Using Utterance and Semantic Level Confidence for Interactive Spoken Dialog Clarification

Sangkeun Jung, Cheongjae Lee and Gary Geunbae Lee
Computer Science and Engineering
Pohang University of Science and Technology
{hugman,lci80, gblee}@postech.ac.kr

Spoken dialog tasks incur many errors including speech recognition errors, understanding errors, and even dialog management errors. These errors create a big gap between the user's intention and the system's understanding, which eventually results in a misinterpretation. To fill in the gap, people in human-to-human dialogs try to clarify the major causes of the misunderstanding to selectively correct them. This paper presents a method of clarification techniques to human-to-machine spoken dialog systems. We viewed the clarification dialog as a two-step problem - Belief confirmation and Clarification strategy establishment. To confirm the belief, we organized the clarification process into three systematic phases. In the belief confirmation phase, we consider the overall dialog system's processes including speech recognition, language understanding and semantic slot and value pairs for clarification dialog management. A clarification expert is developed for establishing clarification dialog strategy. In addition, we proposed a new design of plugging clarification dialog module in a given expert based dialog system. The experiment results demonstrate that the error verifiers effectively catch the word and utterance-level semantic errors and the clarification experts actually increase the dialog success rate and the dialog efficiency.

Categories and Subject Descriptors: Abstract Interpretation [**natural language processing**]:

General Terms: Clarification Dialog, Clarification Expert, Error Verification, Spoken Language Understanding, Dialog Management

Additional Key Words and Phrases: Spoken Dialog System, Error Detection

1. INTRODUCTION

Clarification Dialog is one of the dialog types used to resolve misunderstanding between human to human, or human to machine dialogs. Table I shows a typical clarification dialog in an error-prone spoken environment in a TV-guide domain. For a successful clarification dialog like the example, we need to solve two problems.

Table I. Example of a clarification dialog.

U1: I want to watch the ***drama Hae-Sin.***
S1: ***Documentary Her-Jun*** is on ***MBC.***
Do you want to watch it?
(Both Drama and Hae-Sin are misrecognized, respectively, as Documentary and Her-Jun)
U2: No, I want to watch the ***drama Hae-Sin.***
S2: Please repeat the program name you want to watch
(System internally verifies that the 'drama' and 'ae-Sin' are not correctly recognized. The system tries to clarify the TV-program name first which is relatively more important.).
U3: Hae-Sin
S3: Drama ***Hae-sin*** is on ***KBS***. Do you want to watch it?
U4: Yes
S4: OK, I am playing ***Hae-Sin*** on ***KBS***
(***Bold-italic*** words designate the important content information to a TV-guide domain dialog system. This dialog example is a Korean transcript, but translated into English to help readers' understanding.)

The first problem is to select the targets to be clarified and classify the error types, while the second one is to clarify and recover the targets in an intuitive and efficient way. The first problem is conventionally called the belief confirmation task. Belief confirmation techniques have been explored by many researchers, and many confidence measures have been developed. Most of the research focused on measuring confidence scores of the speech recognition results since the confidence score enables user or system to avoid unnecessary recognition or discourse managing failure. The second problem is establishing an adequate system response to handle the errors detected at the belief confirmation. It is usually implemented by generating system responses using the belief confirmation results. In addition to the above two problems, adequate dialog system architecture is also important since clarification dialog should be carried naturally along with non-clarificaiton dialog.

In this paper, we introduce the word and utterance error verification method for the belief confirmation problem. For establishing an adequate clarification system response, we introduce a new concept of clarification expert. We propose natural clarification dialog enabled dialog system architecture as well.

This paper is organized as follows: Previous researches are surveyed in section 2. System overview, dialog task and an expert-based dialog management architecture as our base-line architecture for the clarification are described in section 3. Confidence measuring based on Automatic Speech Recognition (ASR) and Spoken Language Understanding (SLU) scores are showed in section 4. The details of the clarification expert and clarification strategy will be described in section 5. Extensive experiments and analyses are shown in section 6, and finally a conclusion will be drawn in section 7.

2. RELATED WORKS

Confidence scoring, especially in speech recognition results, has been conducted by

many researchers, who have proposed to compute a confidence score to indicate the reliability of any recognition decision made by ASR systems. Some representative works include [Kamppari and Hazen 2000; San-Segundo et al. 2001; Zhang and Rudnicky 2001], and many others. In addition to the acoustic confidence score, recently, researchers have included more semantic level information for belief confirmation. [Cox and Dasmahapatra 2000] used Latent Semantic Analysis (LSA), and [TORRES et al. 2005] used the information from understanding module. [Huet et al. 2007] proposed a score function that combines the parts of speech (POS), language model, and acoustic scores at the sentence level. This work has some parallels with our work in the sense of computing confidence score for utterance level not only word level. Many good features and methods of confidence measurement in the level of speech recognition decoder and in the level of semantic level are introduced [Hazen et al. 2000; Hazen et al. 2002; Jiang 2005; Cox and Dasmahapatra 2000].

Confidence scoring method is recently applied to verify the result of intention recognition and has been proved helpful for dialog system [Singh et al. 2002]. [Paek and Horvitz 2000] tried to use a single integrated estimator to classify the type of errors. The classifier examines both speech recognition results and language understanding results.

[Gabsdil and Bos 2003] introduced a technique to include acoustic confidence scores as returned by automated speech recognizers in generic semantic representations. Their work tries to pinpoint the source of the problem more accurately, identifying the word(s) or semantic information with the lowest confidence scores. Their and our approaches have a similarity on focusing on slot level information which is relatively important to process dialog management, However, there are differences that our work tries to calculate utterance level confidence as well as slot level confidence for clarification dialog.

[Higashinaka et al. 2005] applied confidence scoring method to detect errors on slot in the frame. They introduced various slot related and discourse related features for training classifier. This work has some parallels with our own work in the sense of detecting errors on the slot level, and using language understanding features into calculating confidence. In this paper, however, we proposed three level confidence scoring method on word, utterance and slot and value level, and suggest new clarification dialog architecture and strategy using these three-level confidences while the previous work only focused on detecting slot level errors.

Establishing clarification dialog strategy to handle error-prone utterances is implemented using the belief confirmation technique. [Torres et al. 2005] showed how to use confidence values for calculating a transition probability in the dialog state-transition network. [Mctear et al. 2005] developed an object-oriented dialog system which determines the confirmation status using the discoursePeg, extended from the set formulated by [Heisterkamp and McGlashan 1996]. [Misu and Kawahara 2005] introduced building clarification question by adding useful constraints which maximize information gain in their information retrieval system.

For multimodal clarification dialog, [Rieser and Lemon 2006] investigated the use of machine learning to explore human multimodal clarification strategies and the use

of those strategies to decide, based on the current dialogue context. They used several features including local, dialog history and user model features for determining clarification strategies while our work uses ASR, SLU and domain knowledge features for calculating confidence scores.

[Purver 2006] introduced an approach to clarification request which is fine-grained enough to include specific clarification request questions about individual words and phrases, and general enough to handle arbitrary phrase types for both user and system clarification requests. However, this work handles only text-based dialog system while our work tries to solve system side clarification request problem for spoken dialog system.

Tremendous confidence measuring and error detecting methods have been proposed, but plugging clarification dialog in dialog system architecture and building clarification strategy using domain knowledge have not been researched deeply. In this paper, we introduce a three level confidence scoring method which covers from speech recognition to semantic interpretation, and propose an actual implementation of clarification dialog in the given dialog system architecture as well.

3. USER FRIENDLY TV PROGRAM GUIDE SYSTEM WITH SPEECH INTERFACE

3.1 System Overview

We aim to overcome the problems of information transaction and device control taking natural speech input. To make the proposed system robust, we added clarification dialog ability by verifying word, utterance and semantic level errors and establishing clarification dialog strategies. The flow of the clarification dialog can be summarized as:

(1) ASR recognizes user utterance
(2) Error verifier detects each word of the utterance that may include ASR errors (1st phase).
(3) SLU tries to understand user utterance and makes the slot value structures with confidence scores
(4) Error verifier detects user utterance that may not be adequate to continue the dialog processing (2nd phase).
(5) Error verifier checks each slot and value pair to retain the important information to continue the dialog processing (3rd phase).
(6) Error verification results are passed to Dialog Manager (DM). If needed, DM calls the clarification expert to make possible clarification questions to clarify user's original intention.
    (a) If needed, clarification expert establishes adequate clarification questions to confirm user's original intention. It decides the clarification strategy using error verification result, discourse status and domain knowledge.
(7) System retrieves TV-program data or controls the device.

3.2 EPG Dialog Task

The task involves TV control and information retrieval using natural speech input.

TV control includes several functionalities such as turn on/off, alarm setting and reservation of the recording. Users can retrieve specific information from the TV program database (Electronic Program Guide; EPG) with interactive dialog. Semantic frame and the entity list are described in Table II and Table III respectively. Typical TV-guide dialog scenarios are shown in Table IV.

Table II. Example of a semantic frame

User Utterance	When does the drama haesin start?
Dialog Act **Main Goal**	wh search starttime
Component : **[slot]=value**	[genre]="drama" [program]="haesin"

Table III. List of semantic frame entities

Dialog Act wh,yn,request,say,say_noun,say_prop
Main Goal search_program, search_channel, search_day, search_starttime, search_endtime, search_currenttime, search_currentdate, move_channel, alarm, record, TV_on, TV_off
Components genre, channel, date, start_time, end_time, cast, day, program

Table IV. Example of typcial TV-guide dialog scenarios

Example	
U1:	Turn on the TV.
S1:	Turned on the TV.
	I am playing channel KBC now.
U2:	What's on MBC?
S2:	'Wangkotsunyounim' is on MBC.
U3:	Who is the main actor?
S3:	'Lee dahae' is the main actrees of the 'Wangkotsunyounim'.
	Do you want to watch it?
U4:	No. Is there any news now on?
S4:	**(System detects an error and tries to clarify user's original intention)**
	Please say the genre of the program again?
	(specific information level clarification)
U5:	News
S5:	Do you want to watch a news program?
U6:	Yes
S6:	'SBS 8 si news' is on now.
	Do you want to watch it?
U7:	Yes
S7:	Ok, I am playing SBS now.

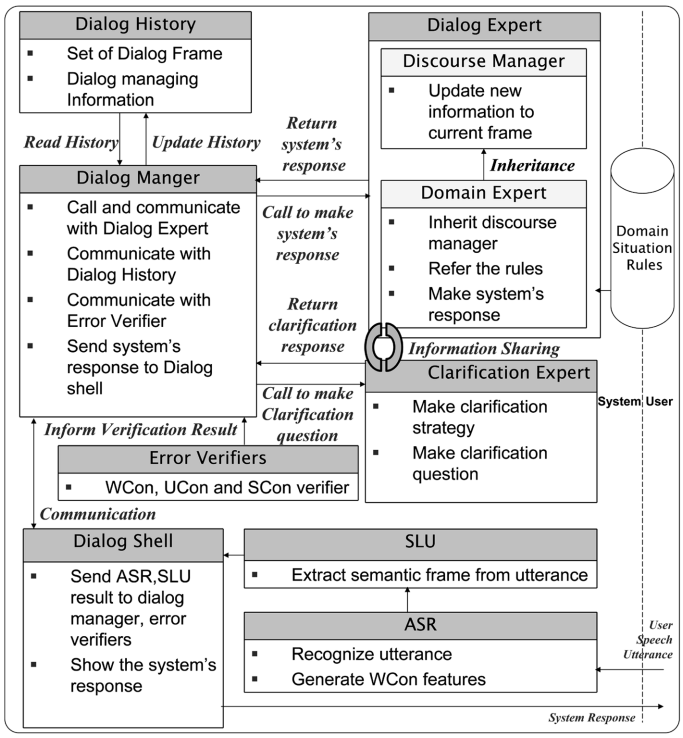


Fig. 1. Situation based dialog management architecture with the clarification expert.

3.3 Situation-based Dialog Management

In recent years, many number of frame based approaches have been developed. CMU Communicator [Rudnicky et al. 1999] takes event-driven, mixed-initiative approach and decides next system's response based on the current context. CMU Communicator continues the dialog by filling each form without a specific order. Recently Queen's Communicator [ONeill et al. 2005] designed the dialog system in object-oriented architecture with distributed and inherited functionality.

Inspired by the work of [ONeill et al. 2005], and motivated to overcome the conventional dialog systems' rigidity and in inflexibility, we developed a situation based spoken dialog management system [Lee et al. 2006]. Our system uses a situation-based dialog management strategy. A situation is determined by various information of the current dialog status including user's utterance and intention (dialog acts), the set of semantic slots and values, the confidence status of each slot, the discourse history of the dialog in a current session, the system's previous intention (response) and the database retrieval results of the current user query.

Like [ONeill et al. 2005]'s domain experts, we pursued an expert-based dialog management strategy to conduct a specific domain-oriented dialog. Each expert is designed as a specialist for handling specific dialog patterns. For example, the TV-guide expert handles TV-guide related utterances.

Table V. An example of the error verifications

User Utterance & Speech Recognized Utterance
I want to watch Drama HaeSin on KBS (User utter.) I want to watch Drama BaeSin on SBS (ASR Result)
1st Phase : WCon Verification
I(Wcon = 89.45% / WCon ouput=Believable) want (78% / Believable) to(45.32% / Unbelievable) watch (78.67% / Believable) Drama (87.32% / Believable) Baesin (65.2% / Believable) on (93% / Believable) SBS (29.24% / Unbelievable)
2nd Phase : UCon Verification
I want to watch Drama BaeSin on SBS UCon_output = (67.42%, Believable)
3rd Phase : SCon Verification
channel - SBS: (SCon = 65%, ouput = Believable) genre - drama: (97%, Believable) program_name - BaeSin: (45%, Unbelievable)

Fig. 1 illustrates the situation-based dialog system architecture with the connection to the clarification expert. Dialog Manager is a hub module that communicates with ASR, SLU and error verifiers. It also manages other dialog components in the architecture. Detail explanation about the dialog system architecture can be found in [Lee et al. 2006].

For implementing ASR, we developed a Korean speech recognizer based on the Hidden Markov Model Toolkit (HTK). We modified the HTK ([Young et al. 2002]) for making and providing decoder level information. Our SLU was implemented using the concept-spotting approach to extract the semantic structures of the utterances such as dialog acts, main goal and component slots for the main goal [Eun et al. 2004].

4. SLU-BASED CONFIDENCE MEASURE AND CLARIFICATION STRATEGY

We formulated the clarification dialog process as a two-step problem. The first problem is deciding on a clarifying target by belief confirmation and the second problem is establishing an adequate clarification dialog strategy to confirm user's original intention. For the first problem, to select the proper targets to be clarified, we designed error verifiers in three step and developed a method to calculate each measure of confidence. The first phase is Word Confidence (WCon) verification which is similar to conventional belief confirmation on words recognized by the speech recognizer. The second phase is Utterance Confidence Score (UCon) verification which assesses the whole utterance's appropriateness level to progress the dialog further. The third phase is Semantic Level Confidence Score (SCon)

verification which examines the semantically important slot-and-values which are extracted by the SLU from user utterances. The targets of the verification in each step along with an example are shown in Table V.

In general, confidence scores in the ASR can be formulated as follows:

- Word confidence score (WCon) = P(Believable | Hypothesized word)
- Utterance confidence score (UCon) = P(Believable | Hypothesized utterance = list of hypothesized words)

In the past years, these confidence scores have been used for clarification dialog processing. However, confidence scores of the ASR alone are not enough measures to decide belief status and establish clarification strategy, since WCon and UCon usually do not contain semantic level information which is also important for clarification dialog. Therefore, we need to redefine confidence scores to capture the belief statuses which reflect both speech recognition and semantic interpretation level.

4.1 Word Confidence Score Verification

To measure the belief status in the word level (1st phase), we followed conventional belief confirmation approaches. It examines every word that is recognized by the speech recognizer. However, we don't use the word confidence scores directly for our clarification dialog management. Instead, the goal of this step is to provide basic information to the utterance confidence measuring phase (2nd phase) and semantic level confidence measuring phase (3rd phase).

We adopted some of the good confidence measures from [Hazen et al. 2000]. We used a Maximum Entropy (MaxEnt) classifier for combining good confidence features and calculating the confidence scores for classifying the word recognition errors. The followings are the description of the Word confidence score (WCon), output classes and the input features for our MaxEnt classifier:

Word Confidence Score

- **Output Class**: Believable/Unbelievable
- **Normalized acoustic scores (NAC)**: frame normalized acoustic scores of a node in the lattice
- **Language model scores (LM)**: word trigram scores $P(W_i|W_{i-2};W_{i-1})$
- **N-best purity (NP)**: the fraction of the N-best hypotheses in which the hypothesized word appears in the same position in the utterance (position consistency of the hypothesis)
- **Duration**: the duration time of the word in the utterance
- **Word Length**: the length of the word
- **Word Lexical**: the lexical form of the recognized words

We normalized the acoustic score to NAC by dividing it with NFrames (number of frames) which is relative duration time. N-best purity represents the fraction of the n-best candidate words in the same position in the utterance [Hazen et al. 2000]. Word Lexical feature is converted to the binary value on the sparse vector of lexical list. The detail example of the WCon calculating step is shown in Table VI.

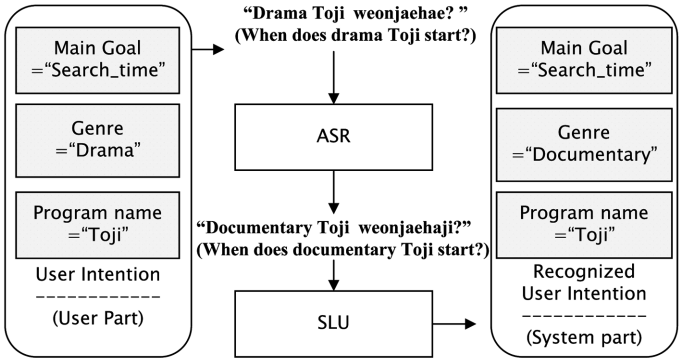
Table VI. Example of features of WCon.

User Utterance :	"Drama Haesin boko sipta" (I want to watch drama Haesin)
ASR Result :	"Drama Baesin boko sipta" (I want to watch drama Baesin)
Targets: each words	"Drama / Baesin / boko / sipta"

Detail Example : "Drama"

NAC	-76.18	LM	-97.73
NP	0.75	Duration(sec)	0.78
Length	3	Lexical["Drama"]	1.0

Words	WCon	Output Classes
Drama	0.78	Believable
Baesin	0.64	Unbelievable
boko	0.82	Believable
sipta	0.96	Believable



1. WCon(documentary)=0.65,
   WCon(Toji)=0.94, WCon(weonjaehaji)=0.87
2. Utterance Confidence Score (Ucon) = 2/3
   # of origianl information = 3
   (Main Goal = "search_time",
   Genre = "Drama", Program Name = "Toji")

   # of survived information = 2
   (Main Goal = "search_time", Program Name = "Toji")

Fig. 2. Example of the concept of UCon.

4.2 Utterance Confidence Score Verification

A common utterance level confidence score is the geometric average of the posterior probabilities of the phones in the hypothesis, or equivalently, the arithmetic average of the log posterior probability of the phones. In this research, however, we viewed the utterance confidence as a survival rate of user's original information rather than

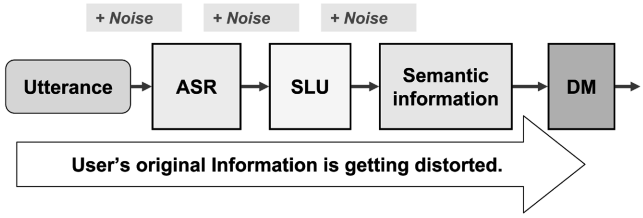


Fig. 3. Typical dialog process.

as a simple average of acoustic posterior probability.

A very noticeable point in spoken dialog processing is that the integration of the whole dialog processing is sequentially chained, which means that the errors and noises are not only propagated to the next module but also amplified (Fig. 3). In this circumstance, in order to measure the utterance level belief status, we should be able to tell whether the current set of information the dialog manager received is believable or not. We need some measures to represent the status of the survivability of the noisy-channeled information. Here, we defined the unit of survival information as slot and value pairs in the semantic frames. We defined the concept of utterance confidence score as follows:

– Utterance Confidence Score = The ratio of correctly carried user's original information (intention) to the dialog manager

Fig. 2 shows the concept of utterance confidence score. We capture the number which represents the ratio of the correctly carried user's original information in the point of the dialog manager. Therefore we can simply define the conceptual utterance confidence score as below:

$$UCon = \frac{\textbf{number of survived information}}{\textbf{number of original information}}$$

However, the direct calculation of UCon is impossible since the system cannot know the number of original information, and the number of survived information should be estimated by the system. It means that we need a classifier to figure out whether the information is successfully survived or not with some approximation. To approximate the UCon without the knowledge of the number of original information, we took the following two assumptions:

– Assumption 1: SLU performance on reference text is high enough to use it to make a reference structured information to generate a training data for the classifier.
– Assumption 2: P (Believable|UCon feature information) indicates the ratio of the approximated survived information.

Our SLU module's performance is over 99% on reference text [Eun et al. 2004]. Therefore assumption 1 can be satisfied. Assumption 2 means that if the probability P (Believable|UCon feature information) is 0.25, we could translate this probability as UCon itself. However, this probability interpretation can be thought in several ways:

Original Sentence		**ASR Recognized Sentence**		
"Drama Toji weonjae hae?" (When does the drama Toji start?")		"Documentary Toji weonjaehaji?" (When does the documentary Toji start?)		
SLU result on reference utterance		**SLU result on recognized utterance**		
Dialg Act	Wh	Dialg Act	Wh	0.98
Main Goal	Search_time	Main Goal	Search_time	0.95
Genre	Drama	Genre	Documentary	0.88
Program name	Toji	Program name	Toji	0.93
Training Feature				
Predicate : Unbelievable (Since 1 information is not survived) **Features** : "The word error rate predicted" = 1/3 "Mean of the understanding scores" = (0.98+0.95+0.88+0.93)/4 Act and Score = ["wh", 0.98] Goal and Score = ["Search_time", 0.95] Component slot's name and its SLU score = { ["Genre",0.88] ["Program_name",0.93]} Act_Goal_and_Component's value relation (binary feature) = 1				

Fig. 4. Example of UCon training corpus generation.

(1) the number of survived information = 1, the number of original information = 4
(2) the number of survived information = 2, the number of original information = 8
(3) ...

To satisfy these assumptions, we trained the classifier in a special way. First, to collect the reference training data, we collected the reference user utterance with speech file, and executed SLU modules on the transcribed utterances. Second, we recognized the speech file with ASR, and then executed SLU module also on the recognized utterances to extract the structured information. An example of the corpus preparation to generate the training data is shown in Fig. 4.

The output predicates in the training data are decided as below:

– Predicate = 'Believable'; all information of reference utterance has survived in the SLU on the recognized utterance
– Otherwise, 'Unbelievable'

Namely, the predicate 'Believable' in the training data means that all of the information has successfully survived. Therefore, if P(Believable|UCon feature information) is 1, it means that all information of the recognized utterance has successfully survived even though the number of original information is unknown. To calculate the utterance confidence score, we trained the MaxEnt classifier using the following features:

UCon Verification

- **Output Class**: Believable/Unbelievable
- **The word error rate predicted**: the ratio of the word errors predicted by the WCon verifier
- **Mean of the understanding scores**: mean of the log-likelihood scores of the SLU
- **Act and Score**: the values of the dialog act which are extracted by the SLU and its log-likelihood
- **Goal and Score**: the values of the main goal which are extracted by the SLU and its log-likelihood
- **Component slot's name and its SLU score**: the names of the component slots which are spotted by the SLU and its log-likelihood
- **Act_Goal_and_Component's value relation**: the binary feature representing the concurrent pattern of dialog, and main goal and lexical value of component

Because of the instability of the ASR scores, it is not a good idea to simply use the bare ASR scores, such as NAC and LM, directly for calculating the UCon since these direct scores sometimes would work as noises. Therefore, we used the final results from the WCon verification, i.e., the word error rate predicted. WCon classifier verifies every words of the target utterance and tells 'Believable' or 'Unbelievable' word. We consider the 'Unbelievable' word as a mis-recognized word by speech recognizer, and calculate the word error rate predicted using WCon verification results for the utterance. This word error rate predicted is used as a feature for the UCon classifier. In addition to the predicted word error rate, we found some more useful features for calculating the UCon using our SLU results as represented in Fig. 4. We discovered that there is a reasonable relationship between the value of the dialog act, the main goal, and the name spotted as a component. For example, there is a "`wh - search starttime - genre - program_name`" relation. In other words, there is a certain co-occurred pattern in the values of the dialog act, the main goal, and the names of the component slots. We captured these relationships and used them as binary feature for measuring utterance appropriateness. Fig. 5 shows the example of ASR and WCon results and the calculating step for UCon.

As shown in Fig. 6, after calculating the UCon, the dialog system decides the utterance level clarification strategy. The high UCon means that the utterance may be well recognized, and would be well understood in the sense of ASR and SLU. If the UCon is lower than the threshold, system tags the utterance as "Unbelievable" and passes the information to the clarification expert to decide adequate system responses.

However, if the UCon is higher than the threshold, the system tags the utterance as "Believable" and continues to the third phase verification - Semantic level confidence verification. Even though the utterance itself can be estimated as "Believable", there may be some errors in the level of semantic slot and value interpretation. The goal of the third phase verification is to find the specific slot-value level errors in detail.

4.3 Semantic Confidence Score Verification

User Utterance: "Drama Haesin boko sipta"
(I want to watch drama Haesin)
ASR Result : "Drama Baesin boko sipta"
(I want to watch drama Baesin)
SLU Result :

Target	Value	Understanding Score
Dialog_Act	Request	0.96
Main_Goal	Search_program	0.89
Genre	"Drama"	0.93
Program_Name	"Baesin"	0.76

WCon Result :
WCon("Drama") = 0.78, WCon("Baesin") = 0.64,
WCon("boko") = 0.82, WCon("sipta") = 0.96
[The underlined word is predicted to "Unbelievable"]

Feature Name	Feature Value
"The word error rate predicted"	0.25
"Mean of the understanding scores"	(0.96 + 0.89 + 0.93 + 0.76) / 4 = 0.885
Act and Score = "Request"	0.96
Goal and Score = "Search_Program"	0.89
Component slot's name and its SLU score	
"Genre"	0.93
"Program_Name"	0.76
Act_Goal_and_Component's value relation (binary feature)	
"Request→ Search_Program→Genre→Program_name"	1

Measure the UCon :
UCon("Drama Baesin boko sipta") = 0.84

Fig. 5. Example of features and calculation of UCon.

The semantic confidence score (SCon) verifier examines every slot and value pair which is extracted by the SLU in the sense of both ASR and SLU. Hence, we can capture the combined confidence of both speech recognition and semantic interpretation together.

We used the same features and the same MaxEnt classification method that we have used in the WCon measuring phase except one additional feature:

Semantic Confidence Score Verification
– **Output Class**: Believable/Unbelievable
– **Understanding Scores**: the likelihood of slot and value of the spotted words generated by the SLU
– **Other features**: using the same features of WCon measuring phase

Notice that SCon measuring step is very similar to WCon measuring step. The difference of the two measuring steps is that SCon only checks the slot and value

pairs in the utterance with the confidence score of SLU and ASR's information together while WCon checks every word in the utterance with only the information of ASR. The detailed example of calculating SCon is shown in Table VII.

If all of the slots and values of the utterance are classified as "Believable", the utterance and the slot-values are directly passed to the dialog manager. However the "Unbelievable" tagged slot-values and the confidence scores are passed to the clarification expert to decide on a proper clarification strategy on the specific slot values.

5. EXPERT-BASED CLARIFICATION DIALOG STRATEGY

From the results of the utterance and the UCon/SCon scores, we can obtain the specific targets which should be clarified. The targets can be a full sentence or a set of important words in a dialog. To clarify these targets efficiently and systematically, we introduce a clarification expert in our situation-based dialog management architecture.

Our dialog system is strongly based on expert system architecture. Each expert is responsible for handling a certain domain dialog, and is designed to manage domain specific dialog patterns. Therefore, if we reformulate the 'clarification' as specific dialog patterns, we could model the clarification as one of the specific dialog expert. However, there should be some differences between a clarification expert and other domain experts. The differences are as follows:

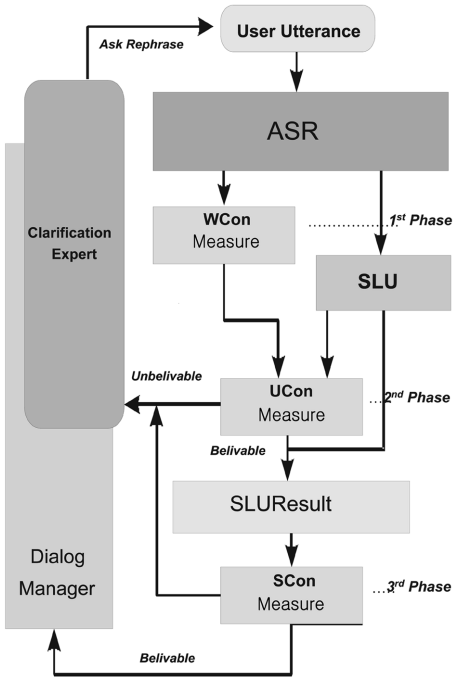


Fig. 6. Flow of error verification and clarification.

Table VII. Example of features and calculation of SCon.

User Utterance :	"Drama Haesin boko sipta" (I want to watch drama Haesin)
ASR Result :	"Drama Baesin boko sipta" (I want to watch drama Baesin)

Slot_and_Value of the components

Slot	Value	Understanding Score
Genre	"Drama"	0.93
Program Name	"Baesin"	0.76

NAC	-76.18	LM	-97.73
NP	0.75	Duration(sec)	0.78
Length	3	Lexical["Drama"]	1.0
Understanding Score	0.93		

SCon results on the slot and value pairs:

Slot_and_value pairs	SCon
[Genre]="Drama"	0.99
[Program_Name]="Baesin"	0.76

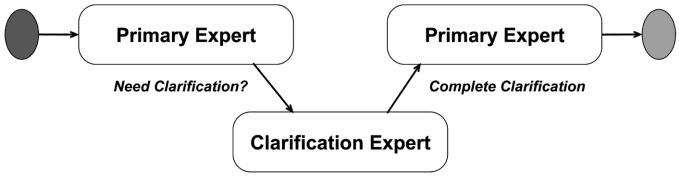


Fig. 7. Switch to the clarification expert.

– The clarification expert is a **secondary expert** different from the primary working domain experts.
– The clarification expert should be **domain independent**.

– The clarification expert should be able to **share all the information** of the current primary working domain expert.

As shown in Fig. 7, if the utterance or some specific words/phrases need to be recovered, the dialog manager stops the working primary domain expert and gives the control to the clarification expert. In this step, dialog manager enables the clarification expert to access all of the primary expert's information including the dialog frame and the discourse history. Clarification expert decides on a clarification dialog strategy by considering the confidence scores provided by the error verification result and other domain-related information. The clarification expert provides two different clarification strategies, *Utterance Clarification (UC)* and *Slot_and_Value Clarification (SC)*, based on the error verification results.

UC occurs when the UCon verifier alarms the dialog manager that the user's utterance is unbelievable to continue the dialog further. If the utterance clarification is needed, the dialog manager and the clarification expert work as follows:

(1) The dialog manager asks the clarification expert to check if the UCon is lower than the rephrase threshold.
(2) If the UCon is lower than the threshold, the clarification expert asks the user to rephrase the whole utterance again, and then returns the dialog control to the dialog manager.
(3) If there was whole utterance-level clarification rephrase, the dialog manager continues the dialog with the new utterance after user's rephrasing.

The SC occurs when the third phase SCon verifier alarms the dialog manager that the user's utterance is believable but the specific slot_and_value pairs are not. If the slot_and_value clarification is needed, the dialog manager and the clarification expert work as follows:

(1) The dialog manager only takes the believable slot_and_value pairs, but ignores the unbelievable slot_and_value pairs
(2) The dialog manager calls the clarification expert with SC flag, informing that the slot_and_value clarification is necessary and simultaneously delivering the current unbelievable slot_and_value pairs.
(3) The clarification expert examines the set of unbelievable slot_and_value pairs, and decides which slot_and_value pair should be clarified first.
(4) The clarification expert asks the user to rephrase the word for the decided slot_and_value pair (in Table 7, if the clarification expert decides to clarify the 'Program_Name = Bae-Sin' first, the clarification expert asks the user to repeat the program name again).
(5) After a partial rephrasing, the clarification expert writes the new information from the partially rephrased into the primary expert's dialog frame.
(6) Repeat steps 3 to 5 until there are no more unbelievable slot_and_value pairs.

In this process, the most important step is the step 3, which is the step of selecting and choosing the target slot_and_value pair to be clarified first. To develop a more efficient and systematic slot_and_value clarification strategy, the following properties are being considered for the clarification expert:

– Property 1: Dependency between each slot's error information and dynamic change of the error information relationship
– Property 2: Relative importance of the error information

Most of the slots have a strong dependency with other slots in the same dialog domain. For example, 'Larry King Live' is always broadcasted on 'CNN' and the Korean popular drama 'Hae-Sin' is always on 'KBS'. There is a certain dependency between the program title and the channel. Without considering these dependencies, we end up with taking unnecessary clarification steps. The following clarification target example (Table VIII) demonstrates the importance of considering the dependency in choosing a proper clarification strategy. In this example, if we are

Table VIII. Target example of the selected slot-values for clarification

User Utterance	I want to watch drama Hae-Sin on KBS
ASR Result	I want to watch drama Bae-Sin on SBS
Targets needed to be clarified	[Program_Name] = Bae-Sin [Channel] = SBS

unaware that there is a strong dependency between 'Hae-Sin' and 'KBS', the clarification expert asks users to rephrase the names of both channel and program. However, if we are aware of the dependency, clarification expert does not need to clarify the channel at the moment that 'Bae-sin' is clarified to 'Hae-sin' because system already knows that 'Hae-Sin' is always on 'KBS'.

Property 2 can be used for choosing the clarification order among multiple targets. As in Table VIII, when there are more than two slot-values to be clarified, the clarification expert considers the relative importance property to set the priority of the clarification. In most of the cases, the priority from relative importance is closely related to the range of the slot types (Fig. 8). As we can see, in most of the cases, if the 'playing actor/actress' in the smallest range is determined surely, 'program_name', 'channel' and 'genre' are determined automatically. Therefore, in Table VIII, the clarification expert tries to clarify the 'program name' first.

In addition to the priority of each slot, clarification expert also considers the semantic confidence score that is generated by the SCon measuring module. The clarification ordering algorithm can be described in Table IX. The values of α and β were determined heuristically. Through several experiments on development data set, we decided $\alpha = 2/3$ and $\beta = 1/3$.

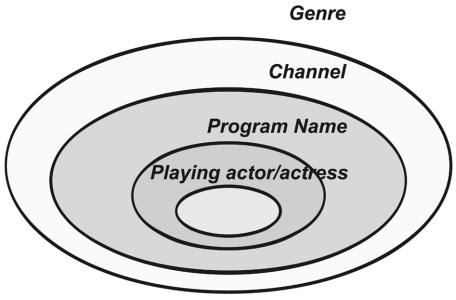


Fig. 8. Range of the information types on EPG domain.

Table IX. Clarification ordering algorithm

For each slot-and-value, calculate
Clarification Order Score(k)
$= \alpha * Priority\_Score(k) + \beta * (1 - SCon(k))$
Where k is the slot-and-value pair

Choose the slot-and-value k which has
the largest Clarification Order Score

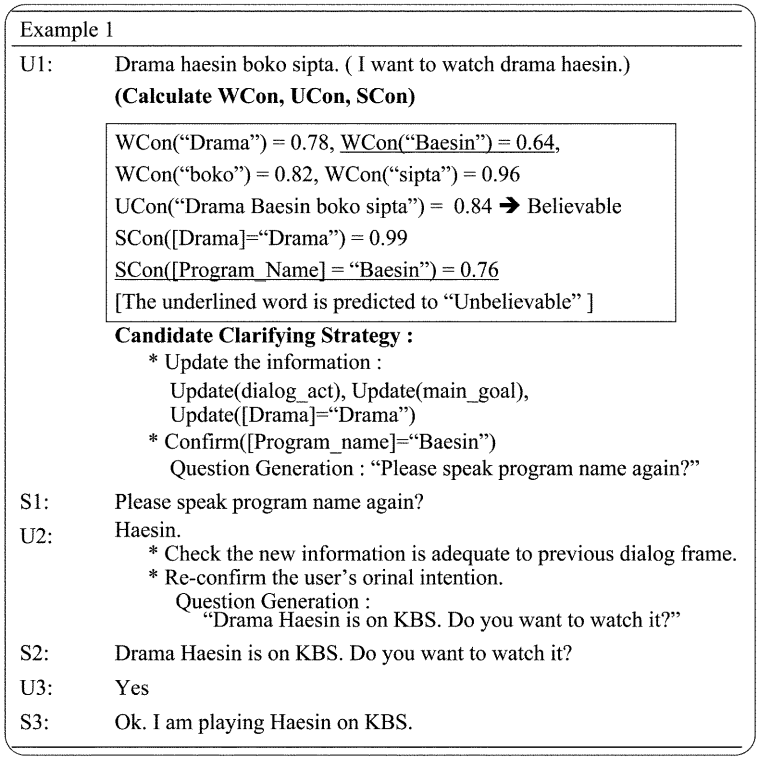


Fig. 9. Example of clarification dialog with detail confidence and strategy.

Even though the clarification expert uses the prior knowledge of the specific domain in this step, the clarification expert itself is totally domain independent, since the prior knowledge is not directly embedded in the clarification expert itself but is embedded in the primary expert's dialog frames. One of the examples of the clarification dialog is shown in Fig. 9.

6. EXPERIMENTS AND ANALYSES

We collected a monthly TV schedule for developing and testing our dialog system. The ASR's language model and the SLU are trained to be able to handle all kinds of possible utterances which might occur in the selected TV schedule. The performance is word error rate (WER) 15.3% for 45747 utterances. To verify our system's performance, we experimented on both error verifiers and end-to-end dialogs.

6.1 Experiments on the Error Verifications

6.1.1 *Experiments on WCon.* Our error verification method is designed in a cascade manner. WCon is used for calculation of UCon and SCon, therefore, the reliable performance of WCon is very important.

We collected 45747 examples which include speech recognized result, reference speech transcript, spoken language understanding result, and reference language

Table X. Confusion Matrix on WCon based Verification (for 9074 utterances).

Reference \ *Predict*	Correct	Error
Correct	4445	414
Error	378	3837
False Rejection Rate :	9.73%	
False Acceptance Rate :	7.83%	

Table XI. performance based on the UCon (Threshold = 0.71) (for 9074 utterances).

	SLU Performance
UV Positive Utter.	91.04%
UV Negative Utter.	70.06%
All Utter.	75.12%

understanding. We showed the examples by utterance and split 80/20. 80% are used for training and 20% are used for testing. WCon features are extracted from 36598 recognized results and trained. Using the trained classifier, 9074 utterances are tested. To confirm the performance of the WCon verifier, we drew a confusion matrix for WCon verifier and calculated False Rejection Rate (FR) and False Acceptance Rate (FA).

$$FA = \frac{\text{\# of incorrectly accepted words}}{\text{\# of accepted words}}$$

$$FR = \frac{\text{\# of incorrectly rejected words}}{\text{\# of rejected words}}$$

As we can see in Table X, our WCon verification shows 9.73% for false rejection rate and 7.83% for false acceptance rate.

6.1.2 *Experiments on UCon.* UCon verifier was also trained using 36598 utterances. The features are extracted from ASR result and SLU result of each utterance, and used for training MaxEnt classifier for UCon verifier.

Higher UCon utterance keeps the original user's intention much more than the lower UCon utterance. In other words, the original information would survive much better on the higher UCon utterance in the series of ASR and SLU processing. To confirm this, we tested the UCon verifier based on its threshold and *Survived Information Rate (SIR)*. We separated the 9074 (about 20% of 45747) utterances based on UCon scores into UV (Utterance Verification) Positive utterances and UV Negative utterances.

$$SIR = \frac{\text{\# of survived information}}{\text{\# of original information}}$$

Notice that SIR has the same form of UCon, but SIR can be calculated directly for testing by comparing reference user information and extracted user information in the system while UCon can be only approximated by taking classification probability.

Table XI shows the performance of the UCon-based utterance verification. The UV Positive utterances are the utterances that the utterance verifier tags 'Believable'. We tried to perform the SLU on the set of UV-Positive tagged, UV-Negative tagged and all mixed utterances. As we can see in Table XI, the UV-Positive utterances can be more accurately SLU decoded than the UV-Negative utterances. It means that the utterance verifier can guarantee not to process improper utterances that have a higher chance of mis-interpretation for the dialog system.

Fig. 10 depicts the relationship between the SIR and the UCon threshold with the coverage. The SIR of UV positive utterances increases according to the increase of the UCon threshold, and converges finally to 100% but the coverage decreases to 0%. We set the UCon threshold 0.71 by examining the curve on development data set in this research.

6.1.3 *Experiments on SCon.* SCon verification allows the clarification expert to know which information should be clarified. SCon verification occurred when UCon verifier passed the utterance with the sign of 'Believable'. Therefore the tested utterances are fewer than the tested utterances of WCon and UCon. Tested utterances are 1798, and all of these are tagged 'Believable' by the UCon verifier.

As we can see in Table XII, our SCon-based slot and value verifier shows a false rejection rate of 6.75% and a false acceptance rate of 5.74% for the utterances which were marked as UCon 'Believable'.

6.2 Experiments on the Clarification Dialog

To demonstrate the effectiveness of the clarification expert based on the error verification result, we did end-to-end dialog tests on various input modes. We gathered 10 test volunteers who had not used our system before, and tutored them

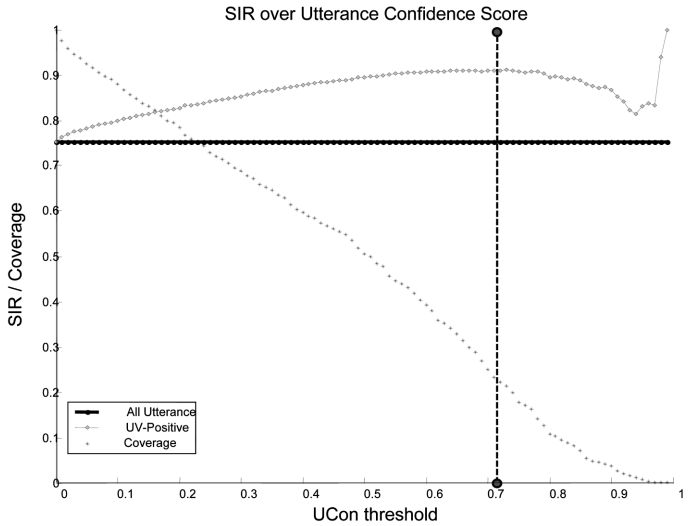


Fig. 10. SIR vs. UCon Threshold.

Table XII. Confusion Matrix on SCon-based Slot and Value Verification (for 1798 utterances)

Reference \ *Predict*	Correct	Error
Correct	1625	5
Error	99	69

False Rejection Rate : 6.75%
False Acceptance Rate : 5.74%

on the functionality of the system. Then, we asked 10 test volunteers to set the 5 dialog goals (tasks) that they want to achieve using our TV-guide dialog system. Each volunteer was asked to make conversations with the dialog system in three different modes - text input mode, speech input mode and speech input with clarification mode. Hence, we collected 50 dialogs to each mode.

The volunteers are asked to evaluate every system's response and the clarification's success. The condition of dialog success and clarification success are below.

- Dialog Success: if the users accomplished the goal that they set before, then the dialog is successful.
- Clarification Success :
- If the system asks the user to repeat the utterance or some information to clarify, and the clarification request is acceptable, then it is successful. In here, 'acceptable' means that error verifier successfully detects some errors on the utterance.
- If the user confirms that the clarification questions induce the dialog process successfully, then it is successful.

The results are summarized in Table XIII. As we can see in Table XIII, the speech input with clarification dialog has higher success rate than the one without it.

Another noticeable point of this table is that the average system action turns per dialog decrease from 13.37 in speech input to 11.64 in our clarification dialog mode. This decrease comes from the preventive effects of the system's mis-interpretaion by pre-verifying the appropriateness of the utterances and the slot-values. Namely, our error verifiers discourage the dialog system from doing actual actions on the improper utterances and this discouragement contributes to the decrease of the average total turns per dialog. User, System action and the total dialog turns are defined in this experiment as follows:

- User turn: the number of the user's utterances: + 1 cost for one utterance
- System Action turn: sum of the two action turns
  - System response utterances: + 1 cost for one system response utterance
  - Physical Action: including following actions: turn on/off TV, moving channel, DB Accessing : + 1 cost for each physical action
- Total turn: sum of the total cost of user's turn and the system action's turn

As shown in the experiment results, our clarification approach is successful at decreasing the system action turn cost. It shortens the conversation and therefore increases the user's satisfaction.

Table XIII also shows the occurrence and success rates of the utterance clarification (UC) and the slot-value clarification (SC). While the occurrence rate of the UC is quite high, the occurrence rate of the SC is very low, since the UC works as a first-step remedy against the errors.

There is a trade-off between utterance clarification and the slot_and_value clarification. Utterance clarification has a high speech recognition performance but forces the user to speak again much longer utterance than partial slot_and_value clarification. To examine which clarification strategy is better, Clarification Efficiency Rate (CER) was calculated as follows:

$$\text{Clarification Efficiency Rate} = \frac{\sum_{i=1}^{NC} \frac{NI_i}{NW_i} U_i}{NC}$$

NI :	number of semantic slots in the clarification utterance
NW :	number of words in the clarification utterance
NC :	number of times the clarification occurs in the dialog
U :	user's confirmation
	1 : clarification is successful
	0 : clarification has failed

The large value of CER means that the clarification method can recover the information with small effort, which refers to the user's additional labor for clarification.

As shown in Table XIV, the clarification efficiency rate on the utterance clarifi-

Table XIII. Quantitative performance measures for the clarification dialog effects

	Mode0	Mode1	Mode2
Dialog Success rate	0.85	0.72	0.83
Avg. total turn per dialog	17.01	20.05	19.07
Avg. user turn per dialog	5.67	6.68	7.42
Avg. system action turn per dialog	11.34	13.37	11.64
GSR	0.79	0.69	0.57
UC occurrence rate per user utter.	-	-	0.19
UC occurrence rate per dialog	-	-	1.42
UC Success rate	-	-	0.81
SC occurrence rate per utter.	-	-	0.02
SC occurrence rate per dialog	-	-	0.19
SC Success rate	-	-	0.45

Mode0: Text input, Mode1: Speech input without clarification, Mode2: Speech input with error verification based clarification, GSR: Good system response rate per system utterance

Table XIV. Efficiency of UC and SC

	UC	SC
Clarification Efficiency Rate	0.72	0.45

cation is much higher than the slot and value clarification. In addition to the superiority of the clarification efficiency rate, the utterance clarification method shows a much higher clarification success rate. This shows that the utterance clarification is a more suitable strategy for the dialog clarification.

The lower success rate of the slot-value clarification is attributed to the low speech recognition performance, especially for the short word recognition used for the slot-values. This problem could be solved by using multi-level speech recognizer according to the different situation.

From the experiment results, we can conclude that our utterance and semantic score based clarification approach is viable and feasible: First, it increases the dialog success rate for better task completion. Second, it reduces the total turn for dialog from 20.05 in speech input to 19.07 in error verification based clarification approach. Third, it prevents possible mis-interpretation by verifying the utterances and the slot values carefully before actual system action.

7. CONCLUSION

In this paper, we solved two different but inter-related problems of spoken language error detection and dialog clarification. To implement clarification dialog in the spoken dialog system, we formulated clarification dialog as two-step problem - belief confirmation and clarification strategy establishment. To measure the belief status, we verified word and utterance error in three step. For modeling an efficient clarification dialog strategy, we modeled a clarification expert to specially handle the clarification dialogs based on our situation-based dialog management model. The clarification expert analyzes the error verification results and two essential properties to determine on efficient and systematic clarification strategies. Our utterance and semantic score based clarification approach shows the incensement of dialog success rate for better task completion, and decrement of the total turn for dialog. Furthermore, we proposed natural architecture of plugging a clarification dialog module into given dialog system. Through various experiments on both error verification and the clarification dialogs, we confirmed that our error verification method and clarification expert approaches were successful for implementing robust clarification dialogs in the spoken human-to-machine interfaces.

ACKNOWLEDGEMENTS

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

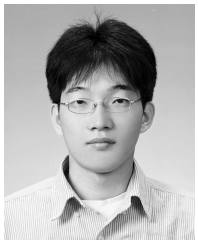
REFERENCES

Cox, S. and Dasmahapatra, S. 2000. A semantically-based confidence measure for speech recognition. Proc. of ICSLP 4, 206–209.

Eun, J., Lee, C., and Lee, G. 2004. An information extraction approach for spoken language understanding. Proc. of ICSLP, 2145–2148.

Gabsdil, M. and Bos, J. 2003. Combining acoustic confidence scores with deep semantic analysis for clarification dialogues. Proc. of the 5th international workshop on computational

semantics (IWCS-5), 137–150.
Hazen, T., Burianek, T., Polifroni, J., and Seneff, S. 2000. Recognition Confidence Scoring for Use in Speech Understanding Systems. Proc. of the the ISCA ASR.
Hazen, T., Seneff, S., and Polifroni, J. 2002. Recognition confidence scoring and its use in speech understanding systems. Computer Speech and Language 16, 1, 49–67.
Heisterkamp, P. and Mcglashan, S. 1996. Units of dialogue management: an example. Proc. of ICSLP 1, 200–203.
Higashinaka, R., Sudoh, K., and Nakano, M. 2005. Incorporating Discourse Features into Confidence Scoring of Intention Recognition Results in Spoken Dialogue Systems. Proc. of ICASSP 1, 25–28.
Huet, S., Gravier, G., and Sebillot, P. 2007. Morphosyntatic processing of N-best lists for improved recognition and confidence measure computation. Proc. of the 8th Eurospeech, 1741–1744.
Jiang, H. 2005. Confidence measures for speech recognition: A survey. Speech Communica- tion 45, 4, 455–470.
Kamppari, S. and Hazen, T. 2000. Word and phone level acoustic confidence scoring. Proc. of ICASSP 3, 1799–1802.
Lee, C., Jung, S., Eun, J., Jeong, M., and Lee, G. 2006. A situation-based dialog management using dialog examples. In Proc. of the ICASSP. 69–72.
McTear, M., O'Neill, I., Hanna, P., and Liu, X. 2005. Handling errors and determining confirmation strategies: An object-based approach. Speech communication 45, 3, 249–269.
Misu, T. and Kawahara, T. 2006. Dialogue strategy to clarify users queries for document retrieval system with speech interface. Speech Communication 48, 9, 1137–1150.
ONeill, I., Hanna, P., Liu, X., Greer, D., and Mctear, M. 2005. Implementing advanced spoken dialogue management in Java. Science of Computer Programming 54, 1, 99–124.
Paek, T. and Horvitz, E. 2000. Conversation as action under uncertainty. Proc. of Uncertainty in Artificial Intelligence, 455–464.
Purver, M. 2006. CLARIE: Handling Clarification Requests in a Dialogue System. Research on Language & Computation 4, 2, 259–288.
Rieser, V. and Lemon, O. 2006. Using machine learning to explore human multimodal clarification strategies. Proc. of the COLING/ACL on Main conference poster sessions, 659–666.
Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu, W., and Oh, A. 1999. Creating natural dialogs in the Carnegie Mellon Communicator system. Proc. of Eurospeech 4, 1531–1534.



Sangkeun Jung received the B.S., M.S. degress in Computer Science and Engineering from POSTECH, 2004, 2006 respectively. He is now a Ph.D student of Pohang University of Science and Technology (POSTECH). His research includes development, modeling and simulation of dialog.

Cheongjae Lee received the B.S. degree of both Computer Science & Engineering and Life Science from POSTECH, 2004. He now is a Ph.D. student of POSTECH. His research interests include dialog modeling, affective dialog.

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