$$WER = \frac{S_w + D_w + I_w}{N_w} \tag{4}$$

$$SER = \frac{S_s}{N_s}, \quad \text{and} \tag{5}$$

$$CER = \frac{S_c + D_c + I_c}{N_c} \tag{6}$$

In Eq. (4), S_w, D_w and I_w are the numbers of substitutions, deletions and insertions of words, and N_w is the number of words in the reference (Hunt, 1989). S_s and N_s of Eq. (5) are the number of substituted sentences and total reference sentences. SLU performance can be measured with CER (Boros et al., 1996). The semantic or information units are treated as concepts. To calculate the concept error rate, each concept is considered as one token that is equivalent to a word in the calculation of WER. In Eq. (6), S_c, D_c and I_c are the numbers of substitutions, deletions and insertions of concepts, and N_c is the number of concepts in the reference.

The measured WER was almost the same as that set by the developer, while SER increased more rapidly and CER increased more slowly (Fig. 25). The metrics for evaluating the dialog system were CER for the SLU performance and dialog length and Task Completion Rate (TCR) for the overall dialog system performance. Figs. 25 and 26 show the overall dialog system behavior using the user simulator and ASR channel simulator. As the WER increased, SLU performance decreased, which made the dialog length longer and TCR lower. This result is similar to the behaviors in typical real human–machine dialogs.

Fig. 27 shows a simulated dialog example. The dialogs were simulated at WER = 0.10, and the qualities of user intention and utterances are D-BLEU = 1.0 and average of SWB = 0.74, respectively. Even though the simulated utterances have some noise (average WER = 0.16), the simulator and dialog system made a good conversation.

5.1.6. Comparison of dialog managers using user simulation

The user simulator provides a chance to compare and predict the behaviors of different dialog managers on the same task more easily. We implemented three different dialog systems and connected them to the proposed user simulator to predict and analyze the system behaviors. The first dialog manager is the same one that we have used in the experiments of previous sections. It is based on the Example Based Dialog Managing (EBDM) technique which can handle only the top 1-best of ASR results (Lee et al., 2005). The second dialog manager is an extension of the EBDM technique which supports the n-best hypotheses of ASR results with an agenda

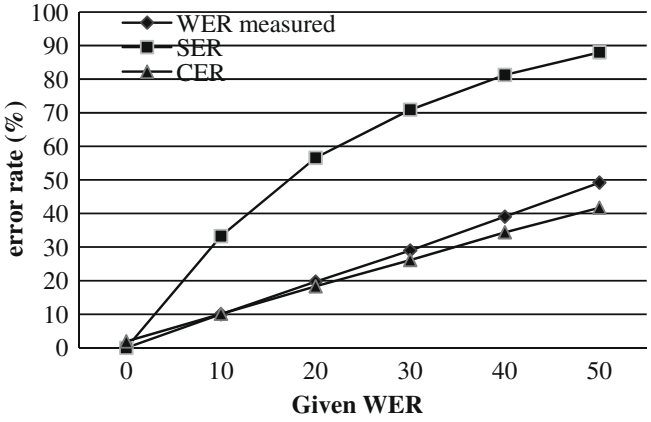

Fig. 25. Relationship between given WER and other measured error rates.

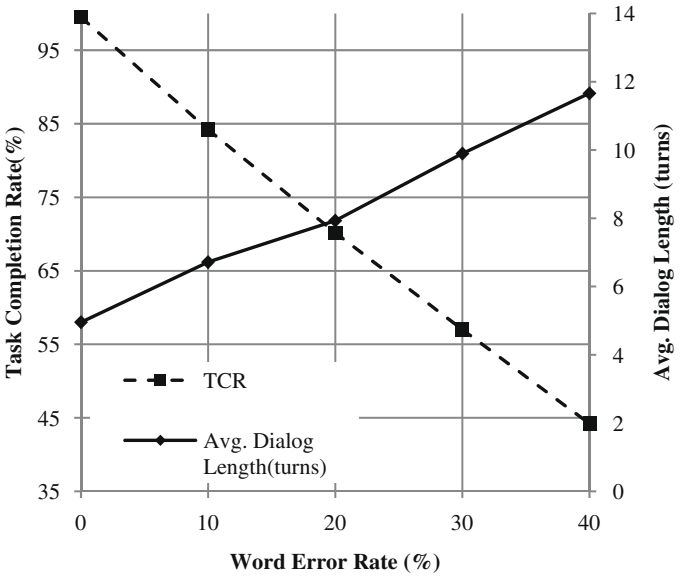

Fig. 26. Dialog simulation result on car navigation domain. (TCR = Task Completion Rate).

	Intention	Utterance	Noisy utterance	SWB, WER
U1	request, quide_loc, [loc_keyword]	il-sik ga ja (I want to go japanes restaurant.)	il-sik ga ja	0.78, 0.0
S1	inform(keyword, address)	il-si-geun dae-do-dong-e no-gwon-cho-bap, dae-i-dong-e ga-jok-jjeong, dae-i-dong-e sa-ga-i, dae-i-dong-e mi-song-il-si-gi it-sseum-ni-da. (There is Nokwonchobab in Daedodong and Gajokjung, Sagai and Misongilsik in Daeidong.)	-	
U2	statement, guide_loc, [loc_address, loc_name]	dae-i-dong ga-jok-jjeong eu-ro gap-ssi-da (Let's go Gajokjung in Daeidong)	dae-i-dong ga-jok-jjeong eu-ro (deletion)	0.64, 0.25
S2	inform(address, name)	dae-i-dong-e in-neun ga-jok-jjeong-i it-sseum-ni-da. (It is Gajokjung in Daeidong.)	-	
U3	request, guide_loc	geo-gi ro ga ja (Let's go there)	geo-gi eu-ro ga ja (substituion)	0.78, 0.25
S3	select(all), specify(routetype)	mok-jjeok-jji-ro gyeong-buk po-hang dae-i-dong-e in-neun ga-jok-jjeong-eul seon-tae-ka-syeot-sseum-ni-da. gyeong-no-tam-sae-geul wi-hae choe-dan geo-ri mit choe-jeok geo-ri-reul seon-tae-kae-ju-se-yo. (You selected Gajokjung in Daeidong. Please choose the route-type among 'shortest' or 'optimal' path.)	-	
U4	statement, guide_loc, route_type	choe-dan geo-ri ga jot get da (I prefer shortest path)	choe-dan ga jot get da (deletion)	0.52, 0.16
S4	select(routetype), confirm(address, name, routetype)	gyeong-buk po-hang dae-i-dong-e in-neun ga-jok-jjeong-eul choe-dan geo-ri-ro an-nae-hae-deu-ril-kka-yo? (Do you want shortest path to Gajokjung in Daeidong?)	-	
U5	accept, none, none	go-ma-wo (Ok. Thanks.)	go-ma-wo (no error)	1.00, 0
S5	guide()	ga-jok-jjeong-eu-ro an-nae-hae-deu-ri-get-sseum-ni-da. (I guide you to Gajokjung.)	-	

Fig. 27. Example of simulated dialog (given WER = 0.10). SWB = Structure and Word BLEU, WER = Word Error Rate. D-BLEU = 1.0, Average of SWB = 0.74, Average of WER = 0.16.

graph. (see more detail in Lee et al. (2008).) This approach supports *n*-best hypotheses in the dialog manager and keeps track of the dialog state using a discourse interpretation algorithm with the agenda graph and focus stack. The user agenda works as heuristic guide lines to increase the task completion rate. In this paper, we used the top 10-best hypotheses of ASR results. The third dialog manger is a frame-based probabilistic dialog

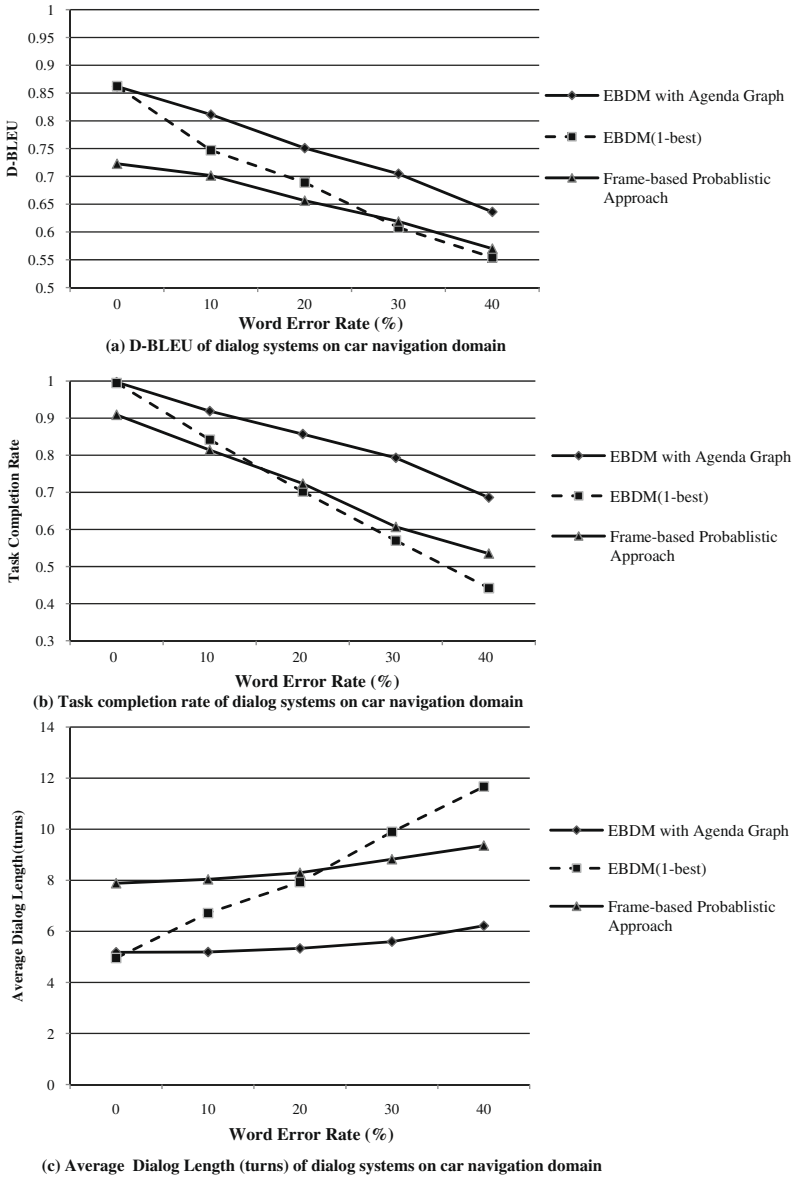

Fig. 28. Evaluation of three different dialog systems for the car navigation domain using user simulator. (EBDM means Example Based Dialog Manager).

manager. It uses the frame-based belief state representation to overcome the complexity problems of classic partially observable Markov decision processes (POMDP). (See more detail in Kim et al. (2008).) This approach employed the POMDP formalism to maintain a belief distribution over dialog states so that the system can be robust to speech recognition errors by considering the uncertainty of user input.

To compare the behaviors, we evaluated the task completion rate, discourse BLEU and average dialog length (turns) over word error rates from 0% to 40%. The results are shown in Fig. 28. As we expected, the EBDM with an agenda graph approach showed better performance in D-BLEU, TCR and average of dialog length than the original EBDM and the frame-based probabilistic approach. Apparently the heuristic agenda and top 10-best ASR hypotheses made the dialog shorter and closer to reference human–machine dialog. The frame-based probabilistic approach showed relatively lower performance than the original EBDM and EBDM with agenda graph, especially when the WER is around zero to 10%. This approach considers uncertainty of user input and all possible system actions even when there are not many speech recognition errors, and it may result in the low performance when WER is zero to 10. However, because of the uncertainty handling, it gives better results than the original EBDM when there are lots of speech recognition errors.

5.2. User simulation on robot guide domain

The proposed user simulation methods were also applied to the building guide robot domain to show domain portability. 100 dialog examples from a real user and dialog system in the building guide robot domain were used to train the user intention and utterance simulator. The SLU method of Jeong and Lee (2006), and dialog management method of Lee et al. (2005) were used to build the dialog system. After training, simulations were collected 1000 dialog samples at each WER setting (WER = 0.0–0.5). The simulated examples were used to analyze the behavior of simulator and dialog system statistically.

We examined whether the user intention simulator correctly reproduces the statistical properties of real user dialog acts and main goals. Fig. 29a and b shows the relative frequency of the user dialog acts and main goals in real and simulated data respectively. As shown, the distributions over both dialog acts and main goals generated by the user intention simulator are similar to those of real users.

The user utterance simulation was evaluated by comparing statistical properties of the simulated utterances with those of real user utterances. Fig. 30 shows the distribution over the number of words per utterance, showing that the length of the simulated utterances is similarly distributed to the length of real utterances.

The measured WER was almost the same as that set by the developer, while SER increased more rapidly and CER increased more slowly (Fig. 31). This is similar to the behavior that we found in the car navigation domain. Fig. 32 shows the overall dialog system behavior using the user simulator and ASR channel simulator. As the WER increased, SLU performance decreased, which made the dialog length longer and TCR

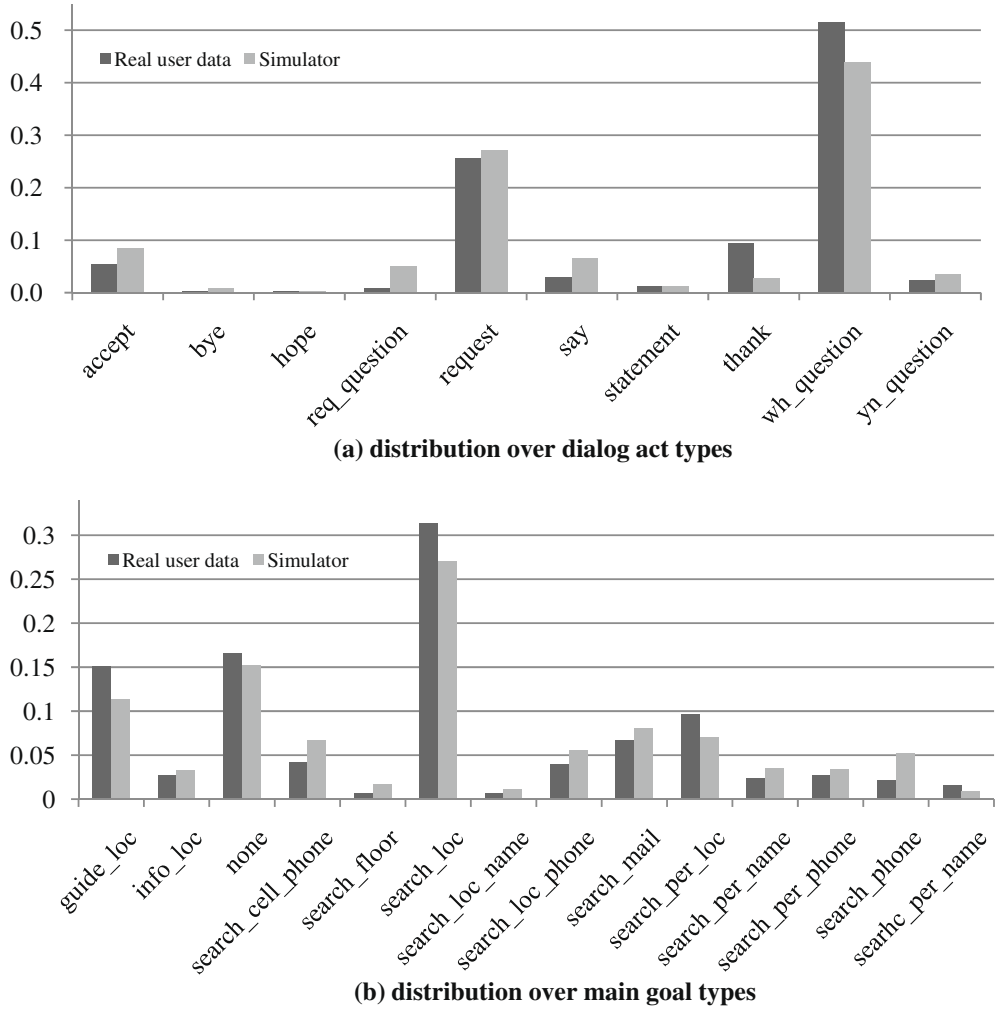

Fig. 29. Distribution over both dialog act and main goal on building guide robot domain.

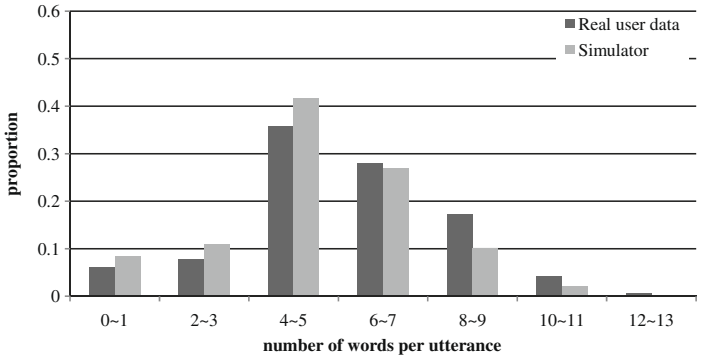

Fig. 30. Distribution over number of words per utterance.

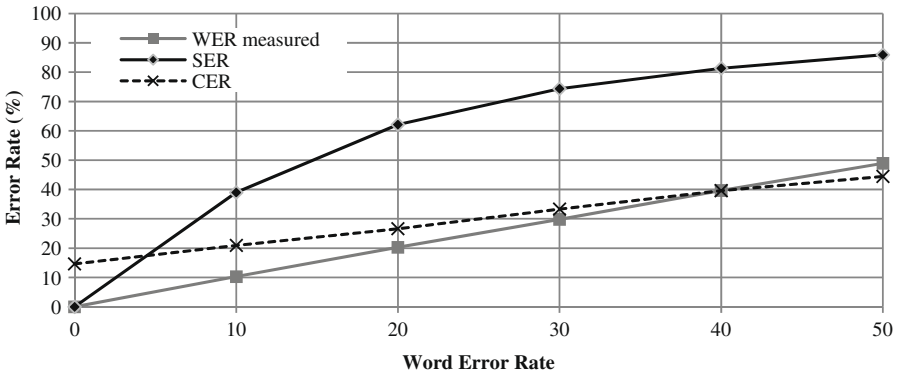

Fig. 31. Relationship between given WER and other measured error rates.

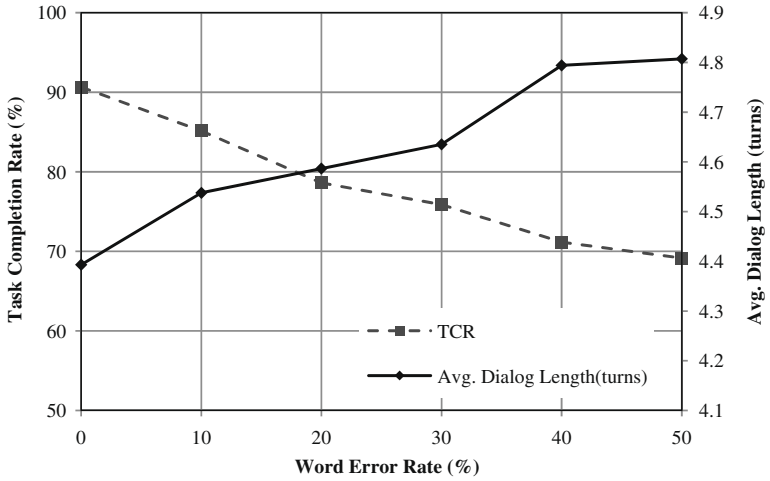

Fig. 32. Dialog simulation result on building guide robot domain.

lower. Using these graphs and dialog examples, developers could predict the dialog system's behavior and find the potential problems.

Fig. 33 shows a simulated dialog example in the building guide robot domain. The dialogs were simulated at the WER = 0.10, and the qualities of user intention and utterances are D-BLEU = 0.96 and average of SWB = 0.83, respectively.

	Intention	Utterance	Noisy utterance	SWB, WER
U1	request, search_per,[per_title]	so-jang nim bang eul ral-lyeo jwo (Let me know the room number of manger of this center.)		0.94, 0.0
S1	inform(roomnumber)	hong-gil-dong so-jang ui opi-seu-neun 201 ho im-ni-da (The room number of manager Honggildong is 201.)	-	
U2	request, search_loc_phone, [loc_ref]	geo-gi jeon-hwa beon-ho do all-yeo jwo (What's the phone number?)	geo-gi jeon-hwa beon-ho all-yeo jwo (deletion)	0.83, 0.16
S2	inform(locphone)	so-jang-sir-ui opi-seu jeon-hwa-beon-ho-neun 000-000-0000 im-ni-da. (The phone number is 000-000-0000.)	-	
U3	say, serach_mail	me-il ju-so ne-un (e-mail address?)	me-il ju-so (deletion)	0.79, 0.33
S3	inform(mail)	hong-gil-dong so-jang-ui me-il ju-so-neun aaaa@aaaa.aa imnida. (The e-mail addrses of manager Honggildong is aaa@aaa.aa)	-	
U4	req_question, guide_loc	an-nae jom hae jul-lae (Ok. Please guide me to his room)		0.78, 0.0
S4	guide()	ji-geum an-nae-hae-deu-ri-get-seum-ni-da. (I will guide you to his room right now.)	-	

Fig. 33. Example of simulated dialog (given WER = 0.10). SWB = Structure and Word BLEU, WER = Word Error Rate. D-BLEU = 0.96, Average of SWB = 0.83, Average of WER = 0.10.

6. Discussion and future work

This paper presented novel dialog simulation methods for use in the evaluation of spoken dialog systems. We proposed methods of simulating utterances and user intentions to replace real human users. We presented a simple and useful ASR channel simulation method based on linguistic knowledge. We also proposed some automatic evaluation metrics for measuring the quality of user simulation. The following briefly discuss its strengths, our experience and future work.

6.1. Data-driven simulation techniques

We introduce a method of simulating user intentions which is based on the CRF sequential graphical model. CRF allows sophisticated intention simulation through modeling an entire intention sequence given discourse information. Also we can take advantage of CRF as a conditional model which is able to capture arbitrary facts to describe the observation in the form of indicator functions. This allows developers to find and use useful rich discourse information for intention modeling.

To generate the utterances which express a certain intention, we developed domain-specific data-driven utterance simulation methods. We divided the domain language space, called SV space, based on the combination of dialog act and main goal, then estimated the transition probability distribution of structure tags to structure tags and emission probability distribution of structure tags to words. Using the probability distributions, we generated many possible user utterances at first and rescored the utterances with a SWB score, which is a structure and word interpolated BLEU. Both user intention and utterance simulators use a fully data-driven approach.

6.2. Evaluation metrics for user simulation

To evaluate the user simulation quality, we proposed several automatic evaluation metrics which can measure the quality of intention and utterance simulation results. As a naturalness measure of simulated dialog, we proposed the D-BLEU which is an applied version of original BLEU to measure the *n*-gram precision of

intention sequence with brevity penalty. While the D-BLEU can measure the similarity of a dialog to real dialogs which have the same dialog goal, KL-divergence can measure the similarity of all the simulated dialogs compared to all the real dialogs in terms of relative entropy. We examined the relationship between D-BLEU and human judgments, and the relationship between D-BLEU and KL-divergence. The examinations showed that D-BLEU score actually has the same tendencies as human judgments and that D-BLEU is feasible for scoring the simulated dialogs.

Inspired by the fact that the evaluation task of statistical machine translation is similar to evaluating simulated user utterances in user simulation, we adopted BLEU scores to evaluate the simulated utterances. We slightly modified the original BLEU to be able to score one utterance with semantically equivalent human utterances rather than machine translated documents with human-translated documents. Also, we expanded BLEU to calculate not only word sequence but also structure tag sequence, and interpolated them into the SWB score. This SWB score is actually used for rescoring the simulated utterances.

6.3. Experiences

To verify the feasibility of our simulation approach, we did two case studies. To build up the user simulation system, we collected human–machine dialog logs which contain the information including user utterances and corresponding SLU results, and discourse contexts. The training examples were collected from the real dialog logs, so we did not need annotation work basically. We examined the dialog examples, and selected good examples for training the intention and utterance simulators. A hundred dialog examples were used to prepare the user simulator. The user intention and utterance simulation models were automatically trained from the dialog logs, and the user simulator and linguistic-based ASR channel simulator were connected to the previously developed Korean navigation domain dialog system and Korean building guide robot domain. The iterative simulation provided the simulated dialog examples. To verify the feasibility of the proposed simulation techniques, we had human judges and our evaluation metrics score randomly selected simulated dialog examples, and examined the qualities of simulation results and the relationship between the metrics and human judgments. The experiments showed that our user simulation techniques actually generate natural dialog examples which are similar to the corresponding real dialogs and that the proposed user simulation evaluation metrics can help score the examples automatically.

6.4. Future work

In this research, we developed user simulation techniques which support intention and surface level simulation. Also we proposed the evaluation metrics which can tell the qualities of intention and utterance simulation. Using the techniques and metrics, we can collect abundant dialog examples, and select the desirable examples. In the future, we will analyze the quality of unseen utterance and intention sequences in terms of cooperativeness, expertise and others in manual or automatic way. Developing a personalized user simulation method to control the type and quality of unseen simulated dialog patterns is also in our future plan. We expect that a controllable personalized user simulator contributes to developing and evaluating spoken dialog systems.

Acknowledgements

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute for Information Technology Advancement) (IITA-2009-C1090-0902-0045).

References

Ai, Hua, Diane, J. Litman, 2008. Assessing Dialog System User Simulation Evaluation Measures Using Human Judges. Association for Computational Linguistics.

Boros, M., Eckert, W., Gallwitz, F., Gorz, G., Hanrieder, G., Niemann, H., 1996. Towards understanding spontaneous speech: Word accuracy vs. conceptaccuracy. In: 4th International Conference on Spoken Language Processing, vol. 2.

Bos, J., Klein, E., Lemon, O., Oka, T., 2003. DIPPER: Description and formalisation of an information-state update dialogue system architecture. In: 4th SIGdial Workshop on Discourse and Dialogue, Sapporo, Japan.

Chung, G., 2004. Developing a Flexible Spoken Dialog System Using Simulation. Association for Computational Linguistics, pp. 63–70.

Cuayahuitl, H., Renals, S., Lemon, O., Shimodaira, H., 2005. Human–computer dialogue simulation using hidden Markov models. Automatic Speech Recognition and Understanding, 100–105.

Deng, Y., Mahajan, M., Acero. A., 2003. Estimating speech recognition error rate without acoustic test data. In: 8th European Conference on Speech Communication and Technology.

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

Fosler-Lussier, E., Amdal, I., Kuo, H.K.J., 2002. On the road to improved lexical confusability metrics. In: ISCA Tutorial and Research Workshop (ITRW) on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology.

Georgila, K., Henderson, J., Lemon, O., 2005. Learning user simulations for information state update dialogue systems. In: 9th European Conference on Speech Communication and Technology: ISCA.

Georgila, K., Henderson, J., Lemon, O., 2006. User simulation for spoken dialogue systems: Learning and evaluation. In: 9th International Conference on Spoken Language Processing.

Greenberg, S., Chang, S., Hollenback, J., 2000. An introduction to the diagnostic evaluation of switchboard-corpus automatic speech recognition systems. NIST Speech Transcription Workshop.

Hunt, M.J., 1989. Figures of merit for assessing connected-word recognisers. Speech Input/Output Assessment and Speech Databases.

Jeong, M., Lee, G.G., 2006. Jointly predicting dialog act and named entity for spoken language understanding. Spoken Language Technology Workshop. IEEE 66–69.

Kim, K., Lee, C., Jung, S., Lee, G.G., 2008. A frame-based probabilistic framework for spoken dialog management using dialog examples. SigDial.

Kneser, R., Ney, H., 1995. Improved backing-off for *M*-gram language modeling. In: International Conference on Acoustics, Speech, and Signal Processing, vol. 1.

Lafferty, J.D., McCallum, A., Pereira, F.C.N., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning table of contents, pp. 282–289.

Lee, C., Jung, S., Eun, J., Jeong, M., Lee, G.G., 2005. Example and situation based dialog management for spoken dialog system. Automatic Speech Recognition and Understanding.

Lee, J., Kim, S., Lee, G.G., 2006. Grapheme-to-phoneme conversion using automatically extracted associative rules for Korean TTS system. In: 9th International Conference on Spoken Language Processing.

Lee, C., Jung, S., Lee, G.G., 2008. Robust dialog management with *n*-best hypotheses using dialog example and agenda. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Ohio.

Lemon, O., Georgila, K., Henderson, J., 2006. Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: The talk towninfo evaluation. In: Eurospeech.

Levin, E., Pieraccini, R., Eckert, W., 2000. A stochastic model of human–machine interaction for learning dialogstrategies. IEEE Transactions on Speech and Audio Processing 8 (1), 11–23.

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

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

López-Cózar, R., Callejas, Z., McTear, M., 2006. Testing the performance of spoken dialogue systems by means of an artificially simulated user. Artificial Intelligence Review 26 (4), 291–323.

Needleman, S.B., Wunsch, C.D., 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.

Papineni, K., Roukos, S., Ward, T., Zhu, W.J., 2001. BLEU: A method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318.

Pietquin, O., 2004. A Framework for Unsupervised Learning of Dialogue Strategies. Ph.D. thesis, Faculty of Engineering, Mons, Belgium.

Pietquin, O., Dutoit, T., 2006. A probabilistic framework for dialog simulation and optimal strategy learning. Audio, Speech and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on] 14 (2), 589–599.

Pietquin, O., Renals, S., 2002. ASR system modeling for automatic evaluation and optimization of dialogue systems. In: ICASSP.

Prommer, T, Holzapfel, H., Waibel, A., 2006. Rapid simulation-driven reinforcement learning of multimodal dialog strategies in human–robot interaction. In: ICSLP: ISCA.

Schatzmann, J., Georgila, K., Young, S., 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In: 6th SIGdial Workshop on Discourse and Dialogue.

Schatzmann, J, Thomson, B ., Weilhammer, K., Ye, H., Young, S., 2007a. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In: Proc. of HLT/NAACL.

Schatzmann, J., Thomson, B., Young, S., 2007b. Error simulation for training statistical dialogue systems. Automatic Speech Recognition and Understanding, 526–531.

Schatzmann, J., Thomson, B., Young, S., 2007c. Statistical user simulation with a hidden agenda. SigDial.

Scheffler, K., Young, S., 2000. Probabilistic simulation of human–machine dialogues. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2.

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

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

Stuttle, M., Williams, J., Young, S., 2004. A framework for dialog systems data collection using a simulated ASR channel. In: International Conference on Spoken Language Processing.

Torres, F., Sanchis, E., Segarra, E., 2008. User simulation in a stochastic dialog system. Computer Speech and Language 22 (3), 230–255.

Williams, J.D., Young, S., 2007. Partially observable Markov decision processes for spoken dialog systems. Computer Speech & Language 21 (2), 393–422.

Young, S., Odell, J ., Ollason, D ., Valtchev, V., Woodland, P., 1997. The HTK Book. Entropic Cambridge Research Laboratory.

Zukerman, I., Albrecht, D.W., 2001. Predictive statistical models for user modeling. User Modeling and User-Adapted Interaction 11 (1), 5–18.